

<http://dx.doi.org/10.17703/JCCT.2019.5.1.401>

JCCT 2019-2-51

이동성 지원을 위한 SDN 기반의 네트워크 플랫폼 개발

Development of SDN-based Network Platform for Mobility Support

이완직*, 이호영**, 허석렬***

Wan-Jik Lee*, Ho-Young Lee**, Seok-Yeol Heo***

요약 최근 급증한 클라우드 컴퓨팅의 수요 해결과 네트워크 가상화 서비스 지원을 위해 등장한 SDN(Software Defined Networking)은 많은 기업과 단체들에 의해 차세대 네트워크 기술로 주목받고 있다. 하지만, 원래 SDN이 설계된 유선 네트워크와는 달리, 무선 네트워크에서의 SDN은 노드의 이동성을 제공해주지 못하는 제약과 가지고 있다. 본 논문에서는 SDN의 기존 오픈플로우 프로토콜을 확장하여, SDN Controller가 네트워크의 무선자원을 관리하고 노드들의 이동성을 지원하는 SDN 기반의 네트워크 플랫폼을 개발하였다. 본 논문의 이동성 지원 기능은 SDN Controller의 무선 자원관리 기능을 이용하여 두 개 이상의 무선 인터페이스를 이용한 노드의 이동을 지원할 수 있는 장점을 가진다. 또한 본 논문에서는 구현 기능을 시험하기 위해, 다양한 이동 실험에 따른 여러 전송 성능에 관한 파라미터들을 측정하였고, 하나의 무선 인터페이스를 사용했을 때와 두 개의 인터페이스를 사용했을 때의 성능에 관한 파라미터를 비교하였다. 본 논문에서 제안한 SDN 기반 네트워크 플랫폼은 SDN 환경에서 무선 네트워크의 자원을 모니터링하고 노드들의 이동성을 지원하는데 사용될 수 있을 것으로 기대된다.

주요어 : SDN, 오픈플로우, 이동성 지원, 네트워크 플랫폼

Abstract SDN(Software Defined Networking) has emerged to address the rapidly growing demand for cloud computing and to support network virtualization services. Therefore many companies and organizations have taken SDN as a next-generation network technology. However, unlike the wired network where the SDN is originally designed, the SDN in the wireless network has a restriction that it can not provide the mobility of the node. In this paper, we extended existing openflow protocol of SDN and developed SDN-based network platform, which enables the SDN controller to manage the radio resources of its network and support the mobility of the nodes. The mobility support function of this paper has the advantage that a node in the network can move using its two or more wireless interfaces by using the radio resource management function of the SDN controller. In order to test the functions implemented in this paper, we measured parameters related to various transmission performance according to various mobile experiments, and compared parameters related to performance using one wireless interface and two interfaces. The SDN-based network platform proposed in this paper is expected to be able to monitor the resources of wireless networks and support the mobility of nodes in the SDN environment.

Key words : SDN, Openflow, Mobility Support, Network Platform

*정회원, 부산대학교 IT응용공학과 (주저자)

Received: October 10, 2018 / Revised: November 12, 2018

**정회원, ㈜포미트, 부산대학교 IT응용공학과 대학원 (참여저자)

Accepted: December 18, 2018

***정회원, 부산대학교 IT응용공학과 (교신저자)

*Corresponding Author: syheo@pusan.ac.kr

접수일: 2018년 10월 10일, 수정완료일: 2018년 11월 12일

Dept. of IT and Engineering, Pusan National Univ, Korea

게재확정일: 2018년 12월 18일

I. 서 론

최근 클라우드 컴퓨팅의 수요 증가와 네트워크 가상화 서비스 지원과 같이 네트워크에 관한 복잡한 요구 사항을 해결하기 위해 등장한 SDN(Software Defined Networking) 개념은 네트워크 제어 기능이 물리적 네트워크와 분리되어 있어서 네트워크 장치에 대해서 독립적이고 효율적인 관리가 가능하다는 장점을 가진다[1]. SDN은 소프트웨어를 활용하여 네트워크 구성에 유연성을 도입하는 것을 목표로 한다. 따라서 네트워크 관리자가 소프트웨어 응용을 통해 서로 다른 네트워크 장비와 다른 제조사들의 장치들을 쉽게 구성하고 모니터링 할 수 있기 때문에 SDN 방식의 네트워크 구현 방식은 복잡한 네트워크 요구 사항을 쉽게 수용할 수 있다[2],[3].

