

# 미래 네트워크의 새로운 패러다임 SDN/NFV에 대하여

백상현, 장인선, 서동은, 이종화\*  
고려대학교, 한국전자통신연구원\*

## 요약

소프트웨어 정의 네트워킹 (Software-Defined Networking: SDN)과 네트워크 기능 가상화 (Network Function Virtualization: NFV) 기술은 네트워크 산업 전반에 걸쳐 가장 중요한 기술/트렌드로 인식되고 있다. 현재 국내외 주요 기업들이 앞다투어 기술 개발을 추진하고 있으며 이를 실제 상용망에 적용하고자 하는 움직임도 활발히 진행되고 있다. 본 고에서는 유사하면서도 다른 SDN과 NFV의 개념을 명확히 살펴보고자 한다. SDN과 NFV의 역사, 핵심 기술, 활용 분야, 그리고 향후 연구 이슈를 폭넓게 제시한다.

## I. 서론

2007년 8월, 스탠포드 대학의 박사과정 학생이던 Martin Casado는 일본 동경에서 열린 ACM SIGCOMM 2007에서 "Ethane: Taking Control of the Enterprise" [21]라는 제목의 논문을 발표한다. 엔터프라이즈 네트워크의 관리를 보다 용이하게 하고 보안을 강화할 수 있는 방안을 제안한 본 논문이 향후 소프트웨어 정의 네트워킹 (Software-Defined Networking: SDN) 연구의 시발점이 될 것이라는 것은 Marin Casado 자신도 몰랐을 것이다. 향후 Martin Casado는 네트워크 가상화와 SDN 기술을 바탕으로 Nicira라는 회사를 창업하였고 Nicira는 10억 달러가 넘는 금액에 대표적인 가상화 솔루션 업체인 VM웨어에 인수되었다. 현재 Martin Casado는 VM웨어의 네트워크&보안 사업부문 사장을 맡고 있으며 실리콘 벨리에서 스타트업 기업 경영인의 대표적인 성공사례로 꼽히고 있다.

한편 네트워크 기능 가상화 (Network Function Virtualization: NFV)는 가상화 기술의 발전을 바탕으로 하여 네트워크 상에 존재하는 다양한 네트워크 기능 또는 미들박스를 소프트웨어 형태로 가상화하여 제공하는 기술이다. NFV는 가상화에 기반한 기술로 SDN과는 다른 개념이지만 SDN과 결

합될 때 더 큰 시너지 효과를 얻을 수 있다. 따라서, 본 고에서는 SDN과 NFV의 등장 배경, 주요 기술, 응용 시나리오를 폭넓게 살펴본다.

## II. SDN 이란?

지금과 같은 IP 기반의 네트워크가 대세가 되기 이전인 90년대에 ATM (Asynchronous Transfer Mode)환경에서 네트워크를 소프트웨어를 통해 프로그래밍하고 제어하는 개념이 영국 케임브리지 대학의 Tempest 프로젝트를 통해 소개되었다 [1]. Tempest 프로젝트에서는 ATM 스위치를 논리적으로 분할하여 외부에서 동적으로 프로그래밍할 수 있는 개념을 소개하였다.

유사한 개념을 IP망에서도 실현하기 위한 연구가 진행되어 IETF (Internet Engineering Task Force)에서는 ForCES (Forwarding and Control Element Separation)이라는 워킹 그룹을 구성하여 데이터 전달 평면과 제어 평면을 분리하고 제어 평면이 별도의 시그널링 프로토콜 [2]을 통해 데이터 전달 평면을 제어할 수 있는 방안을 제안하였다. 하지만 이러한 ForCES의 초기 규격은 네트워크 장치 내부에서의 데이터 전달 평면과 제어 평면 분리 구조를 정의하고 있으며 보안 측면에서도 취약점이 존재한다. 이러한 한계를 극복하기 위해서 IETF에서는 NETCONF (Network Configuration) 워킹 그룹을 통해 NETCONF [3]라는 개선된 프로토콜을 표준화하였다. 하지만 여전히 NETCONF는 네트워크 관리 기능에 초점을 맞추고 있고 유연한 트래픽 제어와 이를 바탕으로 한 부가적인 서비스 제공 측면에서는 한계가 있다.

현재 우리가 이야기하는 SDN의 기본적인 개념은 미국 과학재단 (NSF)의 미래 인터넷 연구 프로그램인 FIND (Future Internet Design) [4]가 시작되면서 소개되기 시작하였다. FIND 프로그램의 일부로 스탠포드 대학과 버클리 대학에서 SANE (clean-slate Security Architecture for Enterprise Network) [5]이라는 프로젝트가 수행되었는데 SANE에서는

VLAN, ACL, 방화벽, NAT 등의 네트워크 기능을 도메인 컨트롤러를 통해 중앙 집중화된 방식으로 관리하는 개념을 제안하였다. 이를 발전시켜 엔터프라이즈 환경에 보다 적합한 기법을 설계한 것이 앞서 언급한 Ethane 프로젝트 [21]이다. Ethane 프로젝트의 성공을 바탕으로 스탠포드 대학에서 네트워크 장비를 제어하기 위한 OpenFlow 프로토콜 [6]을 개발하였고 많은 회사가 참여하여 OpenFlow를 지원하는 시제품을 출시하게 되었다. 그리고 2008년에 Martin Casado가 창업한 Nicira에서 OpenFlow 컨트롤러인 NOX [7]를 오픈소스로 공개하였다. 보다 체계적인 OpenFlow 기술 개발과 상용화, 규격 정리를 위해 2011년 3월에 ONF (Open Networking Foundation)이 설립되었고 그 이후에는 IETF, IRTF, ITU-T 등의 다양한 표준 기구에서 SDN이라는 키워드를 바탕으로 표준 개발에 힘을 쏟고 있는 상태이다.

SDN은 기존의 라우터나 스위치와 같이 소프트웨어 기반의 제어 평면 (예: 라우팅 프로토콜)과 하드웨어 기반의 데이터 전달 평면 (즉, 패킷 포워딩)이 밀접하게 결합되어 있는 네트워크 장비의 제어 평면을 데이터 전달 평면으로부터 분리하는 개념이다. 구체적으로, 제어 평면을 중앙 SDN 컨트롤러를 통해 중앙 집중화하여 데이터 전달 평면의 네트워크 장비를 제어하며, 네트워크 장비의 기능들을 정의할 수 있는 Application Program Interface (API)를 외부에 개방하여 다양한 네트워크 제어 기술과 라우팅 프로토콜을 개발하여 동작할 수 있도록 한다. <그림 1>은 SDN의 기본 구조 [8]를 도식화 한 것이다.

