한국정보전자통신기술학회논문지(jkiiect)'20-2, Vol.13 No.1
논문 20-13-01-88 http://dx.doi.org/10.17661/jkiiect.2020.13.1.88

시뮬레이션 환경 구축을 통한 소프트웨어-정의 네트워크에서 흐름 분석에 관한 연구

이동윤*

A Study on the Flow Analysis on the Software-Defined Networks through Simulation Environment Establishment

Dong-Yoon Lee*

요 약 최근 SDN 기술이 실제 통신 사업에 적용되면서 사용자가 많아지며 네트워크에 흐르는 데이터 량이 많아짐에 따라 네트워크 데이터 흐름 관리에 대한 관심이 늘고 있다. 이 과정에서 전송되는 네트워크 상의 데이터의 기밀성, 무결성, 가용성, 추적 가능성이 보장되는지 확인할 수 있어야 한다. 또한, 다양한 분야에서 요구되는 네트워크상에서 데이터를 실시간으로 흐름을 관측하고 통제를 시각적으로 확인할 수 있는 환경이 개발이 필요하다. 본 논문에서는 첫 번째로 Mininet을 응용하여 네트워크 토폴로지를 시각적으로 구성하고 다양한 속성을 부여할 수 있는 환경을 구축하였다. 둘째, Mininet 환경에서 OpenDayLight를 추가하여 네트워크 토폴로지에서 네트워크 트래픽 흐름을 시각적으로 확인하고 제어할 수 있는 시뮬레이션 환경을 개발하였다.

Abstract Recently, SDN technology is applied to real communication business, users are getting bigger, and as the amount of data flowing in the network increases, interest in network data flow management is increasing. During this process, it must be ensured that the confidentiality, integrity, availability, and traceability of the data on the network being transmitted. In addition, it is necessary to develop an environment for observing the flow of data in real time on a network required in various fields and visually confirming the control. In this paper, first, Mininet is applied to construct a network topology and various environment attributes. Second, we added OpenDayLight in Mininet environment to develop a simulation environment to visually check and control network traffic flow in network topology.

Key Words : SDN, Network Traffic Flow Management, Mininet, OpenDayLight

1. 서론

기존 네트워크는 클라우드, 분산시스템, 빅데이터 기술의 활용 증가에 따라 North-South 간의 네트워크 데이터 트래픽보다는 East-West간의 네트워크 데이터 트래픽을 급증시키는 상황이다. SDE(Software Defined Environment)는 컴퓨팅과 작업 부하를 전체적으로 제어할 수 있도록 프로그래밍 가능한 새로운 형식의 컨트롤 플레인을 제공한다. SDN은 스위치, 라우터와 같은 네트워크 장비 내에 해당 단말의 동작을 제어하는 기능과 데이터를 전송하는 기능을 모두 포함하는 기존 방식과 차별성을 가지고 있다[1].

최근에는 인터넷 접속 가능한 모바일 기기가 기하급수적으로 증가하고 인터넷 기반으로 디바이스 간 통신의 필요성이 증가에 따라, 이를 지원하는 모바일과 IoT 융합 네트워크 인프라 기술을 확보하여 서비스 확장의

*Department of Electrical & Electronic Engineering, Joongbu University(dylee@joongbu.ac.kr)
Received January 31, 2020 Revised February 25, 2020 Accepted February 25, 2020

기반을 마련하고 디바이스 간에 인터넷을 통해 다양한 시장에서 레거시 기반의 서비스는 물론 새로운 서비스를 생성할 수 있도록 2022년경 까지는 5G 백본망은 SDN을 기반으로 성숙한 기술력을 갖추어 해외 시장에서 경쟁력 확보할 수 있는 관련 연구가 필요하다[2]. 네트워크 가상화는 기존에 물리적으로 연결된 회선에 논리적으로 새로운 네트워크 회선을 구성하여 양자간 통신하는 방법으로서 네트워크 가상화[3] 방법 중 본 논문에서는 최근 SDN 기술이 실제 통신 사업에 적용되고 사용자가 많아지며 네트워크에 흐르는 데이터량이 많아짐에 따라 데이터 보안에 대한 관심이 늘고 있다. 특히, SDN은 기존의 네트워크와는 달리 콘트롤 플레이어와 데이터 플레인으로 두 개의 레이어로 분리되어 있다. 이 과정에서 전송되는 네트워크상의 데이터들의 기밀성, 무결성, 가용성, 추적 가능성이 보장되는지 확인할 수 있어야 하며 다양한 분야에서 요구되는 네트워크상에서 데이터를 실시간으로 흐름을 관측하고 통제를 시각적으로 확인할 수 있는 환경을 개발하였다.