하지만 SDN은 유선 기반의 코어 네트워크 환경에 적용하기 위한 기술로 설계되어 있기 때문에 최근 급격하게 증가하고 있는 무선 및 이동 네트워크 환경에 적용하기에는 적합하지 않다는 단점이 있다.

코어 네트워크 환경과 별도로 접속(access) 네트워크 환경에는 무선, 이동통신 기술의 발전에 따라 스마트폰, 태블릿 및 랩톱과 같은 모바일 단말의 사용이 급속하게 증가하고 있다. 실제로 2016년 월별 모바일 데이터 사용량은 7 엑사바이트(Exabyte(10^{18}))였지만 2018년에는 2.5배인 17 엑사바이트, 2021년에는 49 엑사바이트가 될 것으로 예측되고 있다[4]. 이러한 시장 전망에서 볼 수 있듯이 기존의 포화된 데스크탑 시장과는 달리 스마트폰, 태블릿 등 이동 단말 시장은 끊임없이 성장하고 있다.

이러한 환경에 따라 IETF(Internet Engineering Task Force: 국제 인터넷 표준화 기구)는 단말의 이동성을 위해 Mobile IP(MIP)와 Proxy Mobile IP(PMIP)라는 기술로 해결 방법을 제시하였다. 하지만 이러한 방식은 기존 네트워크 환경에서 호스트 기반 이동성 지원 방안으로 단말의 이동을 위해 IP 프로토콜 스택 수정이 필요하다는 단점 등으로 인해 현재 거의 활용되지 않는 실정이다.

따라서 유선 네트워크를 관리 기술로 주목 받는 SDN 기술에 무선 자원제어와 단말의 이동성 기능을 추가하면, SDN 기반의 통일된 기술에 의해 유·무선 네트워크 자원 관리 및 단말 이동성 제어가 가능하다는 장점을 가질 수 있다.

이러한 점에 착안하여, 본 논문에서는 SDN에서 사용하는 오픈플로우 프로토콜을 확장하여 SDN Controller에 무선 네트워크의 자원을 관리하는 기능과 이를 바탕

으로 단말의 이동성을 지원하는 기능을 추가하였다. 그리고 논문에서 설계한 기능들을 SDN Controller, 무선 AP(Access Point) 및 MN(Mobile Node)로 구성된 SDN 네트워크 플랫폼에 구현하였으며, 자원정보 취득과 이동성 지원 기능을 SDN 네트워크 플랫폼 상에서 직접 시험하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 SDN과 SDN 기반의 이동성 지원에 대한 기존 연구들에 대해 소개하고, 3장에서는 SDN 기반 네트워크에서 이동성 지원 플랫폼 설계에 대해서 설명한다. 그리고 4장에서는 구현과 시험 결과에 대해 기술하며, 마지막으로 5장에서는 결론과 향후 계획에 대해 설명한다.

II. 관련 연구

SDN의 가장 큰 특징은 네트워크의 제어 기능을 물리적 네트워크와 분리한 것이다. 따라서 SDN은 기본적으로 네트워크 제어 기능이 기존의 스위치나 라우터 등의 하드웨어와 별도로 분리되며, 데이터 전달 기능과도 분리되어 개발 및 실행될 수 있는 네트워크 구조를 가진다[5]. 분리된 SDN의 제어 기능은 네트워크 스위치(하드웨어) 상의 데이터 경로와 상호 작용할 수 있는 기능이 필수적인데, 이러한 상호 작용을 위해 개발된 기술이 오픈플로우(Openflow) 프로토콜이다. 오픈플로우 프로토콜은 네트워크 스위치(혹은 라우터)의 패킷 포워딩 기능과 제어 기능을 분리하고 이들 두 기능 간의 통신을 위한 제어 명령을 제공한다. 이는 외부 제어장치(서버)에 의하여 구동되는 소프트웨어가 장비 제조사에 무관하게 스위치 내의 패킷 경로 결정을 가능하게 한다. 오픈플로우 프로토콜은 스탠포드 대학과 UC 버클리 대학의 공동연구 결과로 탄생하였으며, 현재는 개방형 네트워크 재단(ONF: Open Networking Foundation)에서 개발과 관리를 담당하고 있다[6].