SDN 구조는 크게 데이터 전달 평면 (Data Plane), 제어 평면 (Control Plane), 응용 평면 (Application Plane)으로 구성된

다. 먼저, 가장 중요한 제어 평면에서는 SDN 컨트롤러를 이용하여 데이터 전달 평면의 네트워크 장비들을 제어하고 관리한다. 특히, SDN 컨트롤러는 수많은 네트워크 장비들을 하나하나 수작업으로 제어하고 관리하는 것이 아닌, 전체 네트워크 망 상태에 대한 뷰 (즉, 추상화된 형태의 네트워크)를 갖고, 프로그래밍을 통해 네트워크 망의 동작을 제어할 수 있다. 예를 들어, 전체 네트워크 상태를 보면서 토폴로지 변화 등의 이벤트에 따른 반응, 새로운 플로우에 대한 경로 변경 등을 할 수 있다. 이때, 분리된 제어 평면과 데이터 전달 평면을 상호 연결 시켜주는 표준화된 프로토콜 (즉, Open southbound API)가 필요하다. 현재 가장 많이 사용되고 있는 프로토콜은 스탠포드 대학에서 개발한 OpenFlow이며, 이 외에도 다양한 오픈 API가 개발되고 있다. 즉, 네트워크 관리자는 표준화된 인터페이스를 통해 시스코 등의 특정 벤더들에 의존하지 않고 네트워크 망을 제어하고 관리할 수 있으며, 네트워크 설계와 운용을 보다 단순화할 수 있다.

SDN은 또한 제어 평면과 응용 평면 간에 일련의 API (즉, Northbound API)를 제공한다. 이는 API를 이용하여 SDN 컨트롤러의 네트워크 OS 기능을 자유롭게 개발할 수 있다는 것을 의미한다. 즉, 네트워크 관리자는 API를 통해 자신의 목표에 맞는 라우팅 프로토콜, 트래픽 엔지니어링, QoS 관리 등을 제공할 수 있게 된다.

마지막으로, 데이터 전달 평면의 네트워크 하드웨어 장비는 기존의 라우터 스위치와는 달리 패킷 포워딩 기능만 수행하고, SDN 컨트롤러의 명령에 의해서 제어 된다. 그러므로, SDN에서의 스위치는 최소한의 기능만 갖고 컨트롤러에 의해 제어되는 단순화된 형태를 갖는다.

결국, SDN은 현재 네트워크가 가지고 있는 각 네트워크 장비와 라우팅 프로토콜 등의 네트워크 제어 프로토콜의 독립성을 지양하며, 네트워크의 지능이 SDN 컨트롤러에 집중되어 네트워크 전체를 관리하고, NOS와 API를 이용하여 네트워크를 프로그래밍 할 수 있는 것에 큰 의미가 있다. 특히, 중앙 SDN 컨트롤러를 이용하여 전체 네트워크 상태의 뷰, 즉 네트워크 추상화를 통해 유연하고, 효율적인 네트워크 관리가 가능하다는 것이 SDN의 큰 장점이다.

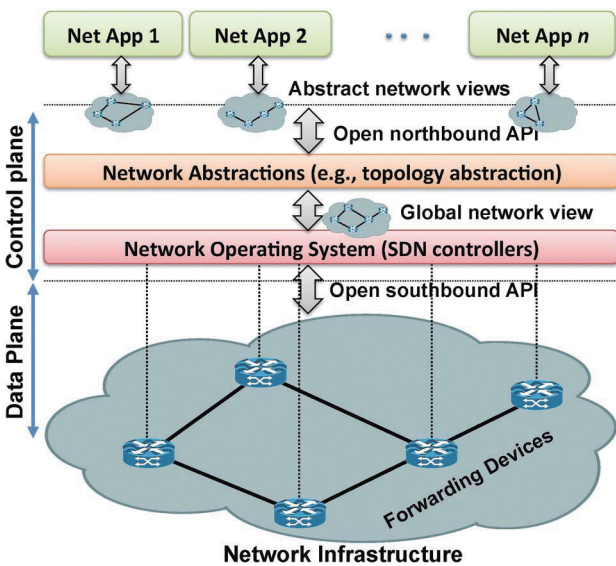


그림 1. SDN의 기본 구조 [8]

### III. SDN 핵심 기술

SDN 기술을 실현하기 위해서는 개방형 프로토콜을 기반으로 네트워크 장비들을 제어하고 다양한 네트워크 서비스들을 제공하는 SDN 컨트롤러, 고성능의 트래픽 전달 기능 및 개방형 인

터페이스를 제공하는 SDN 스위치, 그리고 SDN 컨트롤러와 네트워크 장비 간의 인터페이스 (즉, Southbound API) 및 네트워크 애플리케이션과 컨트롤러 간의 인터페이스 (즉, Northbound API)에 대한 정의가 필수적이다. 본 장에서는 이러한 SDN 주요 기술들에 대한 연구 및 개발 동향에 대하여 알아본다.

## 1. SDN 컨트롤러

기존의 네트워크 관리자는 하위 레벨의, 네트워크 장비 벤더에 의존적인 프로그래밍 언어를 수작업으로 입력하여, 분산적으로 네트워크를 관리 및 제어해야 하였다. 이러한 문제를 해결하고자 등장한 SDN 기술에서는 SDN 컨트롤러를 통해 중앙 집중 형식의 추상화된 네트워크 관리 및 제어 기능을 제공하여 복잡했던 기존의 네트워크 관리를 편리하게 하고, 네트워크 전체의 논리적인 뷰를 기반으로 네트워크 문제를 쉽게 풀 수 있게 한다.

SDN 기술의 등장 이래로 다양한 SDN 컨트롤러들이 오픈소스 커뮤니티를 기반으로 개발되었다. NOX [7]은 OpenFlow를 지원하는 최초의 SDN 컨트롤러로 연구 목적으로 개발되어 초기 SDN 컨트롤러 연구의 초석 역할을 하였다. NOX는 낮은 사용성 및 불편한 개발 환경 등의 단점을 보완한 다음 버전인 POX [9]를 기반으로 SDN 연구에 기여하고 있다. Floodlight [10]은 Big Switch Networks에서 주도하여 개발중인 오픈소스 SDN 컨트롤러로 가상의 혹은 물리적인 OpenFlow 스위치들의 제어 기능을 모두 지원하고 기업 수준의 컨트롤러 성능 제공을 목적으로 한다. 한편 NOX, POX, Floodlight 모두 OpenFlow 프로토콜 외의 기존의 네트워크 장비에서 사용하는 프로토콜들 (예: NETCONF, RESTCONF [11], SNMP [12])을 지원하지 못하는 문제점으로 인해 실제적인 대규모 네트워크에 적용하기에는 한계점이 존재한다.

최근 이러한 한계점을 극복하고 실제 대규모 네트워크에 적용 가능한 수준의 SDN 컨트롤러 개발을 목적으로 하는 오픈 커뮤니티 프로젝트로 ODL (OpenDaylight) [13] 프로젝트와 ONOS [14] 프로젝트가 등장하였다. 먼저 ODL은 리눅스 재단이 운영하는 오픈 소스 기반의 SDN 컨트롤러 개발 프로젝트로, 앞서 언급한 한계를 극복하고 실제 대규모 네트워크에 적용할 수 있는 수준의 SDN 컨트롤러 플랫폼 개발을 목표로 하여 진행 중에 있다. 현재 ODL 프로젝트에는 Cisco, Juniper, Ericsson, VMware와 같은 글로벌 IT 업체들이 개발에 참여하고 있으며 OpenFlow만을 지원하는 기존의 SDN 컨트롤러들과 달리, ODL의 경우 OpenFlow 뿐만 아니라 SNMP, LISP [15], NETCONF 등의 다양한 Southbound API를 지원한다는 장점이 있다. 또한 SDN 기술과 더불어 최근 각광받고 있는 NFV 기술과의 연동을 위해 클라우드 컴퓨팅 솔루션인 OpenStack

[16]의 Neutron API와의 연동을 지원한다.