이를 위해, 첫 번째로 시뮬레이션 도구인 Mininet을 응용하여 네트워크 토폴로지를 관리자의 의도에 따라 시각적으로 구성하고 다양한 속성을 부여할 수 있는 환경을 구축하였다. 둘째, Mininet 환경에서 OpenDayLight를 추가하여 구성된 네트워크 토폴로지에서 네트워크 트래픽 흐름을 시각적으로 확인하고 제어할 수 있는 시뮬레이션 환경을 구축하였다. 이러한 결과는 SDN에서 네트워크 트래픽 흐름을 보다 쉽게 파악하고 트래픽의 흐름을 조절하는데 매우 유용하게 활용할 수 있다.

또한 실제로 네트워크 토폴로지를 구성하기 전에 다양한 토폴로지 구성을 실험할 수 있으며 이를 통해, SDN의 충분한 특성을 활용하고 실제적인 검증을 시각적으로 확인할 수 있다.

본 논문의 구성은 2장에서 배경 연구로 오픈웨어인 Mininet과 OpenDayLight에 대해 소개한다. 3장에서는 Mininet을 활용하여 네트워크 토폴로지를 쉽게 구성할 수 있는 환경 구축 결과와 OpenDayLight를 기반으로 네트워크 트래픽을 모니터링 할 수 있는 개발 결과를 제시한다. 4장에서는 결론을 기술한다.

2. 연구배경 및 관련연구

OpenDayLight[4]는 2013년 4월 리눅스 파운데이션의 오픈소스 프레임워크이다. OpenDayLight 컨트롤러 플랫폼은 모듈 형태 구조에 따라 관련 네트워킹 기능들을 모듈 형태로 추가할 수 있는 장점을 가지고 있다. 응용 요구사항에 따라 웹기반 사용자 인터페이스 접근을 위한 DLUX UI, 고확장성/고가용성을 위한 클러스터링 모듈, 오픈플로우 스위치 모듈, 클라우드와의 연동을 위한 오픈스택 서비스, 멀티테넌트 가상 네트워크를 위한 VTN(Virtual Tenant Networking) 서비스, NFV를 위한 서비스 기능 체이닝(Service Function Chaining) 서비스 등을 자유로이 확장할 수 있다. OpenDayLight는 응용 계층을 위해 OSGI 프레임워크 및 노스바운드 API를 지원하며, 응용 계층에서는 컨트롤러를 통하여 네트워크 정보를 수집/모니터링하거나 네트워크 전반에 대한 제어관리 알고리즘 및 비즈니스 로직을 적용할 수 있다.

Mininet[5]은 개인 PC나 노트북에서 쉽고 빠르게, 그리고 가상 OpenFlow 네트워크를 구성하여 테스트할 수 있는 BSD 오픈소스 프로젝트이다. 또한 리눅스 커널에 수백개의 가상 호스트, 스위치, 컨트롤러, 링크 등의 실제와 같은 수준의 가상 네트워크를 생성할 수 있는 네트워크 에뮬레이터이다[6,7]. 가상 호스트는 리눅스의 소프트웨어로 실행되며 가상화를 통해 호스트별로 리눅스 프로그램을 실행할 수 있다. 가상 스위치는 Open vSwitch와 연동하여 하나의 가상스위치는 Open vSwitch의 하나의 브리지로 생성되며, 유연한 맞춤형 라우팅과 SDN을 위한 OpenFlow를 제공한다[8]. 또한 스위치와 호스트는 가상 이더넷(veth) 페어를 사용하여 연결되며, 실제 네트워크와 연결을 할 수 있다[9]. Mininet은 네트워크 네임스페이스와 별도로 네트워크 인터페이스, 개별 프로세스를 제공하는 경량 가상화 기능, 라우팅 테이블, ARP 테이블 등을 지원한다.

