

네트워크 기능 가상화를 위한 공개 플랫폼

OPNFV

(An Open Platform to accelerate Network Functions Virtualisation)

공개SW개발자Lab 오픈소스프론티어 2기 조만석

[목차]

들어가면서3	3
네트워크(Network)란?3	3
소프트웨어 정의 네트워크(SDN)란?3	3
오픈 플로우(OpenFlow)란?5	5
소프트웨어 정의 네트워크(SDN)를 제어하기 위한 사용자 어플리케이션7	7
기존과 소프트웨어 정의 네트워크의 비교)
네트워크 기능 가상화 (Network Function Virtualization, NFV) 9)
네트워크 기능 가상화를 위한 공개 플랫폼 (An Open Plantform for Network	
Function Virtualization, OPNFV)11	1
네트워크 기능 가상화를 위한 공개 플랫폼을 위한 계층별 아키텍쳐13	3
소프트웨어 정의 네트워크 컨트롤러 OpenDaylight15	5
클라우드 인프라 오픈스택(OpenStack)16	ĵ
OPNFV 세부 프로젝트17	7
NFV 와 OPNFV32	2
착조자료 3 2	1

들어가면서.

2007 년 아이폰이 등장하면서 우리들 생활은 언제 어디서든 인터넷에 연결하여 정보를 이용하는 시대로 바뀌었습니다. 스마트 폰을 시작으로 스마트 디바이스 사물 인터넷(Internet Of Things, IoT), 빅데이터 분석 기술, 클라우드 컴퓨팅 기술, 그리고 인공지능(Artficial Intelligence, A.I.) 기술의 발달이 가속화 됨에 따라서 사람과 기계가 서로 소통하는 방향으로 빠르게 변화하여 나아가고 있습니다. 이런 변화를 뒷받침하는 인프라 서비스 기술은 하드웨어 기반에서 소프트웨어 기반으로 빠르게 바뀌고 있습니다. 서로와 서로를 연결하는 네트워크에서도 이러한 변화에 따라 소프트웨어 정의 네트워크(SDN), 네트워크 기능 가상화(NFV)등의 기술로 빠르게 변화합니다. 네트워크 기능 가상화를 위한 기반 플랫폼 인프라를 공개 SW 기반으로 협업하여 개발하려는 것이 OPNFV(An Open Platform to accelerate Network Functions Virtualisation)입니다. 이 글 통하여 네트워크 기술을 시작하여 소프트웨어 기반 (SDN), 네트워크 기능가상화(NFV), 네트워크 기능 가상화를 위한 공개 플랫폼(OPNFV)에 대하여 설명하고, 그리고 마지막으로 제가 참여하는 OPNF의 중요 프로젝트를 설명하겠습니다.

네트워크(Network)란?

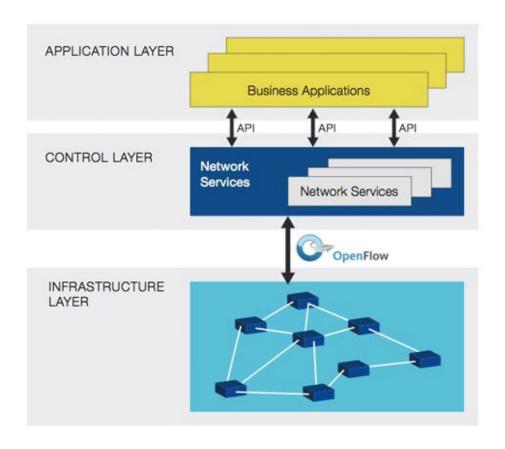
네트워크(network)란 여러가지 요소가 서로 연결된 그물과 같은 구조의 것을 말합니다. 사람과 사람간의 연결, 조직이나 집단의 구조등을 말할 수도 있지만, 정보화 기술(IT)에서는 여러 컴퓨터와 장치 등을 연결하여 제어 신호 또는 데이터와 같은 정보를 교환하는 것을 말합니다. 이것을 컴퓨터 네트워크(computer network) 또는 통신 네트워크(communication network)라고 합니다. 네트워크를 각 구성 요소를 "노드(Node)"라고 하고 노드 사이의 연결을 "링크(Link)" 또는 "엣지(Edge)"라고 합니다.

소프트웨어 정의 네트워크(SDN)란?

서로 다른 대상을 어떻게 연결하고 끊을지가 네트워크 성능을 향상시키는 기술입니다. 네트워크 업체들은 그들만의 특정한 하드웨어 장비들과 특정한 소프트웨어를 구성하여 네트워크 장비를 만듭니다. 예를 들면 100 개의 대상을 서로 연결하는 네트워크를 구성하려면 100 대의 대상에 따른 네트워크 구성을 세심히 검토하고 각각의 역할에 맞는 네트워크 제작사의 장비를 선정하여 구성합니다. 대상과 대상의 데어터 전송뿐만 아니라 네트워크가 다양하고 복잡하면서 방화벽, L4 스위치, DNS 서비스등 다양한 네트워크 서비스를 사용위해서나 또 다른 네트워크 장치를 추가하여 구축할 때마다 각각 다른 네트워크 장비제조사의 특정한 설정에 따라서 설정을 조정하고 관리했습니다.

네트워크 장비를 만드는 제조사는 네트워크 표준화를 표면적으로 내세웠지만 실제로는 각제조사마다 조금씩 다른 방식으로 기능을 제공하였고, 이로 인하여 공통된 표준에 따른네트워크라고 하더라도 각각의 상화에 따른 장비별 장단점을 반영하여 다양한 장비들을 구성할 수 밖에 없는 상황입니다. 그리하여 데이터 센터에는 다양한 컴퓨터와 저장장치뿐 만아니라 매우 복잡한 네트워크 장비와 그 연결이 공존하고 있습니다. 그리고 이러한네트워크를 구축, 운영하는 운영자들은 하나의 통일된 프로토콜이 아닌 각자 개별 목적에따른 다수의 프로토콜을 이해하고 운영하는 어려움이 있습니다. 그리고 많은 운영 인력과운영을 하면서 많은 문제점을 만나고 이를 해결하기 위한 많은 노력들이 함께 있어 왔습니다.

물리 네트워크(Physical Network)는 서버나 네트워크 장치를 추가하거나 사용사 어플리케이션이 네트워크 설정을 바꾸려고 할때, 많은 것을 해야만 합니다. 예를 들면 네트워크를 연결하는 케이블 구성부터, 네트워크 스위치, 라우터, 방화벽의 설정을 하나하나 바꾸어야만합니다. 이러한 물리 네트워크 구성을 하드웨어에 의존하지 않는 소프트웨어로 구성한 가상화기술을 통하여 구성한다면 이러한 설정의 불편함을 많이 줄어들 것입니다. 소프트웨어 정의네트워크(Software Defined Network)란? 소프트웨어를 통하여 기존의 물리 네트워크에서가상화된 네트워크 환경을 만드는 기술을 목표로 하여 만들었습니다.



[그림 1] Software Defined Network 의 정의 (https://www.opennetworking.org/sdn-resources/sdn-definition)

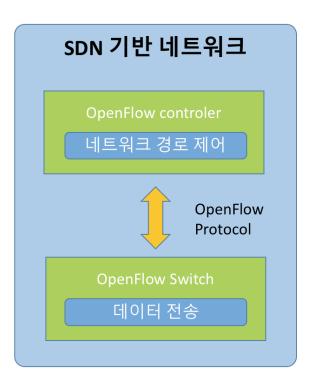
소프트웨어 정의 네트워크(SDN)은 네트워크 가상화와 제어의 두가지 기술이 핵심입니다. 네트워크 가상화는 서버에 네트워크 가상화 소프트웨어가 탑재되어서 구성합니다. 대표적인 네트워크 가상화 기술은 VMware 사의 "VXLAN"이나 마이크로 소프트의 "NVGRE"가 있습니다. 소프트웨어 정의 네트워크의 제어 기술은

"OpenFlow" 라는 표준 기술을 사용합니다.

오픈 플로우(OpenFlow)란?