ONOS의 경우, Stanford 대학과 UC Berkeley 대학의 연구진들이 설립한 ONLAB이 운영하는 오픈 소스 기반의 SDN 컨트롤러 개발 프로젝트로, 고가용성, 확장성을 제공하는 컨트롤러 개발을 목적으로 한다. 즉, 서비스 제공 기업들이 사용할 만한 수준의 고성능 SDN 컨트롤러 플랫폼을 지향한다. 또한 ONOS는 ODL과 마찬가지로 OpenFlow를 포함한 다양한 Southbound API 프로토콜을 지원한다. 현재 ONOS 프로젝트에는 AT&T, NTT Communications, SK 텔레콤 등의 통신 사업자와 Ciena, Cisco, Ericsson, Huawei 등의 벤더들이 개발에 참여하고 있다.

한편 국내의 SDN 플랫폼 개발 사례로는 ETRI 주도 하에 진행 중인 IRIS 프로젝트 [17]가 있다. IRIS는 네트워크 IO 엔진을 사용하는 이벤트 처리기를 기반으로 설계되었다. 구체적으로 IO 엔진 위에 Floodlight에서 지원하는 SDN 기본 기능들 (Learning Switch, Link Discovery, Topology Manager 등)을 탑재한 구조이다. IRIS는 확장성 및 고가용성 보장, 그리고 멀티도메인 환경을 지원하는 것을 목적으로 한다.

## 2. SDN 장치

오늘날의 네트워크 인프라에는 라우터 장비, 스위치 장비 (혹은 가상 스위치), 무선 액세스 포인트 등의 다양한 네트워크 장비들이 존재한다. 이러한 장비들은 패킷 포워딩 기능뿐만 아니라 라우팅, 웹 캐싱, DNS, 방화벽 등의 다양한 기능을 제공한다. 반면, SDN에서의 네트워크 장비는 패킷 포워딩 기능만을 제공하고 그 외의 모든 기능들은 SDN 컨트롤러에서 수행한다. 이를 위해 SDN 스위치에서는 컨트롤러에게 개방형 표준 인터페이스를 제공하고 SDN 컨트롤러는 이를 통해 SDN 스위치를 제어한다.

현재 출시된 하드웨어 기반의 SDN 스위치 제품들은 가장 잘 알려진 OpenFlow 프로토콜을 지원하는 스위치들이다. 이들은 기존의 스위치 및 라우터에서 제공하는 네트워크 장비 제어 프로토콜들 (혹은 인터페이스들)을 제공하지 않고 단지 OpenFlow 프로토콜을 사용하는 컨트롤러의 명령에 의해서만 작동하는 Pure OpenFlow 스위치 장비, OpenFlow 프로토콜 뿐만 아니라 기존의 네트워크 장비 제어 프로토콜들 (NETCONF, RESTCONF, SNMP 등)을 지원하는 Hybrid OpenFlow 스위치 장비로 나뉜다. 현 시점에서 상용화된 하드웨어 기반의 OpenFlow 스위치 장비들의 경우 대부분 Hybrid에 해당하며 Hewlett-Packard사의 8200zl, Brocade사의 5400zl, IBM사의 RackSwitch G8264, Juniper사의 Junos MX-Series 등이 있다.



한편, 하드웨어에 무관하게 사용할 수 있는 소프트웨어 기반의 SDN 스위치 기술들 또한 개발되었다. 이러한 기술들은 연구 목적으로 사용 시 별도의 사용료가 없는 오픈 소스 기반으로 개발되었기 때문에 SDN 관련 연구를 하는 대학 및 연구기관에서 SDN 테스트베드를 구축하는 데에 널리 사용되고 있다. 대표적인 오픈 소스 기반 SDN 스위치들은 다음과 같다.

Open vSwitch [18]: 오픈 소스 기반의 SDN 스위치 플랫폼으로 가상화 서버 장비 위에서 동작한다. 트래픽 전달 프로토콜로 OpenFlow 표준 프로토콜을 지원하며 스위치 구성에 대한 별도의 프로토콜로 OVSDB 프로토콜을 지원한다. 또한 고속 패킷 처리를 위해 ASIC 기반의 하드웨어 스위치 장비와의 연동을 지원한다.

Pantou/OpenWRT [19]: 무선 라우터 (즉, Access Point (AP))에 탑재되는 오픈 소스 기반의 운영체제인 OpenWRT에 OpenFlow 프로토콜 스택을 탑재하여 SDN 스위치로써 동작한다. 즉, 기존의 상용 무선 라우터 장비에 Pantou/OpenWRT를 탑재함으로써 OpenFlow 프로토콜을 지원하는 SDN 스위치로 동작할 수 있게 하는 것이다. 현재 지원하는 무선 라우터 장비로는 Broadcom사의 LinkSys WRT54GL와 Generic Broadcom 그리고 Atheros사의 TP-LINK TL-WR1043ND 등이 있다.

### 3. SDN 인터페이스

앞서 언급하였듯이 SDN 기술에서는 데이터 평면과 제어 평면과의 통신에 사용되는 프로토콜 (즉, 인터페이스)인 Southbound API, 그리고 제어 평면과 네트워크 애플리케이션 계층과의 통신에 사용되는 프로토콜인 Northbound API가 있다.

대표적인 Southbound API 프로토콜로는 ForCES와 OpenFlow가 있다. 두 프로토콜 모두 SDN 기술의 핵심인 제어 평면과 데이터 평면의 분리를 기본적인 전제로 하고 있다.

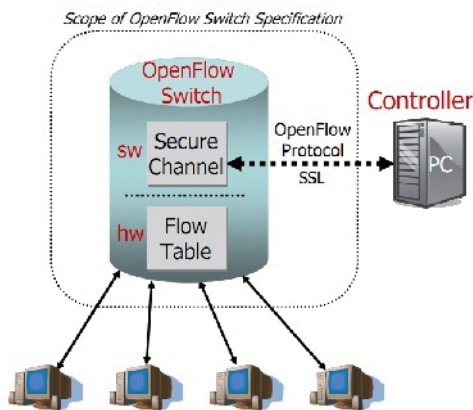


그림 2. OpenFlow 구조 [6]

OpenFlow 프로토콜은 서비스 사업자, 벤처 기업 및 대학으로 구성된 ONF (Open Networking Foundation)에서 개발 및 표준화 작업을 진행중에 있다. OpenFlow 구조는 그림 2와 같이 OpenFlow 스위치와 컨트롤러로 구성되며, 이들은 OpenFlow 프로토콜에 의하여 상호 연결된다. OpenFlow 프로토콜은 기존의 패킷 기반의 데이터 전달 방식과는 달리 플로우 기반의 데이터 전달 방식을 따르기 때문에 OpenFlow 스위치는 플로우 테이블을 유지하여 호스트 간의 트래픽 전달 기능을 수행한다. 한편, 컨트롤러는 보안 채널을 통해 OpenFlow 스위치의 플로우 테이블을 프로그래밍 하여 트래픽 전달에 대한 정책 등을 자유자재로 구현할 수 있다.