3. SDN 흐름 분석 시뮬레이션 환경

3.1 시뮬레이션 환경 구축 및 검증

SDN은 네트워크 트래픽을 소프트웨어적으로 제어하고 관리를 간결하게 하기 위한 네트워크 가상화 기술 중 하나이다. 기존방식은 네트워크에 다양하게 연결된 디바이스에 문제가 생기면 사람이 하드웨어 또는 소프트웨어적인 문제를 해결하는 것이었는데, SDN은 여러 가지 이벤트의 예방 및 관리가 가능하므로 경제적, 효율적인 네트워크 망 관리가 가능한 장점을 가지고 있다. SDN을 이용하게 되면 하나의 메인 컨트롤러로 전체 네트워크망을 관리하게 된다. 네트워크에서 문제가 발생한 경우, 컨트롤러가 해당 장치의 문제가 해결될 때까지 장치를 네트워크에서 제외하고 문제 진단 및 해결 후에 자동으로 네트워크에 재연결할 수 있다. 또한 네트워크 트래픽이 한 경로로 집중되어 있다면 통신이 끝날 때까지 대기 한다거나 통신을 취소하는 기존 방식과 달리 SDN에서는 트래픽이 여유로운 회선으로 연결을 소프트웨어적으로 재설정하여 통신이 가능하게 할 수 있다.

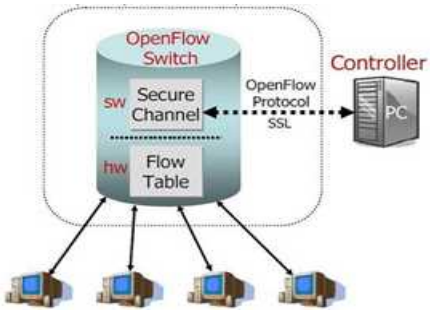

그림 1. 오픈플로우 스위치 연결 구조
Fig. 1. OpenFlow Switch Connection Structure

SDN은 기본적으로 스타형 또는 망형의 모습을 가진 토폴로지를 갖춤으로써 네트워크망을 관리하며 네트워크를 관리하는 운영체제와 같은 역할을 수행한다. SDN은 API를 제공하여 공통된 네트워크 서비스(라우팅, 멀티캐스트, 보안, 접근제어, 대역폭 관리, 트래픽 관리 등)를 구현할 수 있다. 이 경우, 네트워크 장치가 제공하는 오픈 인터페이스의 대표적인 기술이 OpenFlow이다[10,11]. 본 논문에서도 기본적으로 SDN 흐름 분석 시뮬레이션 환경 분석 구축시에 첫 번째로 그림 1의 구조를 기반으로 하였으며, 상세 스펙은 시뮬레이션 과정에 상세 변경하여 실험을 진행하였다. 또한 거대 네트워크 망의 구현은 현실적으로 비용과 시간을 고려하면 불가능하기 때문에 OpenFlow 기술을 기반으로 하는 시뮬레이터가 필요하다. 시뮬레이터중 Add-On을 추가할 수 있는 Mininet은 OpenSource 시뮬레이터를 이용한 네트워크 흐름 관측 및 통제를 시각적으로 확인할 수 있는 환경을 본 연구에서 구현하였다.

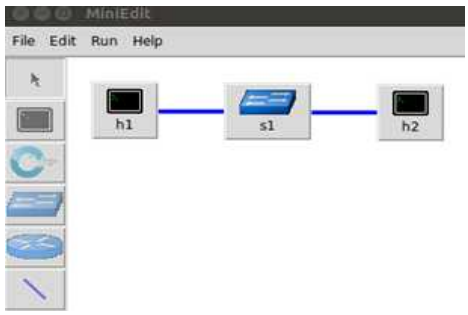

그림 2. Mininet을 이용한 단순 토폴로지 구성
Fig. 2. Simple Topology Construction using Mininet

Mininet은 전체 OpenFlow 네트워크를 시뮬레이션하는 오픈웨어로서 여러 네트워크 토폴로지 환경을 구축하고 여러 종류의 프로토콜을 적용하여 실험할 수 있다. OpenFlow 1.0을 지원하지만 다른 OpenFlow 프로토콜을 지원하도록 수정 가능하다. 실제로 본 연구에서는 그림 2와 같은 단순한 네트워크 토폴로지의 구성과 그림 3과 같이 심화된 네트워크 토폴로지 실험환경을 구축하였다.