SDN에 이동성을 지원하기 위한 연구로서 Guimarães 외 4명[7]은 SDN 환경에서 IEEE 802.21 WG(Working Group)에서 정의한 매체 독립적인 핸드오버(MIH: Media Independent Handover) 서비스를 제공하는 단일 프레임워크를 제안하였고, Raza 외 3명[8]은 무선 이기종 네트워크 환경에서 IP 이동성을 위한 SDN 기반 아키텍처를 제안하였다. 또한 Seong-Mun Kim 외 4인[9]은 IETF에서 정의한 PMIP 구현을 위해 SDN 기능의 일부

도입을 제안하였다.

하지만 이런 연구들은 SDN에서 단말 이동성 지원을 위한 아키텍처 제안이나, IETF MIP 구현을 위한 방안으로 SDN의 일부 기능을 차용한 형태로 볼 수 있다.

III. SDN 기반의 이동성 지원 플랫폼 설계

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

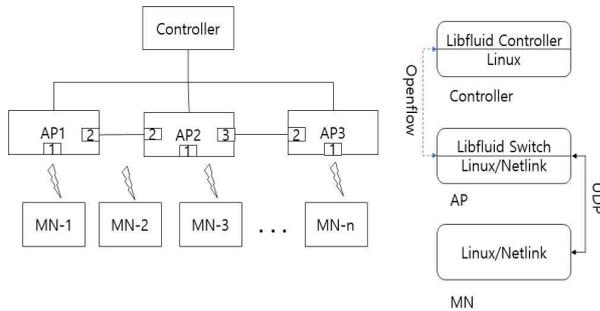


그림 1. SDN 기반 이동성 지원 네트워크 플랫폼
 Figure 1. SDN-based mobility support network platform

표 1은 그림 1의 플랫폼을 구성하는 각 시스템들에 대한 설명이다.

표 1. 네트워크 플랫폼 구성 요소
 Table 1. Components of network platform

Component	H/W	O/S	S/W
Controller	노트북 (Intel i5)	Linux (Ubuntu 14.04)	Libfluid Controller S/W
AP	노트북 (Intel i5)	Linux (Ubuntu 14.04)	Libfluid Switch S/W, Netlink Libnl
MN	노트북 (Intel i5)	Linux (Ubuntu 14.04)	Netlink Libnl

표 1에서 SDN Controller와 SDN AP에는 각각 Libfluid Controller와 Libfluid Switch S/W 모듈이 탑재된다. Libfluid Controller와 Libfluid Switch는 오픈소스인 Libfluid 패키지[10]를 이용하여 본 논문에서 구현한 SDN Controller와 Switch 소프트웨어로써 이동성 제어 메시지를 SDN 오픈플로우 메시지로 연동하는 핵심 기능을 수행하게 된다.

그림 1의 SDN Controller와 AP들 간에는 오픈플로우

프로토콜에 의한 네트워크 제어 정보만 전달되는 Control Path가 설정되는데, 이 Control Path에는 SDN 규격에 의해 일반적인 사용자 데이터는 전달되지 않고, 오픈플로우 프로토콜에 의한 제어 정보만 전달된다. AP와 MN들 간 또는 AP와 AP 사이에는 실제 데이터가 전달되는 Data Path가 생성된다. 이 Data Path에는 데이터 이외, UDP 메시지 형태로 본 논문에서 설계한 AP와 MN 들의 무선/이동 관련 메시지들이 교환된다.

