

K-ONE Technical Document #38

A Design of M-CORD Monitoring System

Document No. K-ONE #38

Version 0.1

Date 2018-12-19

Author(s) 김우중, 홍지범

■History

Version	Date	Author(s)	Contents
Draft - 0.1	2018. 12. 19	김우중, 홍지범	1차 초안 완성
Draft - 0.2	2018. 12. 26	김우중, 홍지범	1차 교정 작업

본 문서는 2017년도 정부(미래창조과학부)의 재원으로 정보통신
기술진흥센터의 지원을 받아 수행된 연구임 (No. B0190-15-2012, 글로벌
SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발)

This work was supported by Institute for Information & communications
Technology Promotion(IITP) grant funded by the Korea government(MSIP)
(No. B0190-15-2012, Global SDN/NFV OpenSource Software Core
Module/Function Development)

Summary

본 기술문서는 M-CORD 상에서의 모니터링 시스템 디자인을 서술한다. M-CORD란 ONF에서 개발 중인 모바일 네트워크를 엣지 클라우드에서 서비스하기 위한 플랫폼이다. 이 플랫폼 내에는 LTE EPC 네트워크를 구성하는 다양한 NFV들이 구동이 되고 있다. M-CORD 모니터링 시스템은 이들 NFV가 송/수신하는 데이터 트래픽 및 각 NFV들의 컴퓨팅 및 메모리 자원을 모니터링 하는 시스템이다. 이 모니터링 시스템을 통해, M-CORD 상의 각 NFV들이 얼마나 많은 컴퓨팅 자원을 소모하고, 얼마나 많은 데이터 트래픽이 송/수신되는지를 모니터링할 수 있다. 본 기술문서에서는 이러한 M-CORD의 모니터링 시스템을 구현하기 위한 디자인을 제안한다.

Contents

K-ONE #38. A Design of M-CORD Monitoring System

1. Introduction 7

2. CORD 9

3. M-CORD architecture 18

4. A Design of M-CORD monitoring system24

List of Figures

그림 1. Vision of CORD..... 10

K-ONE #38. A Design of M-CORD Monitoring System

1. Introduction

Central Office Re-architected as a Data Center(CORD)는 미국의 Open Networking Foundation (ONF)에서 활발히 연구/개발되고 있는 데이터 센터 플랫폼이다. CORD의 개념은 기존의 통신사(예: SK Telecom, AT&T, 등)들이 운용하는 전화국(CO; Central Office)을 데이터 센터로 변경하여 운용하는 개념이다. 기존에 전화국 내에 있는 네트워크 장비들은 가상화하여 CORD 플랫폼 내 서버에서 동작시키는 한편, 이 이외에도 다양한 어플리케이션 및 가상 네트워크 기능(VNF; Virtual Network Function)들을 서버에서 동작시킨다. 기존에 서비스하는 네트워크 장비들 이외에도, 다양한 VNF들을 함께 CORD 플랫폼 상에서 구동시켜, 다양한 유즈케이스 및 어플리케이션들이 제공될 수 있을 것이라 기대한다(예: 엣지클라우드 컴퓨팅, AR/VR 서비스 등).

CORD 플랫폼은 그 목적에 따라 크게 Residential CORD (R-CORD), Enterprise CORD (E-CORD), 그리고 Mobile CORD (M-CORD)로 구분된다. R-CORD는택내 사용자들의 네트워크를 CORD 플랫폼을 통해 연결하는 것이다. E-CORD는 회사 내의 네트워크를 CORD 플랫폼으로 구성하는 것이다. 특히 대규모 회사의 경우, 본사와 지사가 여러 지역 혹은 여러 국가에 위치할 수 있다. CORD 플랫폼은 이러한 본사 및 지사들의 네트워크를 하나의 사내 네트워크로 구성하도록 설정할 수 있다. 마지막으로 M-CORD는 기존에 통신 사업자가 운용하는 모바일 LTE 네트워크를 CORD 플랫폼을 통해 서비스를 제공하는 것을 의미한다.

LTE 네트워크를 서비스하기 위해, M-CORD는 기존의 LTE EPC의 네트워크 기능(예: Serving Gateway, Packet Gateway, Mobility Management Entity 등)을 가상화하고, CORD 내 서버에서 구동한다. 이 후, 외부에 설치된 기지국(eNB; evolved NodeB)들을 M-CORD에서 동작하는 VNF로 연결을 함으로써, M-CORD를 통해 모바일 네트워크를 구축할 수 있다. 또한, M-CORD 역시 하나의 데이터 센터 플랫폼이며, 이 플랫폼에 새로운 어플리케이션 및 다른 VNF들을 동작시킬 수 있다. 이로 인해, 최근 상당히 많은 연구 및 개발이 되고 있는 모바일 엣지 컴퓨팅(MEC: Mobile Edge Computing) 혹은 다양한 유즈케이스 및 어플리케이션을 제공할 수 있을 것으로 생각된다.

하지만 이러한 장점이 있는 M-CORD는 현재 ONF에서 연구/개발 진행 중인 프로젝트이며, 기능적으로 완벽하지 않은 프로젝트이다. 더욱이 M-CORD의 경우, 데이터 센터 플랫폼 중 하나임에도 불구하고, 아직 M-CORD에서 운용되는 다양한 노드 및 VNF들의 컴퓨팅 자원 및 네트워크 대역폭과 같은 기본적인 정보조차 모니터링하는 시스템 부재의 문제가 있다. 모니터링 시스템이 존재하지 않으면, M-CORD의 성능을 조정 및 최적화를 하거나 기타 다

른 의사결정을 하는데 있어 어려움을 겪을 수 있다.

본 기술문서에서는, 이러한 문제를 해결하고자, M-CORD 내에 모니터링 시스템을 어떻게 구축할 지에 대한 디자인을 제시한다. 이를 위해, 기본적인 CORD 및 M-CORD의 아키텍처 및 구성 요소들을 설명한다. 그리고 이를 기반으로 M-CORD 내에 어떠한 요소를 추가함으로써, 모니터링 시스템을 개발할 지에 대한 디자인을 제안한다.

본 기술문서의 구성은 다음과 같다. 먼저 2절 CORD에서는 CORD의 아키텍처 구조 및 M-/E-/R-CORD에 대해 더욱 상세히 알아본다. 다음으로 3절 M-CORD architecture에서는 M-CORD만을 위해 어떤 특징이 있는지를 살펴본다. 마지막으로 4절에서는 M-CORD의 모니터링을 위해 어떻게 디자인을 할 것인지를 제시하고 5절에서 결론을 맺는다.