컴퓨터와 또다른 컴퓨터 혹은 장치를 연결하는 네트워크의 데이터를 각각의 개별 장치를 설정하고 실행합니다. 이러한 기존의 네트워크 관리방식에서 OpenFlow 는 네트워크를 제어부과 전송부로 분리합니다. 무엇을 어디로 이동할지 알려주는 역할을 제어 플레인(control plain)이라는 부분으로 구성하며, 특정 목적지로 원하는 데이터를 보내는 부분은 데이터 플레인(data plain)이라고 합니다. 즉 OpenFlow 에서 네트워크는 제어 플레인과 데이터 플레인으로 나누어서 서로 다른 대상을 연결하고 관리합니다.

기존 네트워크 장치 네트워크 경로 제어 데이터 전송



[그림 2] SDN 기반 네트워크 장치는 경로 제어와 데이터 전송이 분리되어 전송

기존 네트워크 장비는 입력된 데이터(packet)의 도착할 곳을 네트워크 경로와 데이터 전송하나의 기기에서 독자적인 기준으로 다음 기기로 전송하는 방식입니다. 네트워크 구성요소가 각각의 방식에 따라서 네트워크를 제어하기 대문에 부분적인 장애가 발생하여도 다른 곳으로 이동하는 것을 독자적으로 결정하기 때문에 강한 장점을 가지고 있지만, 네트워크 전체로 보았을 때 결로 선택이나 통함된 관리가 어렵다는 단점이 있습니다. 가상화에 따른 클라우드가 데이터 센터의 주류가 되면서 이러한 네트워크를 큰 그림에서 관리해야만 하는 요구가 필요하게 된 것입니다.

OpenFlow 는 경로 제어를 하는 오플 플로우 제어기(OpenFlow controller), 데이터 전송을 하는 오픈플오우 스위치(OpenFlow switch)로 구성되며 제어기와 스위치를 오픈 플로우 프로토콜(OpenFlow protocol)로 관리합니다.

오픈 플로우 스위치는 하드웨어뿐 만 아니라 가상화 스위치나 소프트웨어를 통해서도 구현하여 사용합니다. 기존의 가상화 네트워크 분리를 VLAN(Virtual LAN) 기술을 사용했습니만, 연결 장치의 제약을 비롯한 부족한 부분이 있어서 가상 스위치를 포함한 오픈 플로우 기술로 VLAN 의 제약을 해결하고 더 많은 장치를 지원하는 대규모 환경에서 전가상화(Full virtualization)을 실현할 수 있게 되었습니다.

오픈플로우(OpenFlow)는 구글(Goolge), 페이스북(Facebook), 도이치 텔레콤(Deutsche Tenecom), 마이크로 소프트(Microsoft), 버라이즌(Verizon), 야후(Yahoo) 의 6 개 회사가 처음으로 주축이 되어 산업단체인 공개 네트워크 재단(Open Networking Foundation, ONF)가 표준화를 추친합니다. 처음에는 미국의 스탠포드 대학의 Martin Casado 교수의 연구로 부터 시작하여 스탠포드 대학 중심의 OpenFlow Swiching Consosium 이 2009 년에 버전 1.0, 2011 년 버전 1.1 을 만들었으며, 2011 년 3 월 ONF 가 설립되어 이곳에서 OpenFlow 버전 1.3, ... 현재 1.5.1 이 지난 2015 년 4 월에 공개되었습니다.

(OpenFlow 1.5.1 사양서 링크:

https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf)

소프트웨어 정의 네트워크(SDN)를 제어하기 위한 사용자 어플리케이션

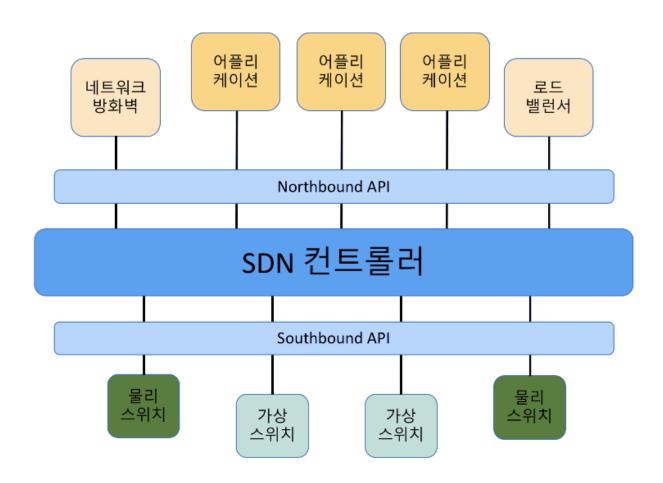
소프트웨어 정의 네트워크에서 물리/가상 스위치 제어를 위한 API를 Southbound API 라고합니다. Sountbound API 는 사용자 어플리케이션이 오플 플로우(OpenFlow)를 통한물리/가상 네트워크 스위칭 제어를 위한 것입니다. 이와 더불어 로드 밸런서(Load Balancer), 방화벽(Network Firewall)과 같은 사용자 어플리케이션이 소프트웨어 정의 네트워크를 사용할수 있어야만 합니다. 이러한 네트워크 어플리케이션이 사용할수 있는 네트워크 제어 정보를제공하는 API를 Northbound API 라고 합니다. 소프트웨어 정의 네트워크는 SDN 컨트롤러를제어 하기 위한 크게 두가지 API 층이 있습니다. 이렇게 크게 두가지 API 층으로 구분하면다음과 같은 장점이 있습니다.

첫 번째, 상위 계층의 어플리케이션도 SDN 컨트롤러도 모두 소프트웨어로 만들어 졌기때문에 하드웨어에 의존한 네트워크 시장에 비하여 새로운 서비스와 이를 지원하기 위한업체들이 네트워크 시장에 보다 쉽게 들어 올 수 있습니다.

두 번째, 애플리케이션 레이어가 지금보다 더 많은 서비스를 통하여 다양한 가치를 창출할 수 있을 것으로 예상됩니다. 어플리케이션이 API 를 통해 네트워크의 제어 정보에 접근할 수 있기

때문에 소프트웨어 정의 네트워크가 가지는 네트워크 패킷을 최적의 경로를 찾아서 전송하거나 환경에 따라 동적으로 QoS를 적용할 수도 있고, 보안에 위협을 주는 패킷의 출처를 찾는다처럼 다양한 상황에 맞추어 유동적으로 서비스를 제어하고 시험하며 적용할 수 있습니다.

세 번째, 네트워크 어플리케이션을 위한 Northbound API 가 있어서 비교적 작은 개발사들도 소프트웨어 정의 네트워크 시장에 접근을 할 수 있습니다. Southbound API 는 시스코와 주니퍼, 델, HP 등 몇몇 거대한 업체가 주로 개발과 시장을 차지하고 있지만 Northbound API 를 사용하는 네트워크 어플리케이션은 비교적 업체가 작지만 다양한 업체와 함께 어플리케이션에 집중하는 업체들이 있습니다. 이들이 Northbound API 를 이용하여 네트워크 시장에 보다 쉽게 참여할 수 있습니다.



[그림 3] SDN 어플리케이션을 위한 Northboud API 와 물리/가상 스위치를 위한 Southbound API

기존과 소프트웨어 정의 네트워크의 비교

	기존 네트워크	SDN
네트워크 관점	하드웨어 중심	소프트웨어 중심
구성 주도권	하드웨어 제공 벤더	사용자
기술 개방성	폐쇄적 구조	개방형 구조
연동 호환성	독자 프로토콜	표준 프로토콜
관리 효율성	비효율/고비용 운용	효율적/합리적 운용
신기술 수용	벤더의 필요에 따름	사용자 요구에 따라 수용
시장의 공정성	독과점 형태	공정 경쟁

[그림 4] 기존 네트워크 방식과 소프트웨어 정의 네트워크의 비교 (출처: 도서-SDN 입문, 서영석,이미주 공저)>

네트워크 기능 가상화 (Network Functions Virtualisation, NFV)

네트워크 기능 가상화(Network Functions Virtualisation: NFV)은 네트워크에 사용하는 여러가지 어플리케이션 구현한 네트워크 장비를 사용했습니다. 특정 제조사가 네트워크 기능을 하드웨어와 소프트웨어를 모두 제공하였습니다. 네트워크 장비 공급사마다 이를 각자의 방식을 사용하여 장비를 만들고 배포하였기 때문에 네트워크 기능은 그것을 만든 제조사에게 의존할 수 밖에 없습니다. 그리고 특정 장비에 종속적이다 보니 새로운 서비스 구성이나 확장이 필요로 할 때 오랜 기간에 검토와 평가가 진행되어 더딘 진행을 할 수 밖에 없는 제한된 환경입니다. 이것을 범용 서버 기술을 활용하여 서버 가상화 기술에 기반한 네트워크 소프트웨어(VNF, Virtual Network Function, 가상 네트워크 기능)로 대치하려는 것이 네트워크 기능 가상화(NFV)입니다.