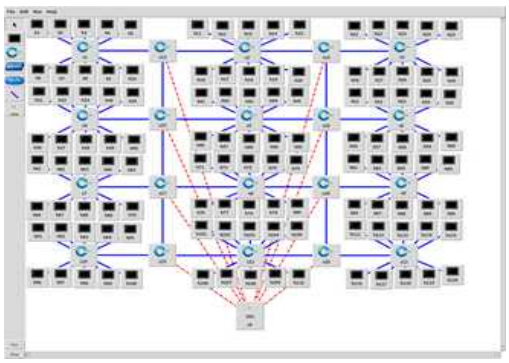

그림 3. Mininet을 이용한 심화된 토폴로지 구성
Fig. 3. Advanced Topology Construction using Mininet

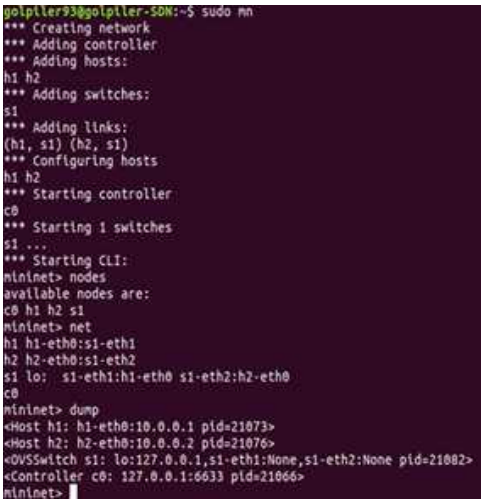

그림 4. 텍스트 모드를 통한 토폴로지 구성
Fig. 4. Topology Construction through Text-mode

3.2 네트워크 흐름 측정 환경 구축

Mininet은 토폴로지를 구현하고 연결 경로를 보여줄 수 있지만 각 노드마다 터미널을 실행하여 네트워크 데이터 흐름을 테스트해야 한다. 즉, 각 노드 별 상태를 자동으로 나타내고 있지 않기 때문에 이에 대해선 추가 프로그램이 필요하므로 Add-On들이 필요하게 되는데 이 중 OpenDayLight(ODL)의 Yang UI 모듈을 적용하여 GUI 형식으로 토폴로지를 나타나게 하였다. 더불어 각 노드의 속성과 데이터 흐름의 속성도 파악 가능하다. 실제로 본 논문에서 구성한 구체적인 시뮬레이션 환경의 구성은 그림 5와 같다.

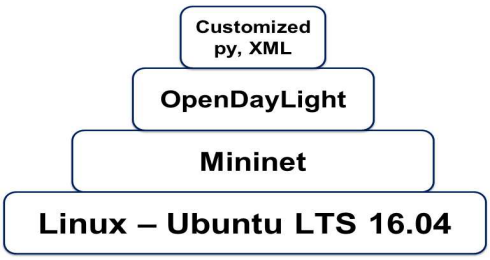

그림 5. ODL을 이용한 네트워크 흐름 측정 환경
Fig. 5. Network Flow Estimation Environment using OpenDayLight

Mininet과 OpenDayLight는 리눅스 우분투 16.04를 기반으로 Openjdk-8-jdk와 jre를 설치 후 OpenDayLight 공식 사이트에서 pre-built된 karaf 파일을 이용하였다. 우분투에서 ./압축이 해제된 폴더/bin/karaf를 실행하면 feature들을 설치할 수 있는 환경을 확인할 수 있으며, 여기서 네트워크 토폴로지 시각화를 위한 YangUI등과 네트워크 흐름을 확인 할 수 있는 Add-On들을 설치하였다. 그 후 브라우저를 구동 후 localhost:8181/index.html을 수행하면 아래 그림 6과 같은 OpenDayLight GUI구성에 대한 실험 결과를 확인할 수 있다.

GUI 결과에서 Topology는 ODL과 Mininet VM을 함께 사용함으로써 구성된 토폴로지를 확인할 수 있으며 Yang 관련 UI를 사용하면 PreBuilt된 토폴로지들의 정보의 확인도 가능하다. 또한 각 노드 간 네트워크 흐름들의 속도설정과 관측이 가능하고 테이블 형태로 추출하도록 명령어를 스크립트를 활용할 수 있다. 이를 이용하면 특정 네트워크 트래픽을 관찰한 후 CSV 파일로 네트워크 흐름을 조절하여 트래픽 흐름을 개선할 수 있다. 특정 노드에 네트워크 흐름 및 모니터링이 집중되지 않고 전체적인 노드에 균형을 맞추어 네트워크 흐름을 조절하는 CSV 파일의 크기를 설정 또는 수정하여 반복적으로 적용하였다.