2. CORD

2.1. Overview of CORD

CORD 플랫폼을 살펴보기 전, CORD 이전의 네트워크 환경에 대해 살펴본다. 통신 사업자는 전화 통신 및 패킷 통신을 위해 전화국을 주요한 위치에 설치를 하고, 전화국에서 통신 교환을 수행하였다. 이와 같은 네트워크 장비들은 개방적(Open)이지 않고 폐쇄적(Closed)이며, 범용 장비(General-purpose devices)가 아닌 전용 장비(Unique and dedicated devices)였다. 이로 인하여, 통신 사업자는 기지국 내에 네트워크 장비들의 유지 및 보수 비용이 많이 들고, 더욱이 구매를 위해서도 많은 비용을 지출했다. 미국의 AT&T의 경우, 미국 전역에 AT&T 전화국이 약 4,700개가 설치되어 있으며, 각각의 전화국에는 최소 300개 이상의 폐쇄적인 전용 장비들이 운용되고 있는 실정이다.

각 전화국에서 이와 같은 폐쇄적인 전용 장비를 운용함으로써 인해 다음과 같은 몇 가지의 문제점이 야기된다. 먼저 어떠한 새로운 기능을 요구되는 경우, 항상 제조사에서 이러한 기능들을 대신 개발해줘야 한다. 이 경우, 새로운 기능 개발에 필요한 금전적 시간적 비용이 많이 요구된다. 다음으로 범용적이지 않은 전용 장비이므로, 장비를 구매하는데 많은 비용이 소모됨은 물론, 장비 제조사에 종속적인 관계를 야기한다. 또한, 더 많은 네트워크 장비들이 필요한 경우, 소프트웨어적으로 기능을 확장하는 개념이 아닌 폐쇄적인 전용 하드웨어를 추가로 구입 및 설치를 해야하는 문제점도 있다. 이러한 경우, 마찬가지로 높은 구매비용 그리고 운영비용이 발생된다.

이러한 문제를 해결하는 방식으로 등장한 것이 바로 CORD(Central Office Re-architected as a Data Center) 프로젝트이다. 이 프로젝트의 주요 목적은 기존의 전화국을 데이터 센터로 변경하는 것을 목표로 한다. 기존의 기지국이 폐쇄적인 전용 장비를 사용했다면, CORD 프로젝트는 이와 반대로 개방적인 범용 장비를 사용한다. 그리고 이러한 범용 장비를 활용하여, 기존의 기지국이 하던 역할을 그대로 하도록 한다. 이를 위해, 기존에 기지국이 가진 다양한 네트워크 기능들을 가상화 하고, 가상화된 VNF들을 범용 장비에 구동시킨다.

위와 같이 개방적인 범용 장비를 사용한다면, 다음과 같은 장점이 있다. 먼저 제조사에 상관 없이, 다양한 제조사에서 생산한 범용 장비를 사용할 수 있다. 이를 통해, CORD 플랫폼은 제조사에 종속적인 관계를 끊을 수 있는 장점이 있다. 그리고 전용 장비의 경우, 새로운 기능을 추가하기 위해선, 장비 제조사에 요청을 하여 새로운 기능을 배포할 수 있었다. 하지만 범용 장비 상에서 VNF들을 구동하는 경우, 새로운 기능이 필요할 때, 운영자가 원하는

기능을 VNF 혹은 범용 장비 상에 직접 구현을 할 수 있는 장점이 생긴다. 이러한 특징들로 인해, 개방적인 범용 장비를 사용하게 되면 장비를 운용하는 비용과 장비들을 구매하는 비용을 낮출 수 있는 장점이 있다.

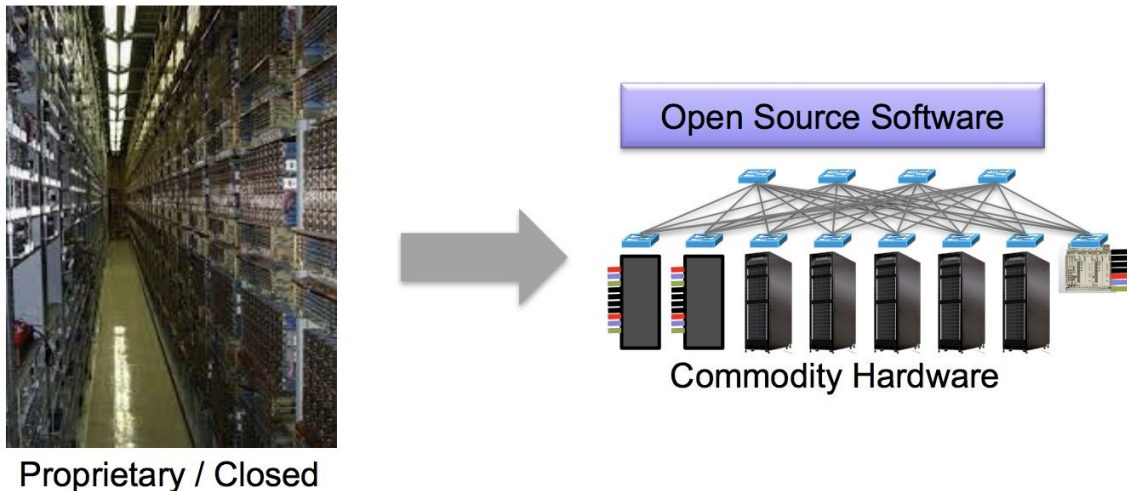


그림 1. Vision of CORD

그림 1은 CORD 프로젝트의 비전을 담은 그림이다. 좌측에 있는 그림은 기존 전화국의 모습인데, 내부에 폐쇄적인 전용 장비들이 무수히 많이 설치됨을 볼 수 있다. 한편 오른쪽 그림은 CORD의 구조를 나타내는 그림이다. CORD는 시장에서 구매할 수 있는 범용 장비를 활용하여 데이터 센터를 구축하였으며, 내부에서 구동되는 소프트웨어는 오픈 소스 소프트웨어(Open-Source Software)로 구현되도록 한다. 그 결과, 운영자는 CORD를 비교적 저렴하게 구축할 수 있음은 물론, 내부에서 구동 중인 VNF 및 소프트웨어 역시 수정 및 추가를 할 수 있는 장점이 있다.

CORD를 그림 1과 같이 구성하기 위해선 기본적으로 SDN(Software-Defined Networking), NFV(Network Function Virtualization), 그리고 클라우드 컴퓨팅 기술을 사용해야 한다. SDN은 소프트웨어를 통해 네트워크를 제어/관리하는 기술을 의미하며, NFV는 폐쇄적인 전용 장비로 구성된 네트워크 기능들을 가상화하여 범용 장비에서 구동시키는 기술을 의미한다. 마지막으로 클라우드 컴퓨팅 기술은 사용자 혹은 관리자의 수요에 따라 서비스하는 가상 머신 등의 자원들을 탄력적으로 조정할 수 있는 기술을 의미한다. 이들 기술을 통해, CORD는 상용 SDN 네트워크 장비를 사용하여 데이터 센터 네트워크를 제어/관리할 수 있으며, 다양한 전용 장비들을 데이터 센터에서 운용할 수 있다.