한편, 2003년에 IETF에 ForCES라는 워킹 그룹이 만들어졌다. ForCES는 IP 네트워크의 제어 평면과 데이터 평면의 분리에 대한 RFC 3654 [20]를 발표한 이후 현재까지 IP 포워딩, IntServ, DiffServ QoS를 제어대상으로 표준화를 추진하고 있다. ForCES는 플로우 기반의 프로토콜인 OpenFlow와는 달리 패킷 기반의 프로토콜이다. ForCES에는 데이터 평면에 위치하여 데이터 전달 기능을 수행하는 FE (Forwarding Element)와 제어 평면에 위치하여 제어 기능을 수행하는 CE (Control Element)가 있다. 한편, ForCES에서는 데이터 평면의 전달 기능 (즉, FE의 기능)을 LFB (Logical Functional Block)으로 모델링 하였다. CE는 LFB를 통해 FE의 구성, 관리, 그리고 패킷 처리 메커니즘 등을 제어할 수 있다.

반면, Northbound API는 SDN 컨트롤러와 네트워크 애플리케이션 간의 통신을 위한 인터페이스이다. Southbound API에는 대표적인 표준 프로토콜로 OpenFlow가 있지만, Northbound API의 경우 컨트롤러 공통의 표준 프로토콜이 아직 존재하지 않는 상황이고 따라서 각 컨트롤러 마다 자신만의 Northbound API를 정의하여 사용하고 있어 다양한 종류의 프로토콜들이 존재한다. 한편, 대다수의 컨트롤러에서 사용하는 Northbound API는 REST (Representational State Transfer) 형식을 기반으로 설계되었다. REST는 프로토콜 설계 형식으로, REST 형식을 따르는 모든 프로토콜들을 REST API라고 한다. REST API는 대부분 웹 페이지를 다운로드 받을 때 사용하는 HTTP 메시지 형식 (GET, POST, PUT, DELETE 등)을 기반으로 구현되며 stateless (즉, 상태 정보를 유지하지 않는 특성) 통신을 하기 때문에 간편하고 확장성이 좋다는 장점이 있다.

## IV. SDN 응용/활용 시나리오

현재 SDN은 기업/캠퍼스 네트워크, 데이터 센터 네트워크 등

과 같은 중소 규모의 네트워크뿐만 아니라 인터넷 서비스 제공자의 망과 같은 대규모 네트워크에도 적용하려는 많은 연구들이 진행되고 있다 [21]. 또한, 최근 가장 화두로 떠오르고 있는 사물인터넷 (Internet of Things: IoT) 인프라에도 SDN이 스며들고 있다. 본 장에서는 SDN을 적용한 응용 사례들을 보여준다.

## 1. 기업/캠퍼스 네트워크

일반 기업과 캠퍼스 네트워크는 엄격한 보안과 성능에 대한 요구사항을 가지고 있으며, 각 네트워크의 특징, 망 이용자 수 등이 각기 다르기 때문에 이를 적절하게 관리하기 위해서 SDN의 적용이 하나의 솔루션으로 등장하고 있다. 스탠포드 대학에서 개발한 Ethane [22]은 SDN 기반의 새로운 네트워크 구조의 시초이다. Ethane이라는 단일 중앙 집중형 컨트롤러가 플로우를 제어하여 복잡한 네트워크 관리를 단순화하고, 강력한 보안기능을 제공한다. 또한, 기업 내의 많은 AP가 설치되어 있는 무선 랜 환경을 SDN을 통해 변화시킨 Odin [23]이 있다. Odin은 기존의 AP를 그대로 쓰는 것이 아닌 컨트롤러가 AP들을 추상화시켜 가상 AP를 제공한다. 즉, 모바일 단말들의 상태 (예: 핸드오프 등)를 관찰하면서 가상 AP를 생성하고 이동시켜 기존 AP가 수행하지 못하는 기능을 수행할 수 있도록 하고, 효율적이고 유연하게 네트워크를 관리할 수 있게 한다.

## 2. 데이터 센터 네트워크

최근 서버들의 가상화 기술로 인해 기업들의 데이터 센터 네트워크가 변화하고 있다. 가상화 기술로 인한 운용 비용 절감, 유연성 등을 제공할 수 있지만, 여전히 복잡한 데이터 센터 네트워크를 효율적으로 관리하고, 제어하는 데에는 어려움이 있다. 따라서, 자원 활용률을 향상시키고, 전력 사용량을 줄이는 등의 최적의 데이터 센터 네트워크 관리를 위해 SDN을 적용하는 사례가 늘고 있다. 학계에서의 대표적인 구현으로 데이터 센터 내의 트래픽 상황에 따라 전원 관리 기능을 제공하는 ElasticTree [24]가 있다. Elastic Tree는 데이터 센터 내부의 트래픽 상황에 대한 모니터링을 기반으로 현재 상황을 처리하기에 적합한 최적의 토폴로지를 구하고, 이에 따라 네트워크 링크와 스위치와 같은 네트워크 구성 요소의 활성화 여부를 동적으로 제어한다. 트래픽의 사용량이 높은 경우에는 전력을 많이 사용하고, 낮은 경우에는 전력을 적게 사용하도록 패킷의 경로를 전력을 줄일 수 있는 방향으로 결정해주기 때문에 에너지 효율을 높일 수 있다. 또한, 기업에서도 자사 데이터 센터의 효율적인 관리를 위해 SDN을 적용하고 있다. 구글은 이미 2012년에 OpenFlow를 도입하여 'G-scale'이라는 프로젝트를 통해 그림 3과 같이 전 세계에 흩어져 있는 자사의 모든 데이터 센터를

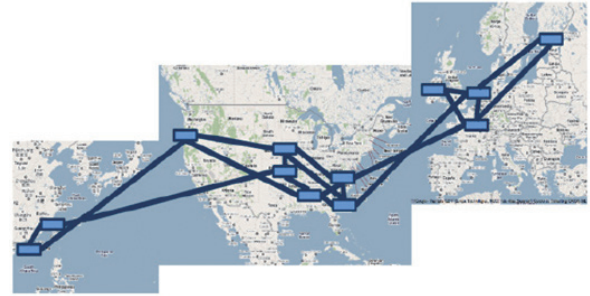


그림 3. 구글의 데이터 센터 네트워크 [25]

SDN 기반 라우터와 스위치로 재구축 [25]했다. 이를 통해, 비용 절감과 동시에 네트워크 트래픽 효율성 (즉, 네트워크 자원 활용률)을 90%이상 끌어올렸다.

## 3. 모바일 네트워크 / 광 전송 네트워크

SDN 개념을 대규모의 무선 및 모바일 네트워크와 광 전송 네트워크 환경에 적용하려는 움직임도 활발히 일어나고 있다. [26]에서 제안한 MobileFlow는 기존에 사용하고 있는 3GPP Evolved Packet System (EPS) 구조에 SDN을 적용하여 모바일 단말의 변화가 필요 없는 모바일 네트워크의 새로운 구조를 제시한다. SDN 개념을 도입하여 각각의 EPS 구성 요소들이 소프트웨어로 구성되어 제어 평면으로 볼 수 있는 Mobile Flow Controller (MFC)에 부여되고, 이를 통해 벤더가 제공하는 하드웨어적인 제한에서 벗어나게 된다. 또한 MFC는 부착된 소프트웨어들의 기능을 담당하여 인터페이스를 제공하고, 데이터 전달 평면으로 볼 수 있는 Mobile Flow Forwarding Engine (MFFE)은 IP/Ethernet 네트워크 간의 포워딩을 담당한다.