이러한 제어 정보 전달 구조에 따라, 본 논문의 목적인 SDN Controller에 의한 이동성 지원을 위해서는 Data Path에서 전송되는 무선/이동 메시지를 오픈플로우 프로토콜 메시지에 캡슐화하여 SDN Controller로 전달하고, 반대로 SDN Controller에서 전달되는 무선/이동정보 메시지를 AP나 MN으로 전달하는 연동 기법이 필요하다.

1. Libfluid와 Netlink 오픈소스

Libfluid는 SDN Controller와 Switch 구현을 위해 필요한 기능을 제공하는 C++ 라이브러리와 예제 구현 소스를 제공하는 오픈소스 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 라이브러리는 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.

Netlink socket[11]은 커널과 유저 영역 프로세스 간의 IPC(Inter Process Communication) 용으로 개발된 커널 인터페이스지만 주로 커널영역과 유저영역 간에 네트워크 정보를 전달하기 위해 사용되고 있다.

본 논문에서는 AP와 MN의 각 네트워크 인터페이스 정보와, Wifi 관련 무선자원 정보를 획득하기 위해 Netlink 소켓의 라이브러리(Libnl) 함수를 사용하였다.

2. SDN 기반의 이동성 지원 플랫폼 설계

본 논문의 플랫폼은 MN, AP, Controller로의 세 가지 시스템으로 구성된다. 플랫폼의 각 시스템 내부에는 본 논문에서 구현한 S/W 모듈이 존재하는데, 이들

S/W 모듈의 기능과 연동 내용은 그림 2와 표 2에 각각 나타내었다.

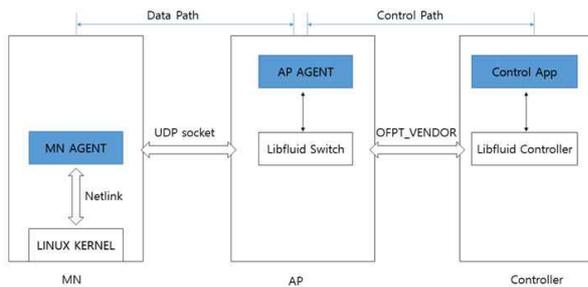


그림 2. 시스템 내의 S/W 모듈 연동
Figure 2. Interworking of S/W modules in system

표 2. S/W 모듈 기능
Table 2. Functions of S/W module

응용	기능 및 동작
MN AGENT	<ul style="list-style-type: none"> Netlink 라이브러리 이용, SDN 네트워크를 구성하고 있는 AP에 최초 접속 시 자신이 접속하고 있는 AP명, 인터페이스, 신호 세기 등 자신의 정보를 Control App에게 전송 무선 품질이 일정 품질 이하로 떨어졌을 때 Controller에게 이를 통보하고 신호가 양호한 AP를 찾아 접속 시도 Control App의 요구 시에 자신의 무선 자원정보를 Control App에게 전송
AP AGENT	<ul style="list-style-type: none"> Netlink 라이브러리를 이용하여 SDN 네트워크에 최초로 접속 시 자신의 AP 이름, 자신과 접속 중인 MN의 정보를 Control App에게 전송 Control App가 요구할 경우, 자신의 무선 자원정보를 Control App에게 전송
Control App	<ul style="list-style-type: none"> MN과 AP의 Agent로부터 정보를 받아 저장 MN과 AP의 무선 자원 정보를 모니터링 MN Agent로부터 이벤트를 받아 다음에 접속할 AP 정보를 받아 스위치의 flow table를 수정하고 MN Agent에게 이를 통보

그림 2의 MN과 AP 시스템 내부에는 본 논문에서 구현한 MN AGENT, AP AGENT라는 응용 S/W 모듈이 탑재되고, Controller에는 Control App이 탑재된다. 음영 박스로 표시되어 있는 이 모듈들은 MN의 이동성을 지원하고, Controller가 AP와 MN의 자원정보 수집을 수행하는 모듈들이다. 또한 Libfluid Switch와 Libfluid Controller는 libfluid 오픈소스로 구현한 SDN Switch와 Controller 기능을 수행하는 S/W이다. Libfluid Switch/Controller는 Libfluid 오픈소스를 이용하여 구현한 기본적인 SDN Switch와 Controller 기능에 이동성 지원 기능을 추가적으로 구현한 S/W이다.