다음의 그림 2는 CORD 플랫폼을 사용할 때, 각 사업자 혹은 사용자들이 가질 수 있는 새

로운 서비스들을 도식화 한 그림이다. 먼저 통신 사업자(Telco)의 경우, 기존에 운용하는 전화국을 새로운 데이터 센터로써 사용을 할 수 있다는 장점이 있다. 또한 통신 사업자 및 네트워크 사업자의 경우, CORD를 활용함으로써 인해, 5G 네트워크나 기타 발전된 유선 네트워크 서비스들을 제공할 수 있는 장점이 있다. 서비스 제공 업체들의 경우, 포그 컴퓨팅(Fog computing)이라는 새로운 서비스를 제공할 수 있으며, 장비 제조사 혹은 서비스 제공 업체들은 CORD 플랫폼을 엣지 클라우드로써 사용하여, 모바일 엣지 컴퓨팅 기술을 활용할 수 있는 기저가 된다.

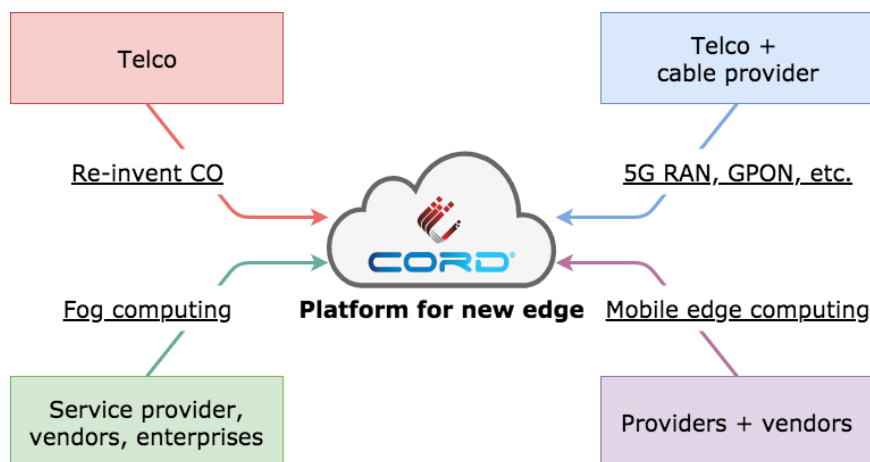


그림 2. New services with CORD

이러한 장점이 있는 CORD를 구축하기 위해선, 다음의 요구사항이 있다. 기본적으로 범용 서버와 화이트 박스 스위치들을 사용해야 한다. 그리고 데이터 센터 내에 구동 중인 서비스는 현재보다 훨씬 더 많을 수 있으므로, CORD 플랫폼의 확장 가능성 역시 고려를 해야 한다. CORD는 데이터 센터 플랫폼이므로, 당연히 다중 태넌시 환경을 고려해야 하며, 강건성이 보장되어야 하는 요구사항이 있다.

2.2. CORD Architecture

이와 같은 요구사항을 고려하여 CORD는 그림 3과 같은 구성 요소를 갖는다. 우선 전체적으로 CORD를 제어/관리하는 CORD 컨트롤러가 존재하는데 이는 XOS라고 부른다. 이 XOS를 통해, CORD가 포함하는 다양한 구성요소를 모두 제어/관리하도록 하며, 이를 서비스 제어 평면(Service control plane)이라고도 한다. 반면 XOS 아래에 있는 구성 요소들은 서비스 데이터 평면(Service data plane)이라고 하며, 크게 ONOS 컨트롤러, OpenStack/Kubernetes, 그리고 하드웨어로 구성되어 있다. ONOS 컨트롤러는 하드웨어 중 화이트 박스 스위치를 제어하는 오픈 소스 소프트웨어이다. 이 컨트롤러에는 화이트 박스 스위치의 제어를 위한 다양한 어플리케이션들이 정의되어 있다(Ctrl App). OpenStack/Kubernetes는 하드웨어 중 서버(Compute node) 내의 가상 머신/컨테이너를 제어/관리하는 소프트웨어이다. 이를 통해, CORD는 사용자가 원하는 VNF들을 서버에 배포할 수 있는 특징이 있다. 하드웨어는 오픈 컴퓨트 프로젝트(Open Compute Project)에 속한 하드웨어를 일반적으로 사용한다고 가정한다. 그리고 이들은 앞서 ONOS 컨트롤러로부터 네트워크를 제어 받거나, 혹은 서버 상에 배포된 다양한 VNF들을 구동하는 역할을 수행한다.