OpenRoads (혹은 OpenFlow Wireless) [27]는 연구자 또는 네트워크 관리자들이 무선 모바일 네트워크의 혁신을 위해서 이동성 관리 솔루션, 라우팅 프로토콜 등의 다양한 연구를 실험할 수 있는 환경을 제공하는 오픈 플랫폼이다. 이 플랫폼을 통해 이동성 서비스를 실제 환경에서 검증 및 실험할 수 있으며, 이기종 무선 망간 (WiFi, WiMAX 등) 간의 손실 없는 핸드오버를 가능하게 한다. OpenRoads는 SDN 컨트롤러와 외부 네트워크를 직접 제어 가능하게 하는 OpenFlow와, 그에 맞게 각 물리적인 네트워크 장비들의 환경설정을 제어할 수 있는 SNMP를 기반으로 하며, 사용자가 쉽게 원하는 기능을 구현하기 위한 API를 제공한다.

실제 망을 관리하는 통신사에서도 SDN을 적용하기 위해 다양한 프로젝트와 실험을 진행하고 있다. SK텔레콤은 vSOST (virtualized Service Oriented & Flexible T) 코어 (Core)라는 개발 프로젝트를 시작하였다. 이 프로젝트를 통해 SDN과 NFV

기술에 자체 보유한 실시간 네트워크 분석 시스템을 결합해 차세대 모바일 네트워크를 만들 예정이다. 예를 들어, 일시적으로 사람들이 붐비는 혼잡 지역에서도 간단한 솔루션 제어로 네트워크 품질을 최적화할 수 있다.

광 전송 망에도 차별화된 SDN 기술을 적용하려는 움직임이 보이고 있다. OpenFlow 표준화를 진행하고 있는 ONF에서는 Optical Transport Working Group (OTWG) [28]라는 워킹 그룹을 만들어 광 네트워크와 전송 네트워크를 지원하기 위해 OpenFlow를 확장하려고 한다. 이는 광 전송 네트워크의 제어와 관리의 유연성을 가능하게 해주고, 개발자들이 관리와 제어 시스템을 개발할 수 있도록 지원한다. 또한, 다수의 통신 사업자, 업체들이 참여하여 T-SDN (Transport-Software Defined Network) 기술의 표준화가 진행되고 있다. T-SDN 기술은 SDN 기술을 적용하여 전송 네트워크 관리의 자동화 및 개방화를 통해 BoD (Bandwidth on Demand) 서비스, 망 복구 및 IP 트래픽 오프로딩 등을 신속하게 제공하여 사용자들에게 유연한 전송선 서비스를 제공하고 고객 중심 맞춤형 서비스를 가능케 한다.

#### 4. IoT 네트워크

최근 IoT 시장이 앞으로 폭발적으로 성장하고, 기존 유선망과 융합해 새로운 네트워크 서비스들이 등장할 것으로 전망하고 있다. 이를 위해 먼저 다양한 모바일 기기를 지원하기 위해 네트워크 인프라에 확장성과 유연성이 요구되며, 기업들은 SDN을 활용해 IoT에 어울리는 인프라를 갖추려고 하고 있다. 화웨이 이는 SDN을 이용한 IoT 솔루션을 개발하였다. 사물을 인터넷에 연결해 기기간 통신을 구현하는 동시에 효율적인 네트워크 관리 및 유지 보수가 가능하도록 지원한다. 특히, IoT 게이트웨이를 제공하여 사물의 센서와 IP 네트워크를 연결하도록 해주며, 컨트롤러가 IoT 터미널, 게이트웨이, 컴퓨팅 자원, 애플리케이션 및 데이터의 통합 관리를 지원한다.

### V. SDN 연구/표준화 이슈

2007년 기본적인 개념이 제시된 이후 2010년 이후부터 SDN과 관련된 연구/개발 활동이 폭발적으로 증가하게 되었다. 초창기에는 중앙 집중 방식을 사용하는 SDN의 개념으로 인해 소규모 네트워크 위주로 활용 시나리오가 주로 언급되었으나 최근에는 사업자 망 수준에서도 SDN을 적용하고자 하는 연구가 많이 진행되고 있다. 이러한 캐리어급의 SDN 환경을 위해서는 무엇보다도 규모성 (Scalability) 보장이 무엇보다도 선행되어야

한다. 이를 위해 SDN 컨트롤러의 기능을 분산시키고 분산된 컨트롤러 간의 동기화를 제공하는 연구가 최근 활발히 진행되고 있다. 실례로 ODL에서는 Clustering 프로젝트를 통해 다수의 SDN 컨트롤러 간의 동기화 방식을 개발하고 있다. 또한 SDN 컨트롤러를 계층적으로 구성하여 규모성을 확보하기 위한 방안도 연구되고 있다. 즉, 하위 SDN 컨트롤러는 작은 규모의 네트워크를 담당하고 되고 상위 SDN 컨트롤러에서는 이러한 하위 SDN 컨트롤러에 대한 제어를 담당하는 방식이다. 그리고 이러한 규모성 보장과 함께 SDN 컨트롤러의 결합 시에도 강건하게 동작할 수 있는 고가용성 (High Availability) 보장 기술도 캐리어급 SDN 환경을 위해서 필수적인 기술이다.

규모성과 함께 여러 사업자에 의해 관리 되는 SDN 도메인을 연동하기 위한 연구도 진행되고 있다. 즉, SDN 컨트롤러 간의 East/Westbound API 정의를 통해 다중 SDN 도메인을 연동하기 위한 기법이 연구되고 있다. 이러한 환경에서도 컨트롤러 간의 정보 전달과 정도의 일관성 유지, 그리고 다중 도메인 데이터 평면에서의 이벤트 관리 등이 주된 이슈이다. 최근에 발표된 SDNi의 경우 이러한 멀티 도메인 환경에서의 주요 이슈들을 정리하고 있다.

SDN의 특징 중의 하나는 사용자/관리자의 요구에 따라 네트워크를 프로그래밍할 수 있다는 것이다. 즉, 우리가 C나 JAVA와 같은 프로그래밍 언어를 통해 프로그래밍한 것을 컴파일해서 컴퓨터에서 실행하는 것과 동일하게 네트워크를 대상으로 한 프로그램을 짜고 이를 컴파일해서 네트워크 상에서 실행할 수 있는 기능을 SDN이 제공할 수 있게 된다. 이러한 기능을 원활하게 지원하기 위해서는 사용자/관리자 측면에서 사용할 수 있는 다양한 API가 제공되어야 한다. 특히, 다양한 SDN 응용 개발을 위해서 SDN 컨트롤러가 제공해야 하는 Northbound API 정의가 필요한데 현 시점에서는 주도적인 또는 표준화된 Northbound API가 존재하지 않기 때문에 이에 대한 연구/개발이 필요한 상황이다. 데이터 평면 상의 다양한 장치들을 제어하기 위한 Southbound API는 가장 대표적인 OpenFlow를 포함하여 NETCONF, ForCES 등 다수 존재하는 상황이고 향후에도 하나로 통일되기 보다는 여러 규격이 경쟁하면서 발전해 나갈 것으로 예측된다. 이러한 API 뿐만 아니라 네트워크 상의 SDN 프로그램에 대한 시험/검증을 수행할 수 있는 시험/검증 소프트웨어 툴이나 디버거도 개발될 필요가 있다.