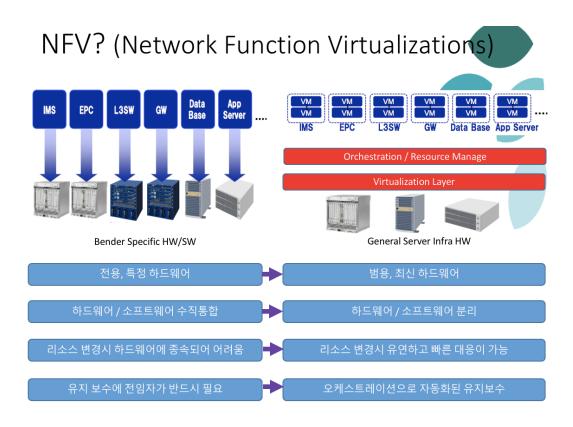
	SDN	NFV
출범 동기	Control, Data Plan 분리 중앙화된 관리 네트워크 프로그래밍	특정장비에 귀속된 네트워크 기능을 일반서버에 재배치
적용 위치	Campus, Data center/Cloud	Service Provider Network
사용 장비	상용서버와 스위치	상용서버와 스위치
Applications	Cloud Orchestration과 Networking	Router, Firewall, Gateway, CDN, WAN가속기 등
대표적 Protocol	OpenFlow	아직 제정 안됨
규격 제정	Open Networking Foundation	ETSI NFV Working Group

[그림 5] 소프트웨어 정의 네트워크과 네트워크 기능 가상화의 비교 (출처: 도서-SDN 입문, 서영석,이미주 공저)>

전 세계 주요 통신사업자가 주축이 되어 ETSI (European Telecommunications Standards Institute) NFV ISG(Industry Specification Group)을 출범하였고, 네트워크 기능 가상화에 필요한 요구사항, 프레임워크 구조, 사용자 환경, 검증 등에 대한 논의를 진행하고 있습니다. 올해(2016 년) 말이면 이를 정리한 사양서(1.0)가 배포될 계획입니다.

네트워크 기능 가상화를 통하여 네트워크의 주된 사용자인 통신사는 네트워크 장비의 구체적인 내부 구조까지 알 수 없었던 "블랙박스"의 구조에서 "화이트 박스"의 구조로 바뀔 수 있습니다. 이것은 네트워크 비지니스가 네트워크 장비를 만드는 제조사에 의해서 결정되는 구조에서 네트워크 서비스 공급자로 이전할 수 있습니다. 하드웨어와 소프트웨어를 분리(decoupling)하여 소프트웨어로 구현된 네트워크 기능을 범용 하드웨어 장비를 사용하는 클라우드 인프라 환경에서 동작합니다. NFV는 네트워크 서비스 사용자의 필요에 따라 네트워크 기능을 유동적으로 구성하고 운용할 수 있습니다. 소프트웨어 중심의 네트워크 인프라를 사용하기 때문에 네트워크 인프라에 대한 막대한 투자와 유지 부담을 안고 있는

통신사업자들은 NFV를 통해 빠르고하고 유연하게 서비스를 제공할 수 있으며, 특정 네트워크 장비 제조사에 대한 종속에서 벗어나 고가의 네트워크 장비에 대한 투자비용(Capital Expenditure: CAPEX)과 운용비용(Operating Expenditure: OPEX)을 절감할것으로 기대합니다.



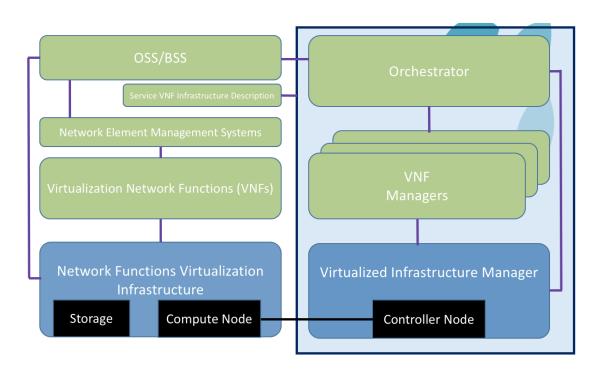
[그림 6] 기존 네트워크 어플리케이션 장비와 네트워크 기능 가상화(Network Functions Virtualisations) 비교

네트워크 기능 가상화를 위한 공개 플랫폼 (An Open Plantform to accelarate Network Functions Virtualisation, OPNFV)

네트워크 기능 가상화를 위한 인프라를 위한 공동 플랫폼이 필요하다는 인식이 여러 네트워크 통신 사업자, 네트워크 장비사업자간 협의가 이루어졌고, 그 결과 만들어진 프로젝트가 OPNFV 공개 SW 로 개발이 각각의 통신 사업자, 장비 사업자간의 협의가 이루어졌고, 이러한 논의로 탄생한 것이 네트워크 기능 가상화를 위한 공개 플랫폼 (an Open Platform to accelarate Networks Functions Virtualisation, OPNFV)입니다.

OPNFV의 주요 창립멤버는 AT&T, 브로케이드, 차이나모바일, 시스코, 델, 에릭슨, HP, 화웨이, IBM, 인텔, 주니퍼네트웍스, NEC, 노키아네트웍스, NTT, 도코모, 레드햇, 텔레콤 이탈리아, 보다폰 등 세계 각지 통신사와 통신장비 제조사들입니다.

OPNFV는 리눅스 재단에서 집행하며 프로젝트 활동 목표를 다음과 같이 정의 하였습니다. "OPNFV는 제조사, 클라우드 및 인프라 업체, 개발자와 사용자가 함께 원하는 새 레퍼런스 유형을 정의하고 기존 오픈소스 빌딩 블록을 새로운 구성요소와 통합하고 NFV 개발과 배포를 가속할 테스트를 진행하는 것"이라고 하였습니다. OPNVF의 프로젝트 범위는 사용 가능한 기존의 오픈 소스 컴포넌트를 이용하여 NFVI (NFV Infrastructure) 과 VIM (Virtualized Infrastructure Management: 가상화 인프라 관리)를 구축하는 것입니다. 네트워크 장비 제조사, 클라우드 및 인프라 업체, 개발자와 사용자가 함께 업계에서 원하는 새 레퍼런스 유형을 정의하고 기존 오픈소스를 활용하여 새로운 구성요소를 개발하거나 통합하여 개발과 배포, 테스트를 위해서 여러가지로 활동합니다.

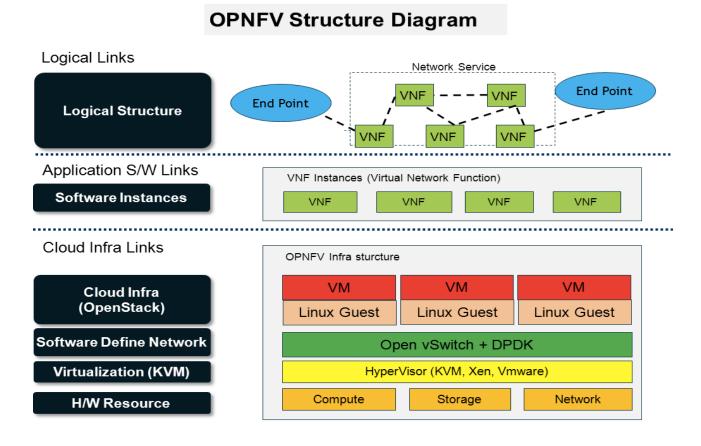


[그림 7] 네트워크 기능 가상화(Network Functions Virtualisations)

네트워크 기능 가상화를 사용하는 통신 사업자는 신뢰성이 높으며 서비스를 동적으로 활용할수 있는 플랫폼을 요구합니다. 네트워크 기능은 다양한 기능이 복잡하게 서로 연결되고 구성되므로 이것을 모두 만족한 플랫폼으로 통합하는 것은 쉽지 않은 일입니다. 따라서 OPNFV 프로젝트는 무엇보다도 통신사용자, 네트워크 장비 제조사가 함께 참여하는 커뮤니티 주도로 공개된 NFV 플랫폼을 위한 통합에 초점을 두고 있습니다.