그림 6. ODL을 이용한 GUI 배경
Fig. 6. GUI Environment using OpenDayLight

3.3 네트워크 흐름 측정 실험 결과 분석

본 논문에서는 SDN 시뮬레이터 중 Mininet을 이용하여 SDN의 통신 흐름 상태를 가시적으로 확인 가능한 환경을 구축한 후 대시보드에 출력되어 있는 정보를 따로 가공 할 수 없었다. 이에 대시보드를 불러오는 ODL를 이용하지 않는 대신에 클라이언트와 Mininet VM 서버와의 직접 연결을 클라이언트 측에서 트래픽을 수

집하여 이를 시각적으로 출력하는 환경으로 재구성 하였다. 즉, Mininet VM과 통신하는 SDN 모델의 트래픽 패킷의 네트워크 로그를 클라이언트측의 수집 프로그램을 개발하여 네트워크 로그에 대한 그래프 형태의 시각적인 정보를 출력하였다. 이를 위해, Mininet VM 서버와 클라이언트를 Virtual Net으로 서로를 연결시킨다. 서버와 클라이언트가 연결되면 클라이언트측에서 패킷 정보 수집 프로그램인 WireShark를 실행시켜 가상 네트워크에서 통신 흐름 정보를 수집하고 기록을 저장한다. 본 논문에서는 데이터 처리에 좀 더 편리하도록 하기 위해 CSV 파일의 형태로 저장하였다. 본 논문에서는 네트워크에서 흐름 정보를 수집한 후 1분간의 데이터, 로그 2075 라인을 이용하였다.

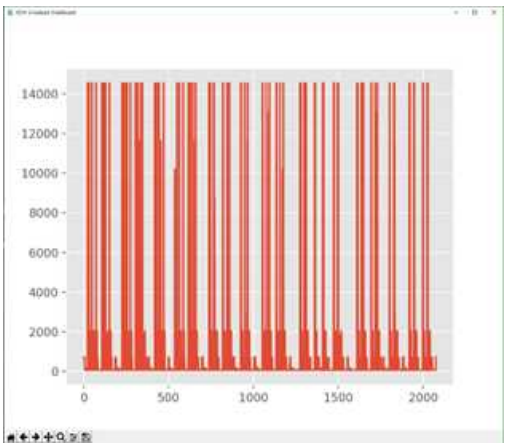

그림 7. 클라이언트 서버간 트래픽 측정(2075)
Fig. 7. Client to Server traffic Estimation(2075)

실제로 그림 7은 네트워크 로그 라인수가 2075개(가로축)의 전체에 대한 결과이며 모수가 더욱 많아질 경우 그래프의 가독성이 떨어지므로 1분간의 데이터만 채택하였다.

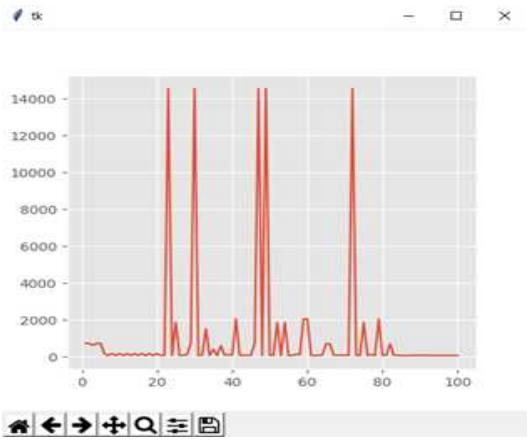

그림 8. 클라이언트 서버간 트래픽 측정(100)
Fig. 8. Client to Server traffic Estimation(100)

또한, 관리자의 가독성을 높이기 위해 모수를 2075에서 100으로 조정한 결과는 그림 8과 같다.