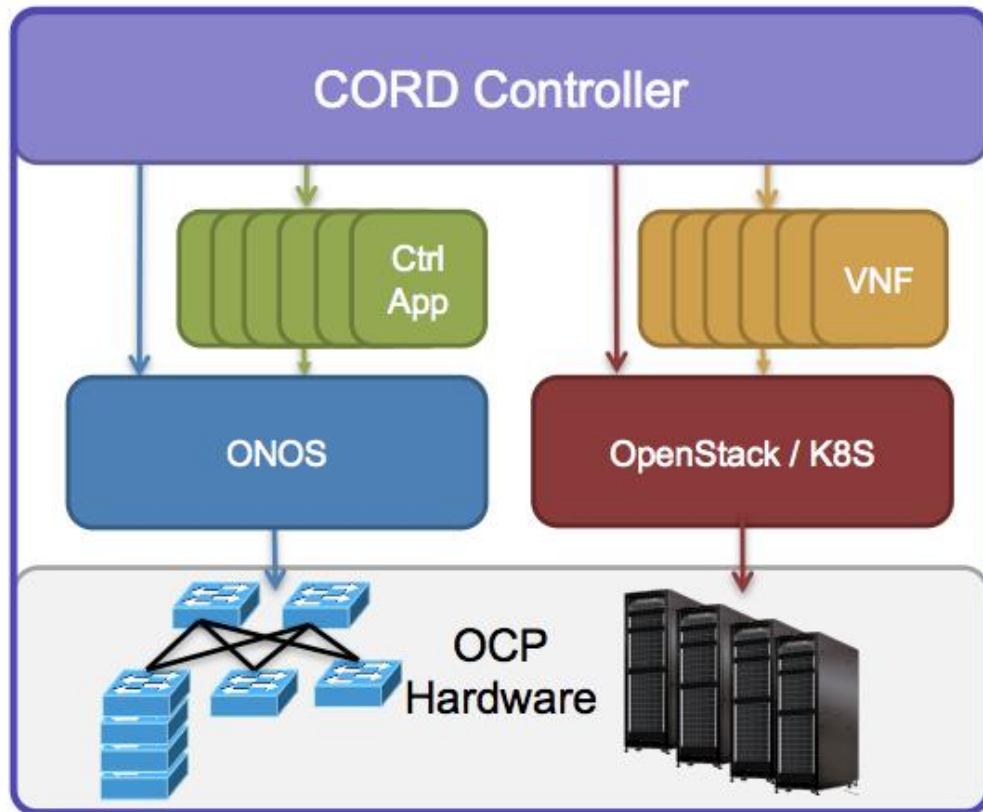


그림 3. CORD architecture

위의 구성 요소들 중, CORD 전체를 제어/관리하는 부분인 XOS는 그림 4와 같이 뷰(View), 데이터 모델(Data model), 그리고 싱크로나이저(Synchronizer)의 계층으로 구분되어 있다. 뷰는 사용자 혹은 운영자와 CORD가 상호작용(Interaction)을 하는 UI(User Interface)를 의미한다. 여기에는 GUI(Graphical User Interface), REST API, TOSCA 등이 포함되어 있다. 데이터 모델은 각 서비스들의 모델을 의미한다. 이 안에는 사용자가 어떠한 서비스를 제공받기 위해, CORD 내에서 어떤 절차대로 수행이 될 것인지, 해당 서비스의 특징은 무엇인지 등과 같은 다양한 정보들이 모델링되어 있다. 마지막으로 싱크로나이저는 모델링된 정보를 실질적으로 서비스 데이터 평면에 내려서 수행하도록 하는 역할을 한다. 이를 위해 싱크로나이저는 각 서비스마다 하나씩 수행이 되며, 사용자 및 CORD 내부의 시스템의 요청을 인지하도록 주기적으로 폴링을 하고 있다. 예를 들어, 어떠한 VNF를 사용자가 생성한다고 가정하자. 그러면, 사용자는 GUI를 통해 VNF 생성 요청을 하게 된다. 요청을 받은 XOS는 해당 VNF를 위한 서비스를 찾게 되고, 이를 참조하여 싱크로나이저는 서비스 내 정의된 일련의 절차를 통해 VNF를 컴퓨터 노드에 배포하게 된다.

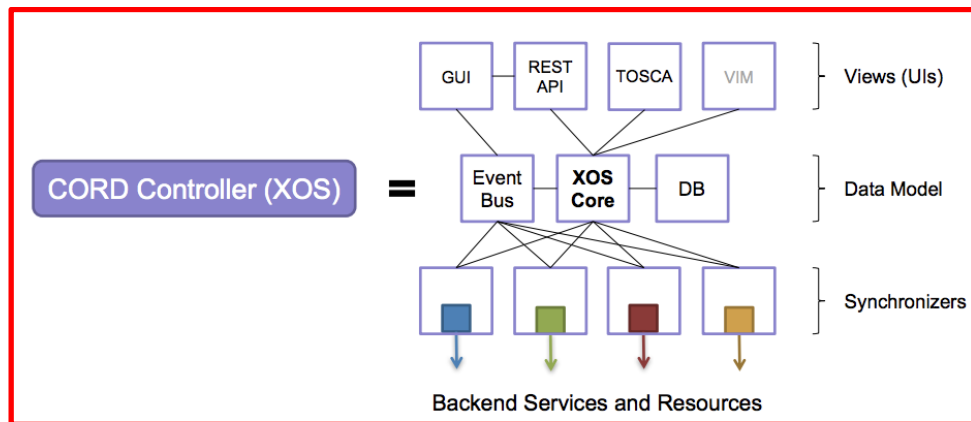


그림 4. XOS architecture

CORD를 사용하기 위해서는 크게 물리적 파드(Physical POD)와 CiaB(CORD-in-a-Box)의 두 타입 중 하나를 선택해서 사용해야 한다. 물리적 파드는 CORD를 실제로 운용할 수 있는 수준의 성능을 가진 물리적 장치들의 집합이다(그림 5). 이를 운용하기 위해, 한 개의 관리 스위치(Management switch), 4개의 패브릭 스위치(Fabric switches), 그리고 3개의 X86 서버가 필요하다. 관리 스위치는 각 장비들의 제어/관리를 위한 관리 네트워크(Management network)를 의미하며, 패브릭 스위치들은 컴퓨터 노드(서버)들 사이의 사용자 데이터 트래픽(User data traffic)을 위한 스위치이다. 일반적인 리프-스파인 패브릭(Leaf-Spine fabric)구조로 서버가 연결되어 있다. 그리고 X86서버 중 두 대는 사용자들의 VNF를 구동하기 위한 컴퓨터 노드이며, 한 대는 CORD 전체를 제어/관리를 위한 다양한 제어 서비스(Control service)들이 동작하는 헤드 노드(Head node)이다.

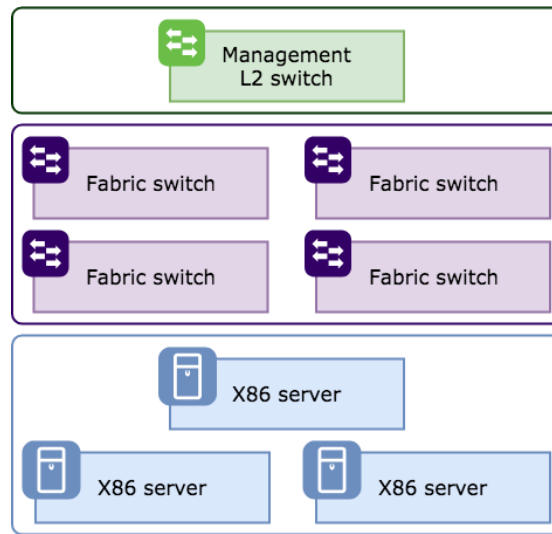


그림 5. Hardware for physical POD

특히 헤드 노드는 컴퓨트 노드 및 패브릭 스위치를 제어/관리하기 위해 관리 네트워크로 다른 모든 노드에 접근할 수 있다. 그림 6은 헤드 노드와 컴퓨트 노드 및 패브릭 스위치 사이의 연결을 위한 개념도이다. 헤드 노드에서 구동 중인 제어 서비스는 MAAS, ONOS, XOS, OpenStack, Kubernetes 등이 동작 중이다. 이러한 서비스들은 컴퓨트 노드의 전원 및 물리적 설정 관리, 패브릭 스위치들의 라우팅 및 데이터 포워딩(Data forwarding), CORD의 전반적인 서비스 관리, 마지막으로 VNF들 및 VNF를 위한 가상 네트워크 관리를 수행한다. 이를 위해서, 헤드 노드는 각 장비들에 관리 네트워크를 경유하여 접속할 수 있도록 설정해야 한다.