네트워크 기능 가상화를 위한 공개 플랫폼을 위한 계층별 아키텍쳐

OPNFV 프로젝트는 개방화된 공개 SW 에 기반하며 운용체제부터 가상화 환경, 소프트웨어 정의 네트워크, 자원 관리를 모니터링 하고 이를 조정하는 소프트웨어로 구성됩니다.
OPNFV 에 사용하는 소프트웨어를 계층별 아키텍쳐를 그리면, 클라우드 인프라 와 네트워크 기능을 위한 어플리케이션으로 나누어 그릴 수 있습니다.



[그림 8] NFV 와 이를 뒷받침하는 공개 SW 기술

NF: Network Function, VNF: Virtual Network Function, NC: Network Controller

VN: Virtual Network

네트워크 기능을 소프트웨어로 구현하는 서비스이기 때문에 OPNFV는 많은 공개 SW 를기반을 필요로 합니다. CPU Core, 메모리, 네트워크와 같은 하드웨어 자원이 있고 이를 관리하기 위하여 공개 SW 운영체계인 Linux 를 사용합니다. Linux 운영 체제의 가상화 도구인 KVM 이나 Xen, VMware 등을 사용하여 가상화를 구현합니다.

가상화 도구를 REST API 로 관리하기 위한 IaaS (Infra As A Service) 공개 클라우드 소프트웨어로 OpenStack 을 사용합니다. OpenStack API 를 이용하여 각 VM 에 가상 프로세스, 가상 메모리, 가상 네트워크 자원을 할당/제거합니다. OpenStack 가상화된 네트워크 API 를 이용합니다. 이때 소프트웨어 정의 네트워크 (Software Defined Network)를 사용하여 네트워크 자원을 가상화로 구성하며, 이를 위해서 Open vSwitch 로 네트워크하드웨어를 사용하고, 이를 제어하기 위한 도구로 OpenDaylight 가 사용됩니다. 네트워크성능을 향상하기 위하여 OPNFV는 DPDK 기술을 사용한 네트워크 패킷 제어를 사용합니다. 가상화된 자원을 할당 받으면 Linux Guest OS 나 LXC 를 이용한 Docker 위에 가상화된네트워크 기능 소프트웨어(VNF Virtual Network Function)가 동작합니다. 하나는 사용자네트워크는 여러가지 VNF와 상호 논리적으로 연결를 필요로 하기 때문에 이를 관리하기위해서 OPNFV는 MANO를 사용하며, OPNFV의 자원을 운용하기 위한 OSS/BSS 가 이 모든자원의 조정을 담당합니다. OPNFV를 구성하는 주요 공개 SW를 나열해 보면 다음과같습니다.

* OPNFV 에 사용하는 주요 공개 SW

OPNFV : 공개 네트워크 기능 가상화 SW

Linux Kernel: 공개 운영 체제 SW

OpenStack : 공개 클라우드 인프라 SW

Open vSwitch: 공개 SW 정의 네트워크 스위치 소프트웨어/하드웨어

OpenDaylight: 공개 SW 정의 네트워크 제어 소프트웨어

DPDK: 공개 SW 정의 네트워크 데이터 영역 개발 도구

소프트웨어 정의 네트워크 컨트롤러 OpenDaylight

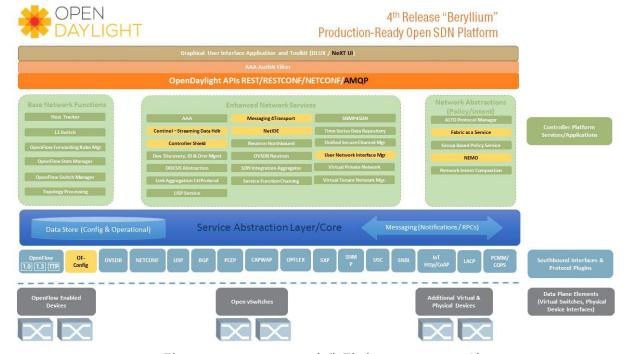
오픈데이라이트(OpenDaylight)는 소프트웨어 정의 네트워크의 구성 요소인 오픈 플로우(OpenFlow)를 제어하기 위한 공개 SW 입니다. 리눅스 재단에서 프로젝트를 관리합니다. 각 릴리즈는 화학 원소 기호를 사용해서 각 릴리즈시 이름을 붙여서 배포합니다. 최초 릴리즈는 2014 년 2 월 Hydrogen(수소)를 시작으로 2014 년 9 월 Helium(헬륨), 2015 년 6 월 Lithium(리튬), 그리고 현재 4 번째 릴리즈인 Beyllium 이 2016 년 2 월 릴리즈했습니다.

OpenDaylingt 는 SDN 을 관리하기 위한 어플리케이션과 연결되는 North Bound API 와 요청에 따른 경로와 대역등을 제어하는 Controller Platform, 각종 네트워크 기기를 제어하는 South Bound API 로 구성됩니다.

OpenDaylight 은 Cisco System, Hewlett-Packard, Juniper Networks, NEC 같은 네트워크 주요 제작사와 IBM, Red Hat, VMware 같은 서버사와 소프트웨어사가 참여하고 있습니다.

OpenDaylight 의 강점은 클라우드 인프라 소프트웨어인 'OpenStack "과의 연계입니다.
OpenStack 은 데이터 센터에 있는 서버, 스토리지, 네트워크라는 세가지 IT 자원을 통합 관리하고 사용자에게 가상화 된 서비스로 제공하는 기능을 가진 소프트웨어입니다. 각 IT 자원의 가상화는 각각의 가상화 소프트웨어가 담당합니다. 예를 들어 서버 자원의 가상화이면 "KVM", "VMware vSphere"나 "Xen Server " 가 담당합니다. OpenDaylight 는 네트워크 자원의 가상화를 담당합니다. OpenStack 은 OpenDaylight API 를 호출하여 OpenDaylight 가 관리하는 가상화 된 소프트웨어 정의 네트워크 자원을 이용합니다.

네번째 릴리즈로 지난 2 월에 출시 된 OpenDaylight 의 "Beyllium"은 OpenStack 가사용자에게 제공하는 네트워크 기능 서비스 로 "보안 그룹 ", "분산 가상 라우터", "Load Balancer as a Service (LBaaS)"등을 지원합니다.

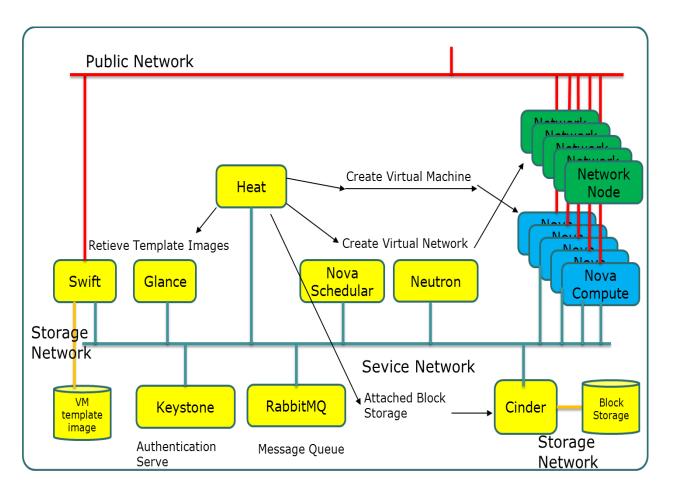


[그림 9] OpenDaylight 4 번째 릴리즈 Beyllium 구성

클라우드 인프라 오픈스택(OpenStack)

서버 가상화(Server Virtualization)와 이를 기반으로 한 클라우드 컴퓨팅(Cloud Computing)이 있습니다. 공개 SW 기반 클라우드 소프트웨어가 OpenStack 입니다. 공개 운영체제인 Linux 를 기반으로 하여 제공되는 소프트웨어로 제공되며, 주요 개발 언어로 Python 을 사용해서 개발됩니다. 라이센스는 Apache License 2.0 입니다. OpenStack 은 2010 년 Rackspace 사와 NASA 가 함께 시작한 프로젝트입니다.