IT 산업 전반에 걸쳐 이슈가 되고 있는 보안은 SDN 환경에서도 중요한 이슈이다. 특히 SDN 컨트롤러에 외부 공격이나 위협이 가해졌을 때 네트워크 전체가 큰 영향을 받을 수 있으므로 SDN 컨트롤러와의 통신 과정에 보안 기법이 필수적으로 적용되어야 한다. 또한 대규모 SDN 환경에서는 SDN 컨트롤러 간



의 통신 과정에서도 일관성있게 정확한 정보 전달을 위해 보안 기법이 적용되어야 한다. 한편 초기 SDN 연구에서 발표된 바와 같이 SDN을 활용하여 네트워크 보안 시스템을 효과적으로 관리하는 것도 가능하다. 특히, 소규모 엔터프라이즈 네트워크나 홈 네트워크와 같이 보안이나 네트워크 관리를 위해 별도의 관리자를 배정하기 어려운 환경에서 SDN 기능을 이용하여 제 3자에게 보안/네트워크 관리를 위탁하는 기능을 쉽게 구현하는 것이 가능하다.

## VI. NFV 란?

NFV 기술이 등장한 배경을 살펴보면 점점 복잡해지는 네트워크 관리의 어려움과 이에 따른 비용 증가에서 비롯됨으로 볼 수 있다. ISP 입장에서는 복잡한 네트워크 관리가 용이하고, 여러 서비스를 지원하는 데 민첩하며 새로운 서비스 제공이 손쉽게 제공될 수 있는 네트워크 구조의 필요성이 부각되었다. 이러한 요구에 따라 기존의 하드웨어 위주의 네트워킹 기술을 하드웨어와 소프트웨어로 분리하고, 제어나 관리가 가능한 네트워킹 기술 개발이 시작되었다. 각 네트워킹 자원을 추상화하고, 자동화시키고 소프트웨어적으로 제어가 가능하도록 하는 NFV 기술이 본격적으로 개발되기 시작하였다. <그림 4>는 기존의 네트워크 장비들이 NFV 환경에서 변화되는 구조를 보여주고 있다. 여러 장비업체가 개발한 네트워크 기능 소프트웨어는 가상화를 지원하는 기능을 통해 물리적 자원을 사용하게 하는 구조이다.

2012년 10월 독일 다름슈타트에서 개최된 SDN World Congress에서 AT&T, BT, 도이치 텔레콤 등 주요 유럽 통신사업자들이 ETSI ISG 산하에 NFV 그룹을 구성함을 발표하였다.

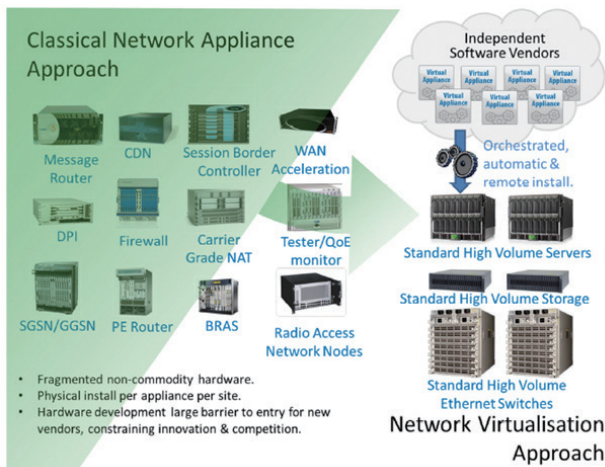


그림 4. NFV로의 네트워크 장비 변화 구조

이 그룹을 구성한 목적은 마켓이 요구하는 신규 서비스 창출이 용이하고, 기존 네트워크 장비와 자원들의 효율성을 높이고, 여러 장비업체가 개발한 네트워크 장비들간 호환성이 가능한 NFV 기술 분야 산업 규격 개발에 초점을 두었다. ETSI NFV 그룹은 통신사업자들의 요구사항을 반영하여 NFV 분야 산업 표준을 개발하는 것으로 현재까지도 38개 세계 주요 통신사업자들과 260여개 장비업체들이 모여 활발한 활동을 추진하고 있는 현황이다.

NFV 기술은 네트워킹에 필요한 자원 (즉, 컴퓨팅, 서버 및 네트워크 자원)을 추상화하고, 오픈 인터페이스를 통해 소프트웨어 적으로 네트워크 기능을 제어하는 기술로 정의할 수 있다 [29]. NFV 기술을 지원하는 전체 네트워크 구조를 마련하고, 주요 기능을 지원하는 기능 블록을 정의하고 이들간의 인터페이스를 정의함으로써, 표준화된 인터페이스에 따라 개발된 제품들간 상호 연결 및 동작되게 하는 기술이다.

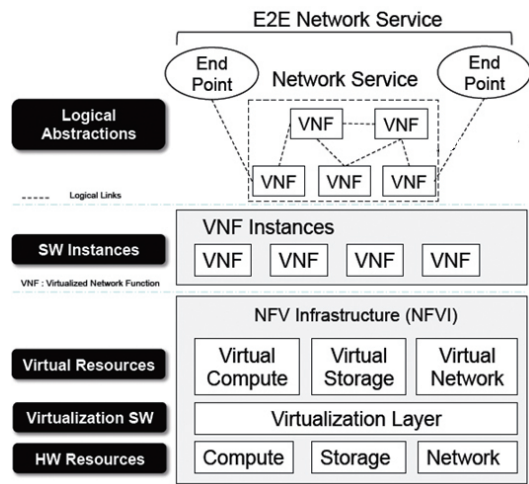


그림 5. NFV 구성 요소

<그림 5>에서 보여주는 바와 같이 NFV 기본 구조는 네트워크 서비스 계층, 가상 네트워크 기능 (VNF) 계층 그리고 NFV 자원 계층으로 구성된다.

- 네트워크 서비스 계층: 제공할 서비스 목적에 따라 사용하는 가상 네트워크 기능 (VNF)들을 이용하여 end-to-end 서비스를 추상화하는 계층
- 가상 네트워크 기능 (VNF) 계층: 업체들이 소프트웨어로 개발한 특성의 네트워크 기능 (예: 게이트웨이, firewall 등)들의 집합 계층
- NFV 자원 계층: 컴퓨팅 가상자원 (예: CPU), 저장 자원 (예: storage), 네트워크 자원 (예: 가상스위치)과 물리적 자원을 가상화해 주는 기능이 존재하는 계층

## VII. NFV 응용/활용 시나리오

본 장에서는 NFV 기술에 적용할 수 있는 여러 유형의 서비스 시나리오를 소개한다. NFV 유즈 케이스 문서에는 NFV 프레임워크와 주요 기능을 이용하여 네트워크 기능 가상화에 대한 다양한 유즈 케이스를 정의하고 있다[30]. 또한, NFV 그룹에 참여 기관들은 NFV 개념 및 기술 적용 가능성을 타진하기 위해 여러 유형의 PoC (Proof of Concepts) 를 개발하여 그 결과를 공유하고 있다.