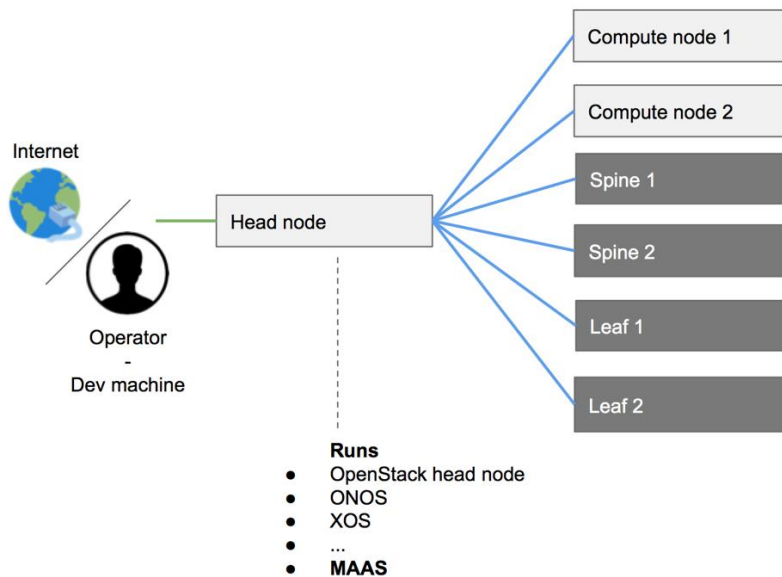


그림 6. Logical configuration in service control plane

한편, 물리적 파드가 실험 및 개발을 위한 용도로는 많은 비용이 드는 이유로, CORD의 개발자 및 연구원들을 위해 CiaB를 제공한다. CiaB는 CORD-in-a-Box의 약자로, 물리적 파드에서 서비스하는 모든 물리적 장치를 한 대의 X86 서버에 에뮬레이션하여 CORD를 구성하는 가상 환경을 의미한다(그림 7). 먼저 관리 스위치를 에뮬레이션하기 위해, X86 서버 내에 리눅스 브릿지를 선언한다. 또한 패브릭 스위치를 에뮬레이션 하기 위해, CiaB는 서버 내에 두 개의 OpenVSwitch를 선언한다. 마지막으로 헤드 노드와 컴퓨트 노드를 에뮬레이션 하기 위해, Vagrant를 사용하여 X86 서버에 가상 머신으로 구동한다. 당연하게도, 헤드 노드 내에는, 물리적 파드와 마찬가지로, 제어를 위한 다양한 서비스들이 동일하게 구동되고 있다.

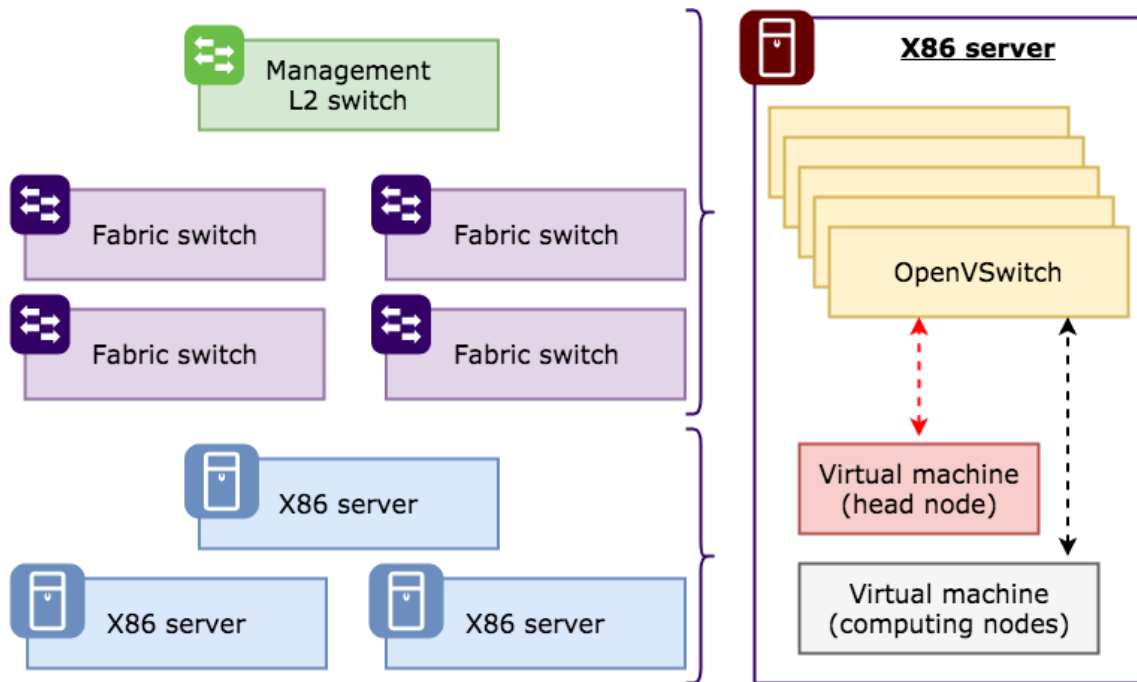


그림 7. Architecture of CORD in a Box

3. M-CORD architecture

기존의 모바일 네트워크를 위한 LTE 네트워크는 중앙 집중식 구조를 가졌다. 그림 8은 초기의 LTE 네트워크 환경을 도식화 한 그림이다. 기존의 LTE 네트워크는 무선 네트워크 부분(EUTRAN; Evolved UTRAN)과 코어 네트워크 부분(EPC; Evolved Packet Core)으로 나눌 수 있다. EUTRAN은 실제 사용자 휴대폰과 기지국 사이의 네트워크를 의미하고, EPC는 기지국으로부터 온 사용자 데이터를 인터넷 혹은 다른 패킷 네트워크(PDN; Packet Data Network)로 전달하거나 사용자의 인증을 하는 역할을 한다. 이를 위해 EPC 내에는 인증 및 제어를 위한 MME(Mobility Management Entity), HSS(Home Subscriber Server) 등이 있으며, 사용자의 데이터를 전달하기 위한 S-GW(Serving Gateway)와 P-GW(PDN Gateway)가 있다. 이 때 EPC 네트워크는 중앙에 위치하며, 통신 사업자가 EPC 내의 MME, HSS, S-GW, P-GW등을 관리하고 있다.