초기의 OpenStack 은 Rackspace 의 객체 저장 서비스(Obejct Storage Service) Cloud Files 와 NASA 의 프로비저닝 시스템인 NASA Nebula 를 모아서 2010 년 10 월에 최초 버전을 릴리즈하였습니다. 이후 매년 2 번의 정규 릴리즈가 되어, 2012 년 9 월에는 오픈스택 재단 (OpenStack Foundation)이 설립되었습니다. Rackspace, NASA 와 같은 초기 구성원 뿐말 아니라, AT&T, Cisco, Dell, EMC, Ericsson, Hewlett-Packard, Huawei, IBM, Intel, NEC, NetApp, Oracle, Red Hat, VMware 과 같은 많은 IT 기업이 이 프로젝트에 동참하여 활발하게 활동하고 있습니다.



[그림 10] OpenStack 의 주요 구성 요소

오픈스택은 클라우드 구성을 여러가지 프로그램(구성요소, 스택 Stack)으로 구분합니다. 여러가지 구성요소가 있으며, 2010 년 10 월 첫 릴리즈 이후 반년에 한번씩 정규 릴리즈가 되며, 각 릴리즈는 알파벳의 첫 글자로 시작하는 도시 이름을 사용합니다.

OPNFV 세부 프로젝트

OPNFV는 여러가지 세부 프로젝트가 있습니다. 이를 크게 4 종류로 구분할 수 있습니다. 1. 요구 사항 프로젝트, 2. 통합 및 검증 프로젝트, 3 협력 개발 프로젝트, 4. 문서화 프로젝트입니다.

요구사항 프로젝트(Requirements and Deployment)는 NFV 를 위한 공동 플랫폼 개발을 위해 필요한 요구사항을 수집하여 함께 논의하여 문서화를 위한 프로젝트입니다. NFV 를 위해서 새롭게 개발이 필요하거나 기존의 것을 바꾸어 적용해야 한 것들이 있습니다. 이런 기능을 찾고, 정의하여 기능을 구현합니다. 통합 프로젝트(Build, Integration and Deployment)는 NFV 를 위한 플랫폼 개발을 위해서 필요한 공개 SW 를 구성하고 기능을 구현합니다.

어플리케이션과 검증 프로젝트(Application Deployment and Testing)는 OPNFV를 구성하는 어플리케이션을 위해서 요구사항을 정의하고, 협의하여 개발하며 이를 검증하는 프로젝트입니다.

문서화 프로젝트(Documentation Projects)는 OPNFV를 위한 기술 문서, 설치등 다양한 문서를 만들고 이를 뒷받침해주는 도구를 개발하는 프로젝트입니다.

OPNFV Project pipeline



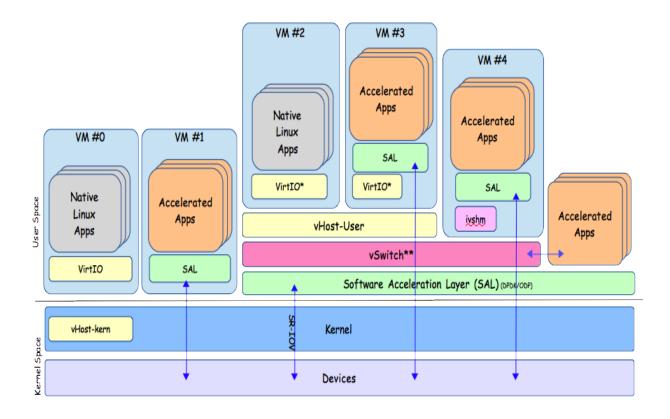
[그림 11] OPNFV 세부 프로젝트

I. 요구사항 프로젝트 (Requiements and Deployment)

I-1) DPACC: Data Plane Acceleration

DPACC (Data Plane Acceleration)는 네트워크 인프라 구조에서 네트워크 데이터 플레인 (data plain)을 빠르게 처리할 수 있는 가속화 기능이 필요합니다. 데이터 플레인을 가속하기 위한 하드웨어를 사용하거나, 범용 CPU 를 사용을 선택할 수 있어야 하며 이를 지원하기 위한 리눅스

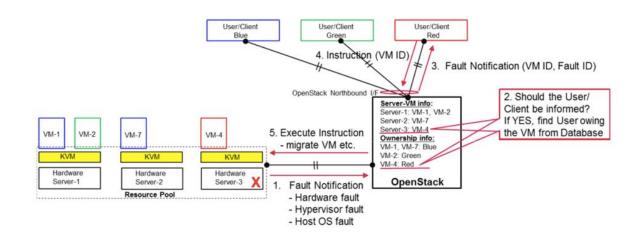
커널 지원부터 사용자 어플리케이션을 위한 API를 정의하여 이를 개발합니다. DPACC는 VNF의 이식성(portability)을 보장하기 위한 프레임 워크와 이를 위한 API를 정의하고 개발합니다.



[그림 12] DPACC architecture

I-2) Doctor: Fault Management

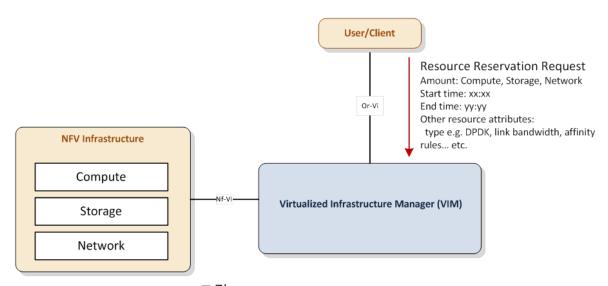
Doctor 는 플랫폼에 대한 장애 관리와 유지 보수를 위한 프로젝트입니다. 가상화된 인프라의 네트워크 서비스의 고 가용성을 보장하기 위해서 이상 상황에 대한 장애 관리와 유지 보수를 위한 프레임 워크를 만드는 것입니다. 주요 기능으로 VNFs 의 장애를 복구하기 위한 처리와 VIM 의 가상화된 자원의 가용성을 유지하기 위한 통지 기능을 포함합니다. OpenStack Nova 와 OpenStack 의 Ceilometer 프로젝트와 함께 협업하면서 개발되고 있습니다.



[그림 13]Fault Managemnet

I-3) Promise: Resource Management

Promise 는 컴퓨팅, 네트워크 및 스토리지 자원을 예약하고 리소스 풀의 용량을 관리하여 서비스 자원을 효과적으로 관리하이 위한 프로젝트입니다. OpenStack Blazar 프로젝트가 관련 프로젝트입니다. ETSI 의 MANO, INF, IFA 와 관련됩니다.



[그림 14] NFVi Resource Managemnet

I-4) Copper: Virtualized Infrastructure Deployment Policies

Copper 는 가상화된 인프라에서 가상 네트워크 기능(Virtual Network Function,VNF)의 속성에 따라 디자인된 정책 기반으로해서 가상 인프라를 구성하는 프로젝트입니다. VNF 의 요구사항에 맞게 플랫폼에서 정책 기능을 공개 SW 프로젝트를 활용하여 개발합니다.

OpenStack 의 Congress 프로젝트와 OpenDaylight 의 Group-based Policy(GBP) 프로젝트에 대한 연구와 OPNFV 의 NFVi 플랫폼 범위해서 관련있는 오픈 소스 프로젝트를 통합하고 테스트를 통해서 평가합니다.

Copper 프로젝트와 관련된 ETSI 표준은 다음과 같은 것을 들 수 있습니다.