- NFV Infrastructure as a Service (NFVIaaS): NFV 전체 인프라 자체를 하나의 서비스로 제공할 수 있는 시나리오에 해당한다. NFVIaaS는 클라우드 컴퓨팅 응용이나 여러 VNF를 대상으로 NFV가 가능한 인프라 기능을 제공할 수 있다는 경우에 해당하는 시나리오이다.

- Virtual Network Platform as a Service (VNPaas): 개발언어, 라이브러리, 개발 툴 등 과 같은 응용 프로그램에 대한 개발 환경과 네트워크상에서 실행에 필요한 기능을 제공하는 가상 네트워크 플랫폼으로 사용될 수 있는 시나리오에 해당한다. 여러 3rd party 가 동일한 ISP 인프라가 제공하는 NFV 기능들을 사용할 수 있는 예제에 해당하는 시나리오이다.

- Virtual Network Function as a Service (VNFaaS): 엔터프라이즈 액세스 라우터, Firewall, DPI 등 각 3rd party 가 VNF 기능을 이용하여 고유의 가상 네트워크를 구성할 수 있는 시나리오에 해당한다.

- Virtualisation of Mobile Core Network and IMS: 모바일 코어 네트워크의 구성요소인 MME, HSS, SGW, PGW 등의 네트워크 기능을 가상화 (예: vEPC) 하거나 IMS 시스템 구성요소인 P-CSCF와 S-CSCF 기능들을 가상화하여 여러 데이터 센터에 있는 하드웨어 장비에 유연하게 설치 및 동작하게 하는 시나리오에 해당한다.

- Virtualisation of Mobile Base station: 멀티 RAN 환경에서 RAN이 제공되는 여러 자원이나 네트워크 기능들을 이용하여 기지국을 가상화 시키는 시나리오에 해당한다.

- Virtualisation of the Home Environment: NFV 가상화 기술을 이용하여 기존의 RGW (Residential Gateway) 기능 (예: DHCP, NAT, uPnP 등)과 STB 기능(예: 리모트 UI 서버, VoD, 미디어 캐쉬 등) 들을 가상화하여 홈 네트워크 서비스를 제공하는 시나리오에 해당한다.

- 서비스 체인or VNF Forwarding Graph: end-to-end 서비스를 개발하거나 실행시키는 경우 서비스가 필요로 하는 여러 VNF를 연결하여 서비스 체인 또는 VNF forwarding graph를 만들 수 있는 시나리오이다.

- Virtualised CDNs (vCDNs): 미디어 캐싱 노드나 CDN 컨트롤러 기능을 이용하여 가상 CDN을 제공하는 시나리오에 해당한다.

## VIII. NFV 연구/표준화 이슈

<그림 6>은 NFV 기능 구조를 보여주고 있으며, 주요한 기능 블록의 수행 기능은 다음과 같다[31].

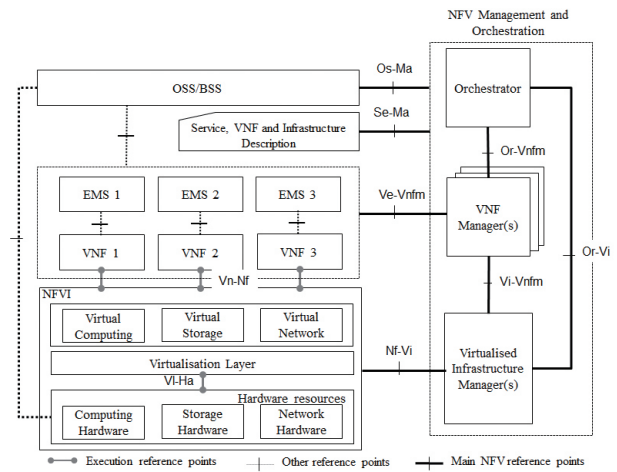


그림 6. NFV 기능 구조

- 오케스트레이터: OSS/BSS로부터 요청받는 네트워크 서비스의 라이프 사이클 관리에 필요한 오케스트레이션 기능 제공
- 가상 네트워크 기능 (VNF): 각 장비업체가 개발한 특정의 네트워크 기능을 제공하는 소프트웨어
- VNF 관리자 (VNFM): VNF 인스턴스의 생성, 종료, 스케일링 등 라이프 사이클 관리 기능 제공
- 가상 자원 관리자 (VIM): 오케스트레이터 또는 VNFM 요청에 따라 가상 자원 할당, 삭제, 예약 기능 제공

위에서 언급한 NFV 기능 블록들이 제공해야 하는 가장 중요한 기능은 네트워크 서비스가 동작하기 위해 필요한 기능과 환경을 제공하는 것이다. 이를 위해서는 네트워크 서비스가 사용하고자 하는 VNF 관리, 자원 할당 그리고 성능과 장애 관리가 제공되어야 한다. 우선 네트워크 서비스의 일반 정보뿐만 아니라 수행 시 요구사항 정보는 네트워크 서비스 기술자 (Network Service Descriptor)에 표기된다. 여기에는 사용하고자 하는 VNF 리스트 정보도 포함되며, 네트워크 서비스의 SLA, VNF 간 연결 관계성, 실행 순서, 정보 모델, 동작 의존성, VNF 업데



이트 발생 시 처리 방식 등을 기술할 수 있다. 이를 통해 네트워크 서비스의 등록, 인스턴트 생성, 종료, 스케일 업/다운, VNF Forwarding Graph 생성, 종료 단계를 관리하게 된다. 한편, VNF 관련 정보는 VNF Descriptor에 기술되어야 한다. VNF 관리자는 VNF 등록, 인스턴트 생성, 종료, 스케일 업/다운 단계를 제어하며, 각 VNF가 필요한 가상 자원은 VIM (가상자원 관리자)에게 요청하는 기능도 포함한다.

모든 가상 자원의 할당, 해지, 예약 등은 가상자원 관리자(VIM)가 수행하게 된다. 오케스트레이터 또는 VNF 관리자가 VIM에게 자원 요청을 하게 되며, 자원 요청 방식은 on-demand 또는 정해진 quota 에 따른 자원 할당 방식이 지원될 수 있다. VNF 레벨에서의 성능관리는 VNF 수행 시 성능 정보에 따라 스케일 업이 필요한 경우, VNF 관리자에게 요청하여 진행될 수 있다. 성능 정보 기반의 스케일 업이 이루어지는 규칙은 VNF Descriptor에 명기된다. 네트워크 서비스 레벨에서의 성능 관리는 관련된 모든 VNF들과 VNF Forwarding Graph 수행 시 성능 수치에 따라 오케스트레이터의 요청에 따라 이루어진다.

VNF 레벨에서의 장애는 VNF 장애를 인지한 VNF 관리자 또는 EM (Element Management) 의 요청에 따라 VNFMI가 정해진 힐링 기능을 수행하게 된다. 네트워크 서비스 레벨에서의 장애는 OSS/BSS 요청에 따라 오케스트레이터가 정해진 힐링 기능을 수행할 수 있다.