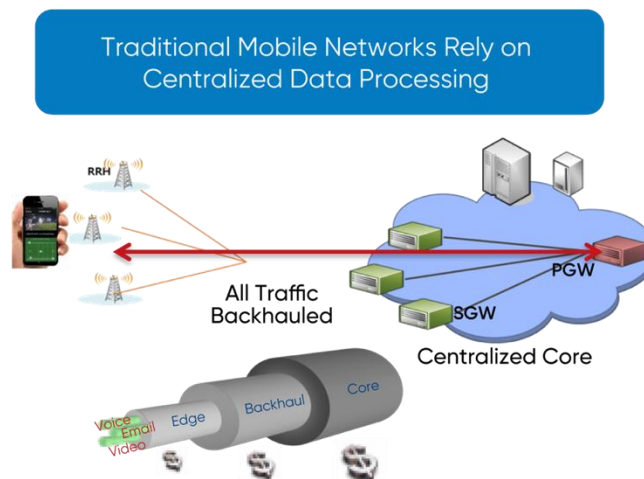


그림 8. Legacy LTE network

M-CORD는 Mobile CORD의 약자로, 기존의 LTE EPC를 구성하는 MME, HSS 등과 같은 다양한 네트워크 기능(NF: Network function)들을 CORD 플랫폼 내에서 구동하는 것이다. 그림 9는 M-CORD를 도식화한 개념도이다. 기본적으로 M-CORD는 CORD 플랫폼 외부의 LTE 기지에서 사용자 트래픽이 수신된다. 이 경우, 기존의 LTE 네트워크는 중앙에 위치한 EPC로 사용자 데이터가 전달되지만, CORD의 경우 중앙이 아닌 분산된 엣지 클라우드로 전송한다. CORD의 개념은 전국에 퍼져있는 전화국을 데이터 센터로 재구성한 프로젝트이다. 이와 같은 CORD 플랫폼에 만일 LTE EPC 내의 NF를 구동한다면, 중앙 집중식 EPC가 아닌 분산된 EPC를 의미한다. 이로 인해, 중앙의 EPC 부하가 분산된 EPC로 분산되는 효과가 있음은 물론

론, 사용자가 EPC까지 도달하는 시간 역시 단축할 수 있는 장점이 있다. 뿐만 아니라, MEC와 같은 다양한 서비스들을 제공할 수 있는 기저가 마련되었다고 할 수 있다.

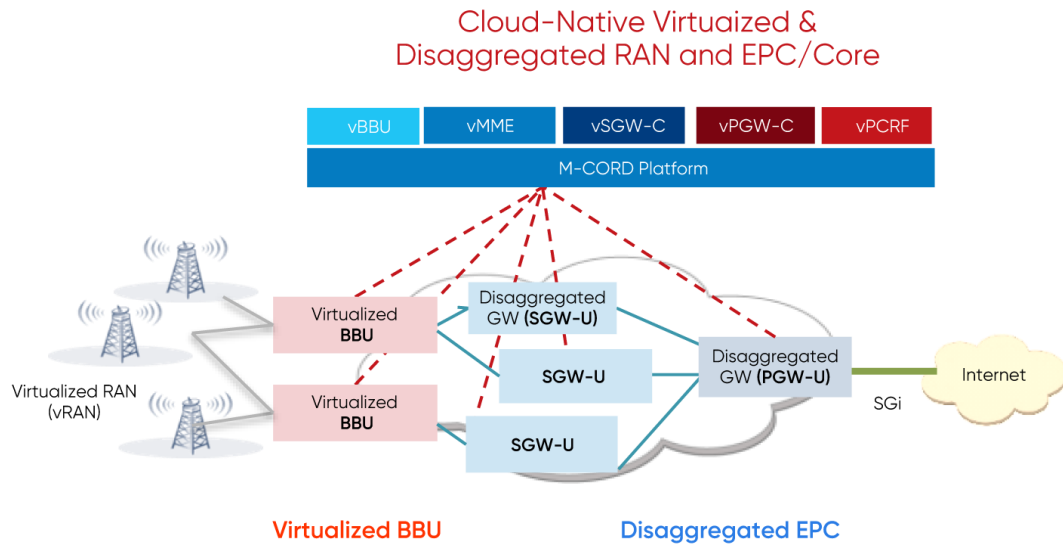


그림 9. M-CORD: an overview

CORD 플랫폼에서 LTE EPC NF를 구동하기 위해서, 기본적으로 EPC NF들의 소프트웨어화 (Softwarization)를 하고 이를 가상화 해야한다. 그림 10은 기존의 LTE EPC 구성 요소이다. 기존의 EPC는 내부의 NF들을 전용 장비로 사용하였다. 하지만 이는 CORD를 설명할 때 언급한 바와 같이, 많은 유지비용과 구매비용이 소비된다. 또한 이를 데이터 센터 내의 범용 장비에서 구동하기란 쉽지 않은 문제가 있다.

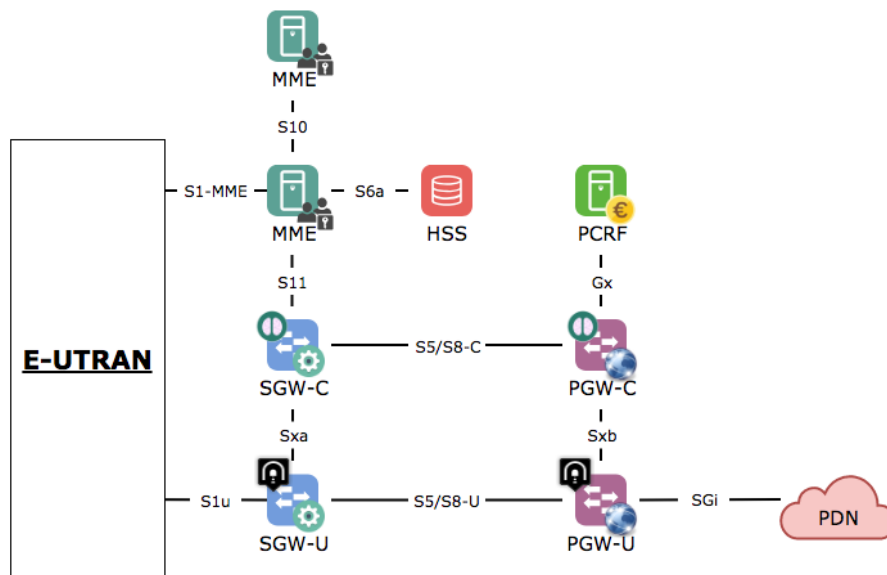


그림 10. Legacy ECP network

따라서 M-CORD에서는 기존의 EPC NF들을 오픈 소스 형태의 소프트웨어로 구현하고, 그림 11과 같이 이를 가상머신(VM; Virtual Machine)이나 컨테이너(Container)의 형태로써 가상화하여 VNF로 생성한다.