ETSI GS NFV001 "Network Functions Virtualisation - Use Cases"

ETSI GS NFV002 "Network Functions Virtualisation – Architectural Framework"

ETSI GS NFV004 "Network Functions Virtualisation – Virtualization Requirements"

ETSI GS NFV INF001 "Network Functions Virtualisation – Infrastructure Architecture Overview"

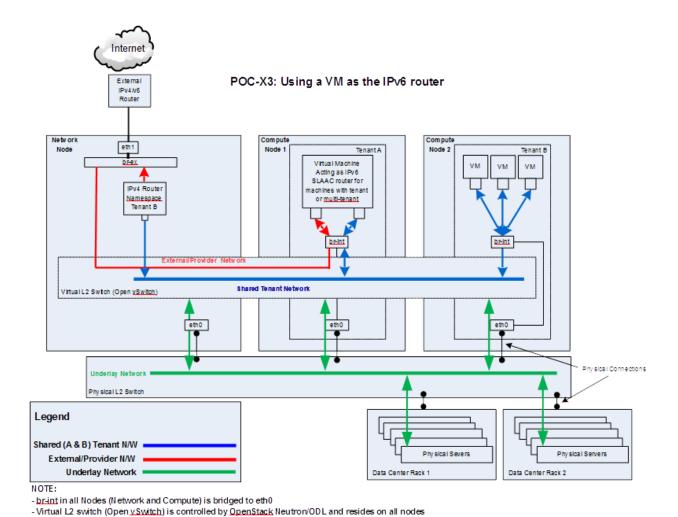
ETSI GS NFV INF003 "Network Functions Virtualisation – Infrastructure Architecture; Architecture of Compute Domain"

ETSI GS NFV INF004 "Network Functions Virtualisation – Infrastructure Architecture; Architecture of Hypervisor Domain"

ETSI GS NFV INF005 "Network Functions Virtualisation – Infrastructure Architecture; Architecture of Infrastructure Network Domain"

I-5) IPv6-enabled OPNFV

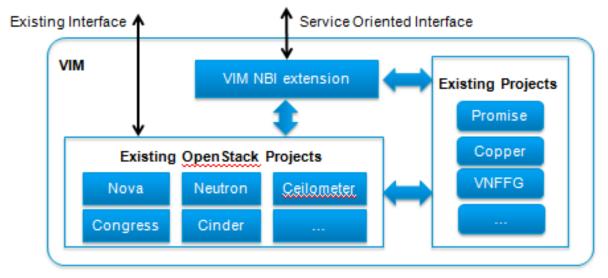
IPv6 프로젝트는 OPNFV 플랫폼에서 IPv6 를 사용하기 위한 추가 구성 요소, 기능 블록 및 도구를 만들고 통합할 수 있는 프로비저닝과함께 IPv6 듀얼 스택 지원을 쉽게 하도록 할 수 있는 패키지 제공을 목표로 합니다. IPv6 를 위한 다양한 통신 업계 요구사항과 함께 지원에 필요한 기능 상의 격차를 분석하고 이를 채우기 위한 해결 방법을 제안합니다.



[그림 15] VM 을 활용한 IPv6 router

I-6) Movie: Model Oriented Virtualization Interface

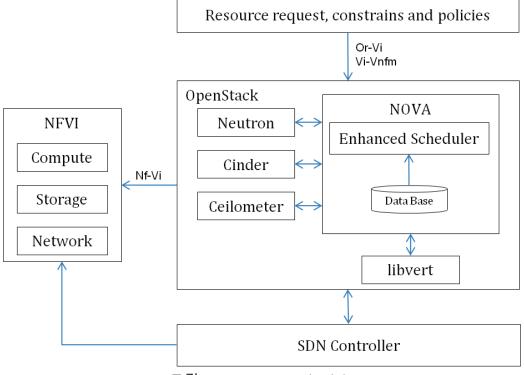
Movie 는 OPNFV를 관리하는 상위 레이어(예. NFV Orchestrator)나 VNFM (VNF Manager)를 위한 인터페이스(northbound interface)를 제공하기 위한 프로젝트입니다. 컴퓨팅, 네트워크, 스토리지 서비스 자원의 접근, VFN 의 접속, 흐름 식별, 정책 조작을 위해서 추상화를 통하여 상위 레이어에서 접글을 할 수 있는 인터페이스를 NBI(northobud interface)와 함께 OpenStack 에서 확장된 인터페이스를 적절하게 조절하여 캡슐화합니다.



[그림 16] VM 을 활용한 IPv6 router

I-7) RS: Resource Scheduler

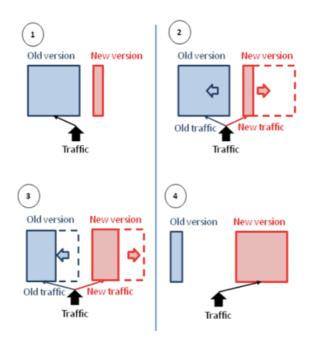
Resource Scheduler 프로젝트는 가상화 인프라에서 CPU 나 메모리, 그리고 네트워크에 대한 자원(resource)를 적절하게 할당하고 관리하기 위하여 필요한 정보를 수집하고 특히 네트워크리소스를 분리하여 할당을 효율적으로 하기 위한 자원 스케쥴러를 확장하는 프로젝트입니다.



[그림 17] Resource Scheduler

I-8) Smooth Upgrade: Escalator

Smooth Upgrade(Escaltor) 는 NFVI 서비스 중단없이 VIM 의 어플리케이션을 원활하게 업그레이드하기 위한 프로젝트입니다.



[그림 18] VNF 소프트웨어 Upgrade 과정(ESTI ISG NFV REL003)

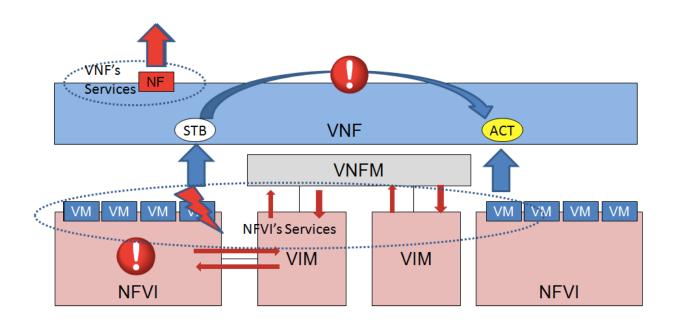
Smooth Upgrade 프로젝트와 관련된 ETSI 표준

ETSI ISG NFV REL003

ETSI ISG NFV REL006

I-9) Availiability: High Availability

고가용성(High Availability) 프로젝트는 통신장비가 반드시 요구되는 수준의 고가용성을 제공하기 위해서 하드웨어 HA, 가상 인프라 HA 및 서비스 HA 를 구체화합니다. OPNFV 의고가용성 요구사항과 함께 HA 를 위한 API 를 정의하고 개발합니다.



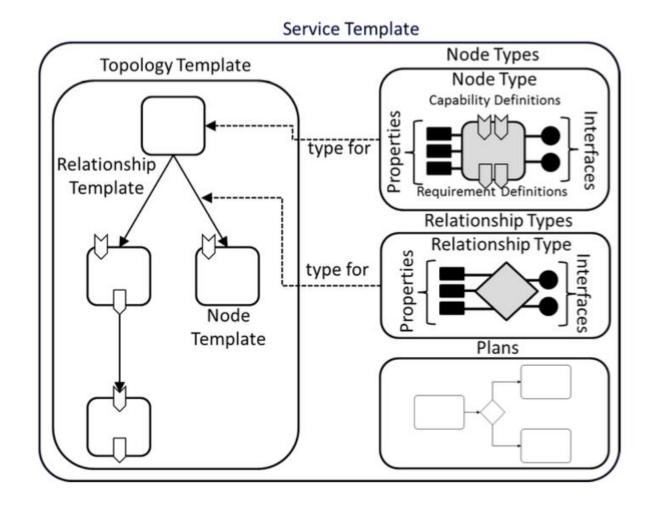
[그림 19] NFVI 고가용성

High Availability 프로젝트와 관련된 ETSI 표준

The ETSI NFV REL GS (e.g. Draft ETSI GS NFV-REL 001 V1.0.0 (2014-11))

I-10) Parser: Deployment Template Translation

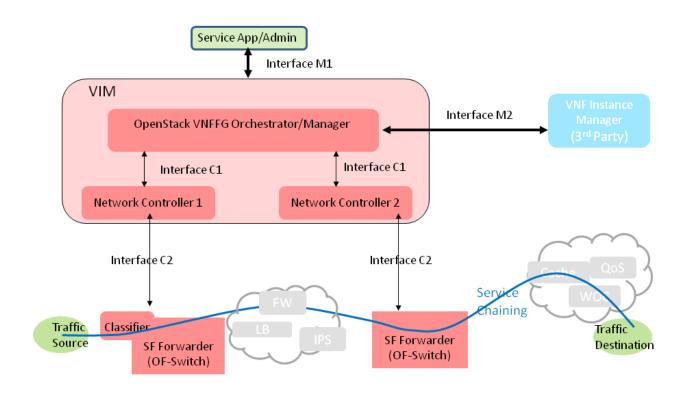
Parser 프로젝트는 통신사업자가 자신들의 환경에 맞게 네트워크 기능을 배포할 때 사용하는 템플릿 양식을 TOSCA(Topology and Orchestration Specification for Cloud) / CAMP(Cloud Application Management for Platforms)로 변환합니다. 템플릿 변환에 필요한 배포 오케스트레이션, 자동화 특정 도구를 OpenStack Heat 의 VIM 템플릿으로 변환합니다.