로그 수를 100으로 줄임으로써(줌-인) 보다 상세하게 트래픽의 증감(세로축)을 상세하게 확인할 수 있다. 실제로 실험 과정에서는 로그 수의 조정을 통해 SDN 네트워크의 특정 노드에서 트래픽이 발생되는 상황을 시각적으로 확인할 수 있는 의미를 가지고 있으며, 로그 수의 조정을 통해 상세 트래픽의 가독성을 조절할 수 있다.

본 논문에서는 기존 네트워크 토폴로지 시뮬레이터의 성능보다는 ODL에서 지원하는 모듈을 적용하고 시각적 기능을 개선하였다.

따라서 시뮬레이터의 성능은 기존 Mininet의 성능과 기본적인 기능을 지원하고 있으며, 기능적인 면에서 SDN 토폴로지를 시각적으로 보다 쉽게 확인할 수 있고, 각 노드의 상태를 확인할 수 있는 장점을 가지고 있다.

4. 결론 및 향후 과제

본 논문에서는 첫 번째로 Mininet을 응용하여 네트워크 토포로지를 관리자의 의도에 따라 시각적으로 구성하고 다양한 속성을 부여할 수 있는 환경을 구축하였다. 둘째, Mininet 환경에서 OpenDayLight를 추가하여 구성된 네트워크 토폴로지에서 네트워크 트래픽 흐름을 시각적으로 확인하고 제어할 수 있는 시뮬레이션

환경을 개발하였다. 이러한 연구 결과는 SDN에서 네트워크 트래픽 흐름을 보다 쉽게 파악하고 트래픽의 흐름을 조절하는데 매우 유용하게 활용할 수 있다.

향후에는 로그 수의 조정에 의한 가독성 조정에 추가하여 특정 시각 특정 다중 노드 구간을 설정하여 네트워크 토폴로지 변경에 따른 시각화 기능을 추가하는 연구를 진행할 예정이다.

REFERENCES

[1] Dixit A., Hao F., Mukherjee S., Lakshman T. V., and Kompella, R., "Towards an elastic distributed SDN controller", SIGCOMM Computer Communication Reveview, vol. 43, No. 44, pp. 7-12, 2013.

[2] Scott-Hayward S., Natarajan S., and Sezer S., "A survey of security in software defined networks", IEEE Communications Surveys & Tutorials, Vol. 18, No. 1, pp. 623-654, 2016.

[3] YunHee Kang, Younhoon Park, and KwngMan Ko, "Adverstise based Adaptive Model for IoT device in Network Virtualization Environment", CUTE 2017, pp.22-28, 2017.

[4] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a Model-Driven SDN Controller architecture," in Proceeding of IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 1-6., 2014.

[5] Mininet, http://mininet.org/, 2018, May.

[6] D. A. Patterson and J. L. Hennessy, Computer Organization and Design MIPS Edition: The Hardware/Software Interface, 5th ed. New York, NY, USA: Elsevier, 2014, p. 800.

[7] P. Megyesi, A. Botta, G. Aceto, A. Pescapé, and S. Molnár, ``Challenges and solution for measuring available bandwidth in software dened networks,'' Comput. Commun., vol. 99, pp. 4861, Feb. 2017.

[8] K. Yi and Y. H. Ding, ``32-bit RISC CPU based on MIPS instruction fetch module design,'' in Proc. Int. Joint Conf. Artif. Intell., 2009, pp. 754760.

[9] A. Botta, A. Dainotti, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios,'' Comput. Netw., vol. 56, no. 15, pp. 35313547, Oct. 2012.

[10] M. Karakus and A. Durresi, ``A survey: Control plane scalability issues and approaches in software-dened networking (SDN),'' Comput. Netw., vol. 112, pp. 279293, Jan. 2017.

[11] C. L. Lim, A. Moffat, and A. Wirth, ``Lazy and eager approaches for the set cover problem,'' presented at the 37th Austral. Comput. Sci. Conf., vol. 147, Auckland, New Zealand, 2014.

저자약력

이 동 윤 (Dong-Yoon Lee) **[정회원]**

- 1990년 2월 : 연세대학교 전기공학과 (공학석사)
- 2001년 2월 : 연세대학교 전기전자공학과 (공학박사)
- 2001년 3월 ~ 2002년 2월 : 원광대학교 BK21교수
- 2002년 3월 ~ 현재 : 중부대학교 전기전자공학과 교수

〈관심분야〉 IT융합, SDN응용 등