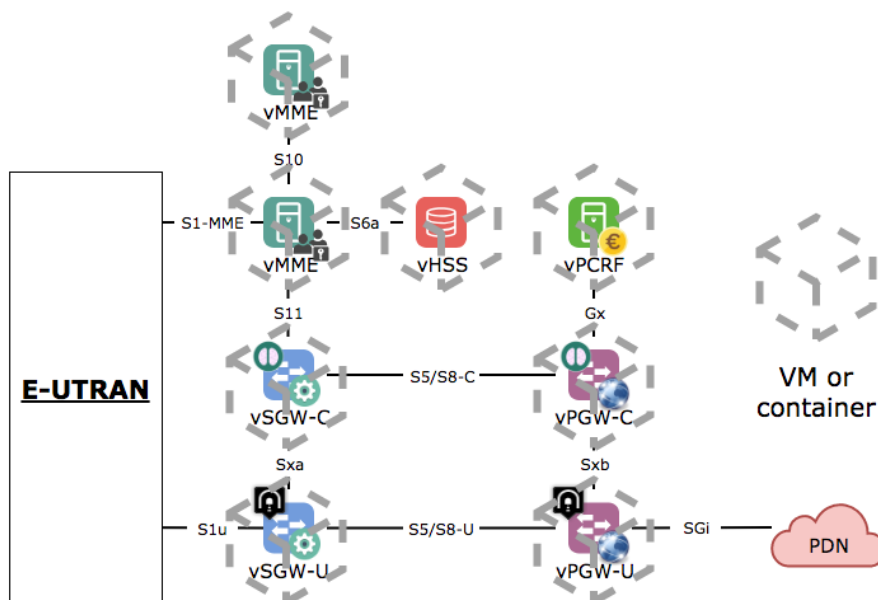


그림 11. M-CORD EPC network

기존의 M-CORD를 위한 CORD 플랫폼은 두 개의 컴퓨트 노드만을 지원하나, 이는 그림 12와 같이 확장될 수 있는 구조이다. 예를 들어, 사용자 급증한 환경, 혹은 다른 다양한 서비스들이 필요한 환경을 가정한다. 이 경우, 두 개의 컴퓨트 노드만으로 원하는 모든 VNF들 및 서비스들을 서비스할 수 없는 문제가 발생한다. 이러한 문제를 해결하기 위해 CORD는 그림 12와 같이 확장된 형태를 제공할 수도 있다.

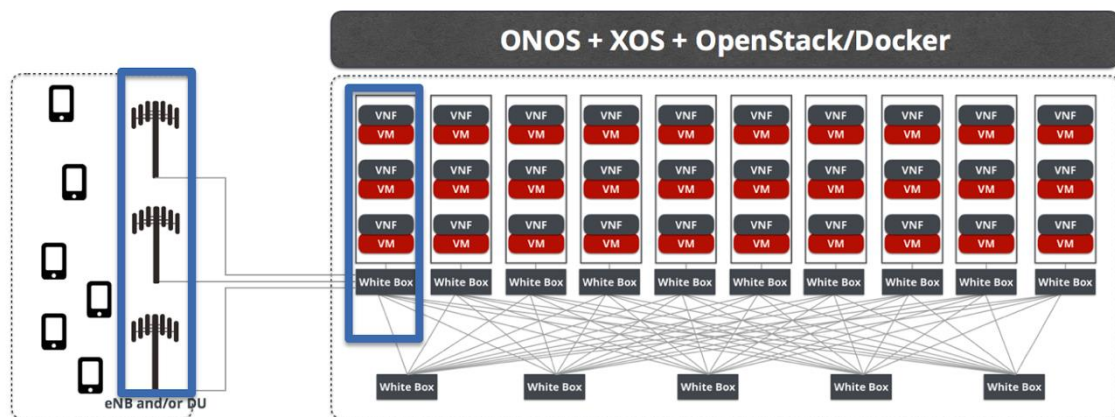


그림 12. Extended M-CORD

M-CORD는 버전에 따라 다른 수의 VNF를 제공한다. 그림 13은 M-CORD 버전 4.1이고, 그림 14는 M-CORD 버전 5.0의 서비스 체인(Service chain)을 도식화 한 그림이다. 버전 4.1의 경우, 총 3개의 VNF들을 가지고 있으며, 버전 5.0의 경우, 6개의 VNF들을 서비스하도록 한다. 4.1의 경우 S-GW와 P-GW를 SPGW로 통합한 후, 제어 평면만 분리한 SPGW-C와 사용자 평면만 분리한 SPGW-U가 있다. 그리고 SPGW-C와 SPGW-U를 검증 하기 위한 eNB 에뮬레이터가 존재한다. 이 에뮬레이터는 NG4T사에서 개발된 소프트웨어로, 내부에는 MME, HSS, eNB, 그리고 인터넷을 에뮬레이션 하도록 설계되어있다. 이들 사이의 연결은 OpenStack Neutron에서 생성한 네 개의 가상 네트워크로 구성이 된다. 각 가상 네트워크의 이름은 LTE 표준 인터페이스를 참조하여 명명되었다.

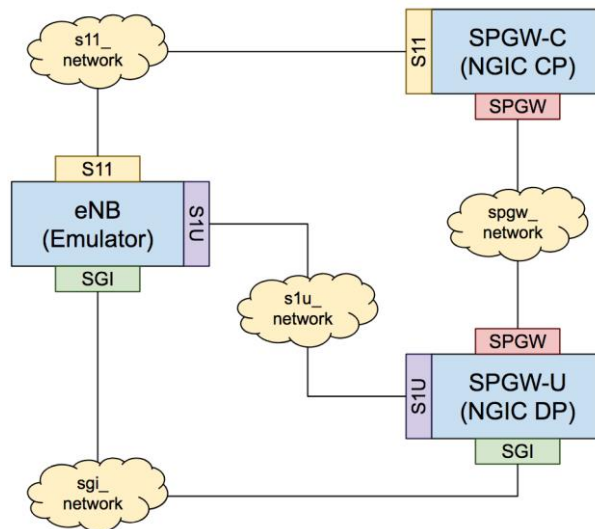


그림 13. Service chain for M-CORD 4.1

한편 버전 5.0의 경우, 4.1과 동일하게 SPGW-C와 SPGW-U가 있으며, 기존에 에뮬레이션 하던 것을 실제 VNF로 대체하였다. 버전 4.1의 eNB 에뮬레이터에서 동작하던 MME와 HSS를 실제 사용이 가능한 VNF로 대체하였으며, HSS에서 사용자 검증 및 인증을 하기 위한 사용자 정보들은 카산드라 데이터베이스(Cassandra Database)라는 VNF에 저장되었다. 인터넷 트래픽 및 비디오 스트리밍을 위해 인터넷 에뮬레이터가 존재하며, 기존에 eNB가 에뮬레이션 하던 EUTRAN 트래픽은 실제 eNB 하드웨어 및 스마트폰을 사용하도록 대체되었다.