[그림 20] TOSCA 서비스 템플릿의 구조

I-11) VNFFG: OpenStack Based VNF Forwarding Graph

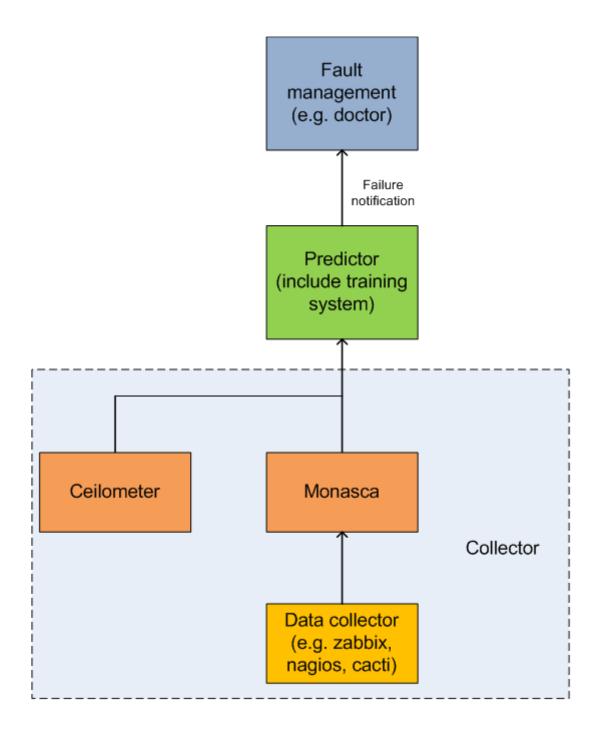
VNFFG 프로젝트는 ETSI NFV 관리 및 오케스트레이션 (MANO) 사양 [NFVMAN001]에 대한 NFV 의 구성 요소중 하나입니다. VNFFG 를 통하여 종단과 종단 간 의 VNF 서비스를 자동화하고 신속한 배포를위한 프레임 워크를 제공합니다. ONF 의 OpenFlow 와 OpenStack 사이의 링크를 생성합니다. 다수의 VNFs 에 대하여 VNFFG 를 통하여 네트워크 서비스를 연결합니다.



[그림 21] VNFFG 인터페이스와 동작

I-12) Prediction: Data Collection for Failure Prediction

Prediction 프로젝트는 NFV 시스템이 예기치 못한 고장으로 부터 방지하기 위한 오류 예측 시스템입니다. 장애에 대비하기 위한 데이터를 수집하는 데이터 콜렉터, 고장 예측 및 장애관리 모듈로 구성됩니다. 예측 프레임 워크, 데이터 수집기 모델 및 대시 보드를 통하여 서비스를 제공하는 것을 목표로 합니다.



[그림 22] Prediction, 장애를 막기 위한 데이터 수집, 예측

I-13) ONOSFW: ONOS Framework

ONOSFW 프로젝트는 VIM 의 SDN(Software-Defined Networking) 컨트롤러로서 ONOS(Open Network Operating System)을 사용할 수 있도록 관련 프레임워크나 라이브러리 등을 개발하는 프로젝트입니다.

Project Scope OpenStack Virtualized Resource Pool INITIAL FOCUS OF OPNFV scope of this proposal Proposed works Swift Neutron NB DevOps APIs - Service framework and library extensions, Topology plug-in to horizon, Intent Open Open framework, Resource allocation, ONOS+ Daylight Contrail Reloadable Services Support Modules SB DevOps APIs - Adaptor library OVSDB OF BYO NetConf and extensions(BYO), Virtualized Resource Pool and Physical Infrastruc To be extended Ready to be Integrated

[그림 23] ONOS 프레임 웍

II. 통합 프로젝트 (Build, Integration)

II-1) Genesis: Deployment

Genesis 프로젝트는 OPNFV를 설치를 위한 인프라 환경을 구성하는 프로젝트입니다.

II-2) Octopus: Continuous Integration

Octopus 프로젝트는 Jenkins 를 기반으로 OPNFV 를 구성하기 위한 다양한 공개 SW 를 통합하고 검증하여 패키치를 빌드하는 프로젝트입니다.

II-3) OSCAR: OPNFV System Configuration and Reporting

OSCAR 프로젝트는 Cobbler/ Puppet 을 이용하여 OPNFV 플랫폼를 설치하고 설정하기 위한 시스템을 개발하는 프로젝트 입니다.

III. 협업, 어플리케이션과 검증 프로젝트 (Collaborative Development & testing)

III-1) FuncTest: Base System Functionality Testing

FuncTest 프로젝트는 OPNFV 플랫폼을 시험하고 검증하기 위한 시험 항목, 시험 도구, 시험 과정을 개발하고 플랫폼에 대한 검증 시험을 실행하는 프로젝트입니다.

III-2) Qtip: Platform Performance Benchmarking

Qtip 프로젝트는 OPNFV 플랫폼에 대한 성능 평가를 하여 NFVi 에 대한 시험검증 및 벤치마킹을 위한 도구를 위한 프로젝트입니다.

III-3) Yardstick: Infrastructure Verification

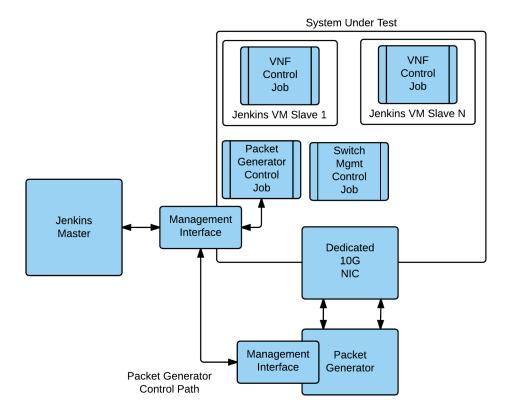
NFV 인프라가 동작할때 인프라 환경을 검증하기 위한 프로젝트입니다.

ETSI 관련 문서로는 다음과 같은 것이 있습니다.

- * ETSI GS NFV 001
- * ETSI TST001

III-4) VSPERF: Characterize vSwitch Performance for Telco NFV Use Cases

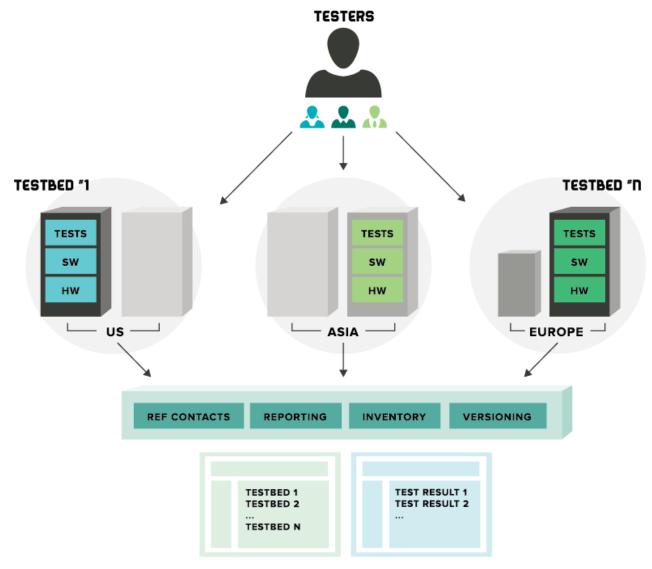
VSPERF 는 서로 다른 다양한 가상 스위치를 시험하고 검증할 수 있는 프레임워크 개발하는 프로젝트입니다.