ETSI NFV 표준화는 전 세계 통신사업자들 중심으로 2012년 12월에 공식적 활동이 시작되어, Phase-1 기간 (2013년~2014년) 동안 NFV 요구사항, 구조, 유즈 케이스 등 산업 규격을 개발하였다. NFV 회의는 매년 4회 개최하며, NFV 그룹 전체에 해당하는 이슈를 점검하고, 각 워킹그룹별 작업 현황과 이슈를 점검하는 형태로 회의가 진행된다. 반면에 각 워킹그룹은 NFV 세부 규격을 개발하고, 정기회의뿐만 아니라 컨퍼런스 콜 또는 인터럽 회의를 개최하고 있다. Phase-2 (2015년 ~ 2016년) 표준화의 목표는 멀티 벤더 환경에서의 상호호환성이 보장되는 산업 규격 개발과 NFV 활성화를 위한 오픈 소스 커뮤니티들과의 협력에 초점이 맞추어 진행되고 있다. Phase-1 기간 동안 NFV 그룹에서는 여러 산업 규격과 NFV 세부 기능을 상호 시험할 수 있는 PoC 규격을 공지하였다[30]. 전 세계적으로 네트워크 시장은 시스코, 주니퍼 등 몇몇 주요 메이저 업체들에 의해 점유되어 있으나, 특정 네트워크 기술력이 있는 중소기업들은 NFV 규격에 맞게 재개발을 함으로써 NFV시장 진출 또는 확대라는 기회를 갖기 위해 NFV 표준 그룹에 매우 활발한 참여를 하고 있다.

## IX. 결론

지금까지 살펴본 바와 같이 SDN과 NFV는 네트워크의 관리/운용을 용이하게 하게 추상화와 가상화를 바탕으로 새롭고 혁신적인 서비스 및 아이디어가 네트워크에 보다 손쉽게 적용되는 것을 가능하게 해 주는 기술이다. 이를 통해 수익성 정체로 어려움을 겪고 있는 통신사업자들은 CAPEX 및 OPEX를 크게 줄일 수 있는 효과를 얻을 수 있을 것이고 이를 바탕으로 사용자는 더 나은 네트워크 환경을 제공받게 될 것이다. 기존의 폐쇄적인 환경이 아닌 오픈 인터페이스에 기반한 SDN/NFV구조를 통해 기존의 해외 장비 업체에 종속된 국내 네트워크 환경을 개선할 수 있는 방안이 될 수도 있지만 이에 대한 국내 네트워크 산업의 경쟁력을 키워 놓지 못한다면 보다 많은 수의 대규모/소규모 해외 장비 업체에 또 다시 종속받게 될 우려 또한 존재하는 것이 사실이다. 따라서, 국내 네트워크 산업의 새로운 부흥을 위해 산업체와 학계, 연구소가 긴밀한 커뮤니티를 결성하여 오픈 소스 기반의 SDN/NFV 시장에 공동으로 대응하는 노력이 경주되어야 할 것이다.

## Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R0127-15-1072, NFV Phase-2 기술 표준개발]

## 참고 문헌

- [1] Van der Merwe, J.E., Rooney, S., Leslie, I., and Crosby, S., "The Tempest—a practical framework for network programmability," IEEE Network, vol.12, no.3, pp.20-28, May/June 1998.
- [2] A. Doria, Ed., J. Hadi Salim, Ed., R. Haas, Ed., H. Khosravi, Ed., W. Wang, Ed., L. Dong, R. Gopal, J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification," RFC 5810, March 2010.
- [3] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)," RFC 6241, June 2011.

- [4] FIND project, <http://www.nets-find.net/>
- [5] SANE project, <http://yuba.stanford.edu/sane/>
- [6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication*, vol. 38, no. 2, April 2008.
- [7] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pp. 105–110, July 2008.
- [8] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, January 2015.
- [9] POX Controller. <http://www.noxrepo.org/pox/about-pox/>.
- [10] Floodlight, an open SDN controller. <http://floodlight.openflowhub.org/>.
- [11] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol," draft-ietf-netconf-restconf-04, January 2015.
- [12] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol (SNMP)," RFC 1098, May 1990.
- [13] OpenDaylight, <http://www.opendaylight.org>.
- [14] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN 2014)*, Chicago, IL, USA, August 2014.
- [15] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," RFC 6830, January 2013.
- [16] OpenStack, <https://www.openstack.org>.
- [17] IRIS: The Recursive SDN Openflow Controller by ETRI, <http://openiris.etri.re.kr/>.
- [18] Open vSwitch, <http://www.openvswitch.org/>.
- [19] Pantou : OpenFlow 1.0 for OpenWRT, [http://archive.openflow.org/wk/index.php/Pantou:\\_OpenFlow\\_1.0\\_for\\_OpenWRT](http://archive.openflow.org/wk/index.php/Pantou:_OpenFlow_1.0_for_OpenWRT).
- [20] H. Khosravi, Ed., T. Anderson, Ed., "Requirements for Separation of IP Control and Forwarding," RFC 3654, November 2003.
- [21] Bruno Astuto A, Nunes, M, Mendonca, X.-N, Nguyen, K, Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communication Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, Third Quarter 2014.
- [22] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 1–12, October 2007.
- [23] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with Odin," in *Proc. ACM SIGCOMM Workshops on HotSDN*, August 2012.
- [24] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving Energy in Data Center Networks," in *Proc. USENIX Conference on NSDI*, April 2010.
- [25] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Holzle, S. Stuart, and A. Vahdat, "B4: Experience with a Globally-Deployed Software Defined WAN," in *Proc. ACM SIGCOMM*, August 2013.
- [26] K. Pentikousis, H. Technologies, Y. Wang, and W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks," *IEEE Communications Magazine*, vol. 51, pp. 44–53, July 2013.
- [27] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "Openroads: Empowering Research in Mobile Networks," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 125–126, January 2010.
- [28] ONF Optical Transport Working Group, <https://>

www.opennetworking.org/images/stories/downloads/working-groups/charter-optical-transport.pdf

[29] ETSI NFV White paper #1, #2, #3, <https://portal.etsi.org/tb.aspx?tbid=789&SubTB=789>

[30] ETSI publication site, <http://docbox.etsi.org/ISG/NFV/Open/Published/>

[31] ETSI GS NFV-MAN 1.0 (2014.12) "NFV; Management and Orchestration"

## 약 력



백 상 헌

2000년 서울대학교 컴퓨터공학부 학사  
2005년 서울대학교 전기컴퓨터공학부 박사  
2007년~현재 고려대학교 전기전자전파공학과  
부교수  
관심분야: 미래 인터넷, 무선 이동 네트워크



장 인 선

2011년 고려대학교 전기전자전파공학부 학사  
2011년~현재 고려대학교 전기전자공학과  
석·박사통합과정  
관심분야: 미래 인터넷, 모바일 협업 네트워크, 차량  
클라우드 네트워크



서 동 은

2012년 고려대학교 전기전자전파공학부 학사  
2012년~현재 고려대학교 IT 융합학과  
석·박사통합과정  
관심분야: 모바일 멀티미디어 스트리밍, 미래 인터넷



이 종 화

1990년 한양대학교 대학원 전자공학 공학석사  
1996년 마드리드 국립대학 정보통신 공학박사  
1990년~현재 ETRI 표준연구센터  
네트워크표준연구실 근무  
관심분야: SDN, NFV, 서비스 가상화 등