MN AGENT 모듈은 본 논문에서 설계한 이동성 지원을 위한 제어 메시지(3.3 절에서 설명)들을 AP를 경유하여 최종적으로 Controller 내부의 Control App과 교환한다. 이러한 이동성 지원 메시지들은 MN과 AP 간에서는 UDP 패킷에 캡슐화 되고, AP와 Controller 간에는 SDN 규격에 의해 오픈플로우 메시지로 변환되어 전달된다. 이러한 연동 작업을 위해 Libfluid Switch 모듈은 먼저 이동성 지원 메시지를 탐지하는 기능을 수행해야 한다. Libfluid Switch 모듈은 특정 조건에 부합하는 패킷을 이동성 지원 메시지로 판단하고 OFPT_VENDOR 메시지로 변환하여 Controller에게 전송하는 기능을 가지고 있다.

이러한 메시지 변환 작업은 AP 상에서 SDN 규격인 flow 테이블의 flow entry를 등록하는 형태로 설계하였는데, 이 flow entry는 그림 3과 같이 특정 조건을 표현하는 flow-match 필드, 조건에 따른 Action 이름, 실제 처리 기능을 구현한 handler function으로 구성된다.

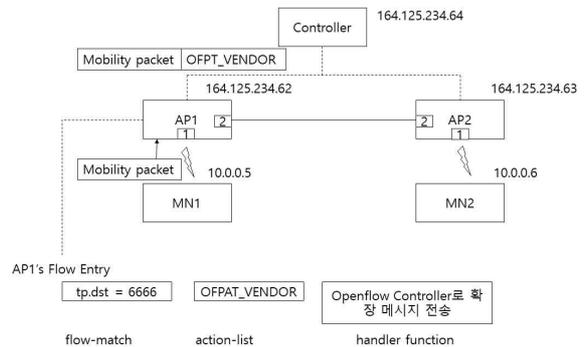


그림 3. AP 내부의 이동성 지원 메시지 변환
Figure 3. Mobility support message conversion in AP

본 논문에서는 그림 3과 같이 flow-match 필드를 목적지 UDP 포트가 6666번(논문에서 임의로 선정한 값)이면 MN에서 보내는 UDP 패킷의 헤더를 제거하고, 메시지 type이 OFPT_VENDOR인 오픈플로우 메시지로 캡슐화를 한 뒤, Controller에게 전송하는 flow entry를 작성하였다.

3. 이동성 지원 메시지 설계

본 논문에서 설계한 SDN 기반의 이동성 지원 메시지는 그림 4와 같은 형식으로 구성된다. 이동성 지원 메시지는 MN과 AP 사이에는 UDP에 의해 전송되고 MN과 Controller 사이의 SDN control path에서는 앞서

설명한 바와 같이 오픈플로우의 OFPT_VENDOR 형식으로 캡슐화되어 전송된다.