[그림 24] VSPERF 를 이용한 자동화된 검증 과정

III-5) Pharos: Testbed Infrastructure

Pharos 프로젝트는 OPNFV 커뮤니티 후원사에서 제공하는 테스트 시험 환경을 서로 교차하여 검증 할 수 있도록 서비스를 제공합니다. 각 커뮤니티 후원사가 제공하는 OPNFV 시험환경은 OPNFV 하드웨어 참조 규격에 따라 구축되며 지리적으로 분산된 시험 환경을 통하여 OPNFV 플랫폼을 검증합니다.



[그림 25] Pharos 프로젝트 (OPNFV 테스트 환경 교차 검증)

IV. 문서화 프로젝트

OPNFV Documentation 프로젝트는 OPNFV 배포판에 대한 릴리즈 노트, 설치 가이드, 사용자가이드 등의 각종 문서를 개발하는 프로젝트입니다.

NFV 와 OPNFV

주니퍼 리서치 사업자를 위한 보고서에 의하면 2014 년에 WhatsApp, Facebook 과 Skype 등음성과 메시징 트래픽의 140 억 달러의 시장 규모입니다. 영국, 이탈리아, 스페인 등 주요통신사들의 보고서 따르면 2020 년이 되면 모바일 음성 수익은 60 %의 감소하지만 데이터수요는 폭팔적으로 증가할 것으로 예상됩니다. 따라서 향후 이동 통신 사업자들의 데이터중심의 네트워크 서비스 비즈니스 모델을 개선하기 위해서 다양한 도전과 전략이 전개 될것으로 예상합니다.

이동 통신 사업자들의 보고서에 따르면 2020 년 메시징 트래픽이 IM, VoIP 및 소셜 미디어를 통한 데이터 트래픽으로 대체할 것으로 예상됩니다. 따라서 이러한 데이터 트래픽 증가로 인하여 향후 2020 년까지 660 억 달러로 예상되는 네트워크 장비를 추가 할 것으로 예상합니다. 이러한 데이터 트래픽 증가는 통신 사업자의 새로운 기회가 될 수 있습니다. 사물 인터넷 (loT Internet of Things)과 같은 M2M(Machine to Machine)등의 새로운 분야의 서비스가 함께 늘어나면서 통신 사업자는 전체 서비스 제공이 아닌 상호 연결성에 초점을 맞춘 인프라를 구성할 필요가 있습니다. 이러한 변화에 NFV는 유용한 해결안이 될 것입니다.

NFV 를 지원하기 위한 공개 SW 플랫폼인 OPNFV 는 여러 통신 장비 사업자(Brocade, CISCO, ERICSSON, HUAWEI, Juniper, NEC, NOKIA, ZTE, 6Wind, ADVA, Array network, BII, Brain4Net, Broadcom, CableLabs, Cavium CentruryLink CertuNet, Ciena, ClearPath Dailogin, Dorade, ENEA, IXIA, HBC, OPEN NET, QOSMOS SANDVINE, SPIRENT, Stratus, Wind River) 와 통신 서비스를 제공하는 통신사(AT&T, China Mobile, docomo, Italia Telecom, Vodafone, KDDI, KT, ORANGE, SKT, Sprint) IT 장비 업체인 (IBM, Intel, DELL, HP, EMC², Lenovo)와 함께 반도체 제조사(, ARM, ALTERA, FreeScale, XILINX), 그리고 소프트웨어 업체로 (RED HAT, SUSE, Ubuntu, Miodkura, Mirantis, VMWARE) 가 새로운 네트워크 시장을 위하여 협업을 통하여, 공동의 플랫폼을 개발하고 있습니다.

OPNFV 에는 아주 많은 공개 SW 를 기반으로 하여 구성됩니다. 운영체계로 Linux kernel 을 공개 SW 운영체제를 기반으로한 배포판인 Redhat/CentOS, Ubuntu Serve 을 사용합니다. 이들은 각각 Redhat 과 Canonical 재단에서 관리합니다. 공개 클라우드 운영 인프라웨어로 OpenStack 을 사용하며 OpenStack Foundation 에서 관리합니다. 클라우드 스토리지 기술인 Cept 는 레드햇에서 관리합니다. 소프트웨어 정의 네트워크(SDN)를 위한 Open Daylight,

Open vSwitch 등을 사용하며 이는 Linux Foundation 에서 OPNFV 와 함께 관리합니다. 이처럼 OPNFV는 공개 SW 커뮤티니를 통하여 소스 코드를 위시한 기술 공개와 함께 커뮤니티 구성원들의 참여를 통한 협업을 통하여 발전을 계승합니다. 대부분의 개발자 통신 장비업체와 통신사, 그리고 Redhat 과 같은 공개 소프트웨어 소속으로 저를 포함하여 약 150 여명이 활동 중입니다. 매년 2 회 정도 공개 Summit 을 통하여 기술을 공개하며 관심 있는 기업이나 단체, 그리고 개발자의 사용과 참여를 유도합니다. 본 글을 통하여 OPNFV 에 대한 이해와 관심, 그리고 함께 참여하기를 희망합니다.

참조자료

- 1. "ReleaseNotes/mitaka OpenStack". Wiki.openstack.org. Retrieved 2016-04-07.
- "OpenStack Open Source Cloud Computing Software". Retrieved 29 November 2013.
- "OpenStack Launches as Independent Foundation, Begins Work Protecting,
 Empowering and Promoting OpenStack". BusinessWire. 19 September 2012.
 Retrieved 7 January 2013.
- 4. "OpenStack Foundation Mission". Retrieved 7 January 2013.
- 5. [NetVM: High Performance and Flexible Networking Using Virtualization on Commodity Platforms] Using refstack templates
- "OpenStack Roadmap » OpenStack Open Source Cloud Computing Software".
 Openstack.org. Retrieved 07 April 2016.
- 7. <u>"OpenStack Compute: An Overview"</u> (PDF). openstack.org. 2010. Retrieved 2014-03-31.
- 8. "Heat OpenStack". Wiki.openstack.org. Retrieved 2014-05-06.
- 9. "Ironic". wiki.openstack.org. Retrieved 2014-09-24

- 10. "OpenStack BoD". OpenStack Foundation. Retrieved 3 February 2014.
- 11. "Mitaka Release Schedule Wiki". Wiki.openstack.org. Retrieved 2016-04-07.
- "Industry Leaders Collaborate on OpenDaylight Project, Donate Key Technologies
 <u>to Accelerate Software-Defined Networking"</u> (Press release). April 8, 2013. Retrieved
 November 18, 2013.
- "OpenDaylight SDN opens the curtains on its initial release". ZDNet. September 12, 2013. Retrieved November 18, 2013.
- 14. Ehrman, Doug (2013-04-15). "Cisco Joins the Open Daylight Project". Motley Fool. Retrieved 2013-04-23.
- 15. "OpenDaylight Project". IBM. Retrieved 2013-04-23.
- Shamus McGillicudy, <u>Intel DPDK</u>, <u>switch and server ref designs push SDN</u> ecosystem forward, SearchSDN, April 2013
- 17. Intel Corporation, Intel® Data Plane Development Kit: Programmers Guide
- 18. "Network Functions Virtualisation (NFV); Use Cases". Retrieved 6 June 2014.
- "Network Functions Virtualisation— Introductory White Paper". ETSI. 22 October
 2012. Retrieved 20 June 2013.
- 20. Tom Nolle (18 September 2013). "Is "Distributed NFV" Teaching Us Something?". CIMI Corporation's Public Blog. Retrieved 2 January 2014.
- 21. Carol Wilson (3 October 2013). <u>"RAD Rolls Out Distributed NFV Strategy"</u>. *Light Reading*. Retrieved 2 January 2014.
- 22. What's NFV Network Functions Virtualisation?, SDN Central

- 23. Network Functions Virtualization (NFV) <u>Use Cases</u>, ETSI GS NFV 001 v1.1.1 (2013-10)
- 24. Mulligan, Ultan. <u>"ETSI Publishes First Specifications for Network Functions Virtualisation"</u>. Retrieved 5 December 2013.
- 25. "Network Functions Virtualisation". ISG web portal. Retrieved 20 June 2013.
- 26. Mulligan, Ultan. <u>"ETSI Publishes First Specifications for Network Functions</u>
 Virtualisation". Retrieved 5 December 2013.
- 27. OPNFV mailing List
- 28. Carrier Network Virtualization, ETSI news