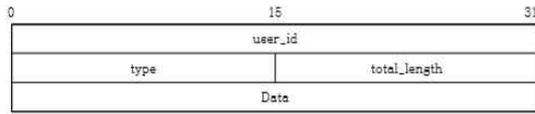


그림 4. 이동성 지원 메시지 형식
 Figure 4. Mobility support message format

표 3. 이동성 지원 메시지들
 Table 3. Mobility support messages

Type	Description
INTERFACE_INFO_REQ	Controller에서 MN의 이동성 관련 자원 정보를 수집하기 위해 MN에게 정보를 요청
INTERFACE_INFO_RES	MN에서 Controller에게 자신의 이동성 관련 자원 정보를 제공
STA_INFO_REQ	Controller에서 AP의 정보를 수집하기 위해 AP에게 해당 AP의 무선 정보를 요청
STA_INFO_RES	AP에서 Controller에게 자신의 무선 정보를 제공
VIOLATE_REPORT	Controller가 Controller에서 지정한 무선 품질(threshold)보다 떨어질 경우 발생
VIOLATE_REPORT_ACK	Controller가 판단한 이동할 AP를 지정하여 연결을 지시
HELLO	SDN 네트워크에 제일 처음 접속하거나 자신의 정보가 바뀌었을 시에 정보를 제공
HELLO_ACK	Controller에서 지정한 최소 무선품질들을 MN에게 알림
AP_CHANGE_D_REPORT	AP 연결 확인 및 flow entry 변경 요청
AP_CHANGE_D_REPORT_ACK	flow entry 변경 완료로 통보
DEL_FLOW_ENTRY_REQ	다중 인터페이스일 때 이전 인터페이스의 연결 flow entry 삭제 요청
DEL_FLOW_ENTRY_RES	이전 연결을 위한 flow entry 삭제 완료 통보

메시지의 user_id 필드는 AP나 MN의 고유 ID(Identification)를 나타낸 값으로, Controller는 0의 값을 가지고, Controller에서는 user_id의 값으로 각 MN이나 AP를 식별한다. 표 3에는 이동성 지원 메시지의 형식(type)과 기능을 요약하였다.

INTERFACE_INFO_REQ, INTERFACE_INFO_RES, STA_INFO_REQ, STA_INFO_RES는 Controller에서 AP나 MN의 정보를 수집할 때 사용되는 메시지이다. HELLO, HELLO_ACK는 프로그램이 시작되거나 연결

이 초기화되면 자신의 정보를 Controller와 통신할 때 사용하는 메시지이다. AP_CHANGED_REPORT는 MN이 Controller부터 새로 접속할 SSID를 받아 연결한 다음 flow entry를 변경해달라는 요청하는 메시지이다. DEL_FLOW_ENTRY_REQ는 MN이 두 개 이상의 무선 인터페이스를 사용할 경우, 이전의 flow entry 삭제를 요청하는 메시지이다.

IV. SDN 기반의 이동성 지원 플랫폼 구현 및 시험 결과

1. 이동성 지원 플랫폼 구현

논문에서는 설계한 이동성 지원 플랫폼은 그림 5와 같은 환경에서 구현하고 테스트하였다.

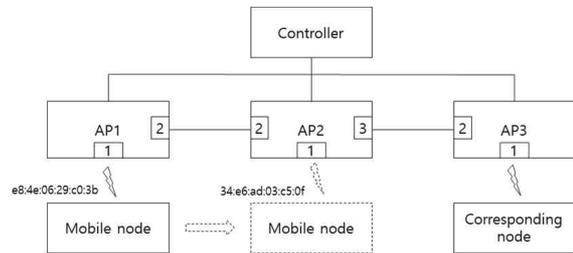


그림 5. 이동성 지원 플랫폼 구현 및 테스트 환경
 Figure 5. Mobility support platform implementation and test environment

그림 5는 MN이 Corresponding node로 표시된 단말과 데이터 전송 중에 AP1에서 AP2 또는 AP2에서 AP1으로 이동하는 환경을 보여준다.

본 논문의 이동성 지원 플랫폼 구현과 테스트를 위한 가정과 특징은 다음과 같다.

- 하나의 단일 SDN 도메인에서의 MN의 이동(AP의 변경)만 고려한다. 즉 MN의 IP 주소 변경에 대한 구현은 수행하지 않는다.
- 두 개 이상의 무선 인터페이스를 가진 MN의 이동을 지원한다. 다만 구현 복잡성 등의 문제로 멀티 인터페이스의 구현은 서로 다른 대역폭(2.4 GHz와 5 GHz)의 Wifi 인터페이스로 대체하였다.
- SDN 기반의 이동성 지원이 수행됨에 따라, Controller는 MN의 이동을 감지하면, 오픈플로우 프로토콜에 의해 AP1과 AP2의 flow 테이블을 즉시 변경함으로써, MN 이동에 따른 전송 손실을 최소화할 수 있다.

그림 5에 나타낸 각 시스템들의 S/W 모듈(3장에서 설명한)들은 C언어를 사용하여 구현하였고, SDN 기능을 수행하는 Controller/Switch S/W 구현은, 앞서 밝힌 바와 같이, Libfluid 오픈소스 라이브러리를 사용하였으며, 무선 자원정보 획득 기능들은 Netlink의 libnl 라이브러리를 사용하여 구현하였다.

2. 이동에 따른 데이터 전송 시험 및 결과

논문에서 구현한 SDN 기반 플랫폼의 이동성 지원 기능과 전송 효율을 시험하기 위해, MN 이동 시에 수행되는 데이터 전송에 관한 여러 값들을 측정하였다. 시험을 위해서 iperf 툴을 사용하였는데, iperf는 지정된 시간 동안 임의적인 인터넷 트래픽을 발생하여 측정 구간 사이의 네트워크 대역폭, 지터 등을 측정하는 용도도 많이 사용된다. 데이터 전송 시험은 그림 5의 Mobile node에서 Corresponding node 간에 수행되며, 시험 기간 중에 Mobile node는 AP1과 AP2를 교대로 접속하여 이동을 수행하게 된다. 노드 이동에 따른 데이터 전송 시험 파라미터들은 표 4에 요약하였다.

표 4. MN 이동에 따른 데이터 전송 시험 파라미터
Table 4. Data transfer test parameters according to MN movement

시험 프로그램	iperf
데이터 전송	총 60초간 512 바이트 데이터를 평균 1.05 Mbps 대역폭으로 전송
이동 횟수	1 ~ 5회 이동(MN의 AP 변경)
사용 인터페이스 수	1개/2개 비교 실험
측정값 산출	동일 시험 20회 수행 후, 평균값 계산

표 5는 MN의 이동횟수와 사용 인터페이스 수에 따라 측정된 평균 대역폭, 전송 패킷 수, 지터, 패킷 손실률 값들을 보여준다. 표 5를 보면, 대체적으로 이동횟수가 많아질수록 대역폭과 전송 패킷 수가 감소하고, 단일 인터페이스를 사용한 경우보다 두 개의 인터페이스를 사용한 경우가 대역폭과 전송 패킷 수가 증가함을 알 수 있다.

현재의 구현에서 다중 인터페이스를 사용하는 경우에는 기존 인터페이스의 AP 접속 해제 이전에 새로운 인터페이스로 이동할 AP 접속을 먼저 수행하고, 데이터 전송 경로를 새로운 AP로 변경하기 때문에, 이동에 의한 패킷 손실을 최대한 줄이는 것으로 판단된다.

표 5. 이동 시험에 의한 데이터 전송 관련 측정 값
Table 5. Data transmission measures by moving test

이동 횟수	인터페이스 수	대역폭 (KBps)	전송 패킷 수	지터 (ms)	패킷 손실률 (%)
1	1	988.5	15052	2.46	4.02
	2	1027.5	15040	0.51	1.89
2	1	959.0	14636	2.30	3.99
	2	1028.0	15054	0.58	2.00
3	1	949.68	14381	0.54	3.35
	2	980.16	14702	0.69	2.33
4	1	924.95	14184	1.50	4.71
	2	979.80	14648	1.83	2.44
5	1	903.00	13997	4.37	5.26
	2	963.28	14311	0.66	1.44

다음 그림 6과 7은 표 5의 이동횟수에 따른 대역폭과 패킷 전송 수를 단일/다중 인터페이스별로 표현한 그래프이다.

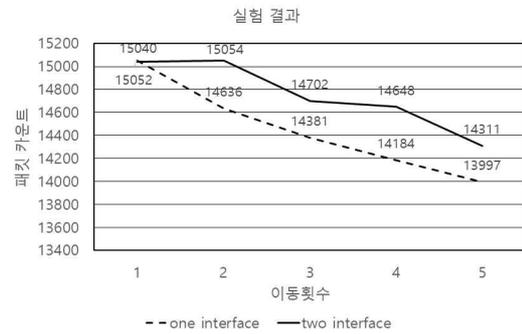


그림 6. 이동횟수에 따른 단일/다중 인터페이스 실험 시에 총 전송 패킷 수
Figure 6. Total number of packets transmitted in single/multi interface test according to the number of movements

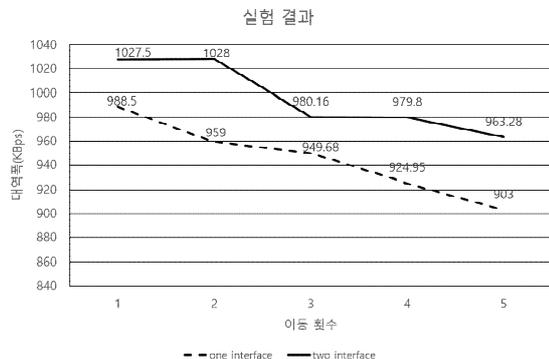


그림 7. 이동횟수에 따른 단일/다중 인터페이스 실험 시에 대역폭
Figure 7. Bandwidth in single/multi interface test according to the number of movements

V. 결 론

본 논문에서는 SDN의 오픈플로우 프로토콜을 수정하여 노드의 이동성을 제공하는 SDN 기반 네트워크 플랫폼을 설계하였다. 그리고 설계한 플랫폼을 오픈소스 형태의 libfluid의 Controller와 Switch를 결합하여 단일 SDN 플랫폼 상에서 단말의 이동성을 제공하는 기능을 구현하였다. 또한 노드의 멀티 인터페이스를 이용하여 노드가 이동할 때마다 발생하는 패킷 손실을 최소화하는 방안도 함께 제안하였다. 이러한 동작 시험 결과에 따라, 이동성을 제공하지 않는 기존 SDN 네트워크가 단말들의 이동성을 제공하는 것을 확인하였고, 멀티 인터페이스를 이용하여 하나의 인터페이스로 이동했을 시에 발생하는 패킷 손실률보다 훨씬 낮은 패킷 손실률을 보이는 것을 확인하였다.

다만, 본 논문의 이동성 지원은 단일 SDN 플랫폼 상에서의 이동만을 대상으로 하였는데, 향후에는 서로 다른 SDN 도메인 간의 노드 이동에 따라 IP 주소가 바뀌는 상황에서의 이동성 지원을 위한 연구가 수행되어야 할 것으로 생각된다. 또한 노드 이동의 다중 인터페이스 지원에서도 구현의 복잡성 때문에 물리적으로 다른 인터페이스를 사용하지 못하고, 서로 다른 대역의 Wifi 인터페이스 형태로 구현하였는데, 향후 연구에서는 4G/5G와 같은 이동통신을 포함한 다중 인터페이스에 의한 이동성 구현도 포함해야 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. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *Communication Magazine*, Vol. 51, No. 7, pp. 44 - 53, 2013.

[3] 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.

[4] Global mobile data traffic from 2016 to 2021 (in exabytes per month) Internet: [https://www.](https://www.statista.com/statistics/271405/global-mobile-data-traffic-forecast/)

[statista.com/statistics/271405/global-mobile-data-traffic-forecast/](https://www.statista.com/statistics/271405/global-mobile-data-traffic-forecast/), Aug. 10, 2018

[5] 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.

[6] ATTOResearch, "What is OpenFlow?," Internet: <http://www.atto-research.com/ko/atto-live-sdn-story/what-is-openflow/>, Jul. 28, 2016.

[7] 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.

[8] 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.

[9] Sung M. Kim, Hyon Y. Choi, Phil W. Park, Sung G. Min, and Youn H. Han, "OpenFlow-based Proxy Mobile IPv6 over Software Defined Network (SDN)," *IEEE CCNC: Mobility Management in the Networks of the Future World*, 2014.

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

[11] "Netlink Protocol Library Suite (libnl)" Internet: <https://www.infradead.org/~tgr/libnl/>, Jan. 2018.

[12] Jung Su Kim and Mon Ho Lee, "5G Mobile Communications: 4th Industrial Aorta," *The Journal of the Convergence on Culture Technology (JCCT)* Vol. 4 pp. 337-351, Feb. 2018.

※ 이 논문은 부산대학교 기본연구지원 사업(2년)에 의하여 연구되었음