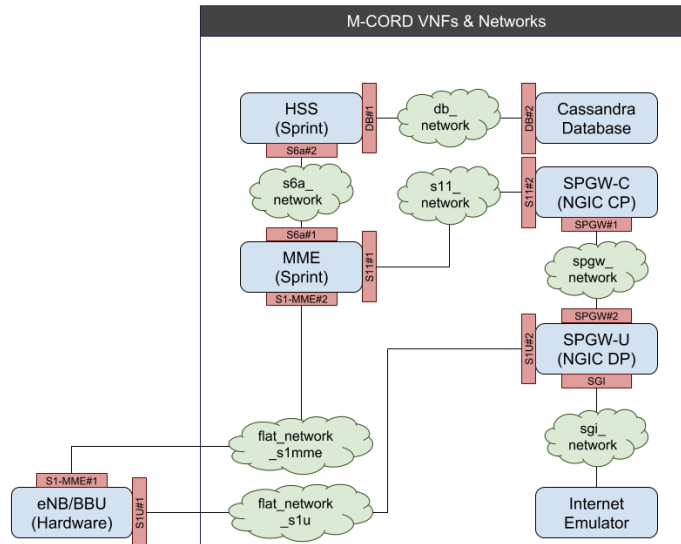


그림 14. Service chain for M-CORD 5.0

4. A Design of M-CORD monitoring system

이와 같은 M-CORD 환경에서, 우리는 M-CORD를 위한 모니터링 시스템을 디자인한다. 그림 15는 M-CORD 모니터링 시스템에 대한 디자인 개념도이다. 앞서 설명한 바와 같이, M-CORD 내에는 헤드 노드와 컴퓨트 노드들로 구성되어 있다. 컴퓨트 노드들에는 M-CORD에서 사용하는 다양한 VNF들이 구동되고 있으며, 컴퓨트 노드가 VNF들을 CORD의 패브릭 스위치와 연결하기 위해 OpenVSwitch를 선언하고 모든 VNF들을 연결하였다.

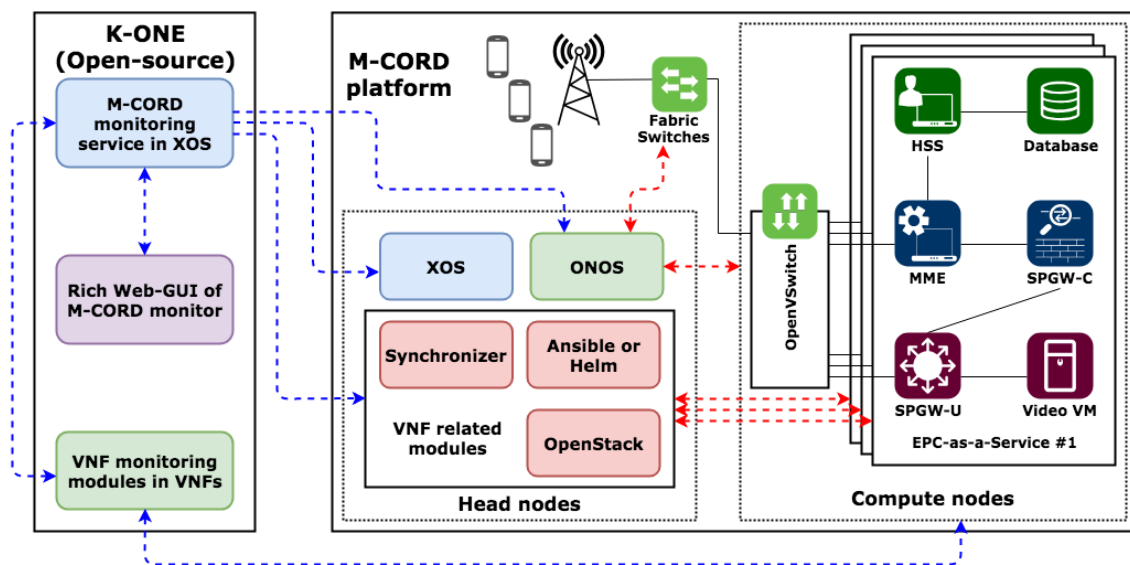


그림 15. M-CORD monitoring system with M-CORD 5.0

헤드 노드 내에는 이러한 OpenVSwitch와 패브릭 스위치를 제어하기 위한 ONOS 컨트롤러들이 구동 중이며, ONOS 컨트롤러는 두 개의 어플리케이션을 구동하고 있다. 첫 째로, ONOS 컨트롤러는 OpenVSwitch를 제어하여 M-CORD 내의 가상 네트워크를 제어하도록 ONOS-VTN이라는 어플리케이션을 사용한다. 두 째로, M-CORD 내의 패브릭 스위치들을 제어하기 위한 ONOS-Fabric이라는 어플리케이션을 사용한다. 당연하게도 M-CORD 내에는 M-CORD 전체의 서비스를 제어하기 위한 서비스 제어 평면인 XOS가 위치하며, 이 밖에도 싱크로나이저, 자동 설정 및 실행을 위한 앤서블(Ansible) 혹은 헬름(Helm)이 있다. 마지막으로 VNF들을 배포하고 가상 네트워크 생성을 위한 OpenStack이 위치한다.

이와 같은 환경에서, M-CORD의 모니터링 시스템을 구축하기 위해, 우리는 세 가지의 모듈화된 서브 태스크들을 정의한다. 첫 째는 XOS상의 M-CORD 모니터링 서비스 정의, 둘째는 Web GUI, 마지막으로 셋 째는 VNF 내부를 모니터링하는 에이전트 구현이다.

1. XOS상의 M-CORD 모니터링 서비스 정의는 M-CORD 내에 모니터링 서비스에 대한 모델링을 수행하는 과정이다. 이 모듈은 XOS상에 정의되며, ONOS 및 VNF를 제어하는 모듈들에 접근하여 가상 네트워크, 패브릭 스위치 상에서 오가는 네트워크를 모니터링 할 수 있도록 한다. 이들 사이의 통신을 위해서 우리는 REST API를 사용하도록 정의한다. 다음으로 이 모듈은 컴퓨터 노드 및 헤드 노드의 CPU, 메모리, 그리고 디스크 사용량을 측정할 수 있도록 한다. 이를 위해, 컴퓨터 노드 및 헤드 노드에 파이썬 기반의 CollectD 어플리케이션을 설치한 후, REST API를 통해 이들 모니터링 정보를 가져온다. 마지막으로 이 서비스는 XOS Web GUI상에서 기본적인 모니터링 정보를 시각화(Visualize)하도록 한다.
2. 다음으로 Web GUI는 XOS 상에서의 시각화보다 더 많은 정보를 자세하게 표현하기 위한 모듈이다. 이는 헤드 노드 상에서 Docker 컨테이너 형태로 구현이 될 예정이며, Web GUI를 구동하는 컨테이너가 XOS 상에서의 모니터링 정보를 주기적으로 폴링하여 정보를 획득한 후, 그래프화하는 모듈이다.
3. 마지막으로 VNF 내부를 모니터링 하는 모듈이다. 이 모듈은, 각 VNF들에서 구동되는 에이전트로서, LibPCAP이나 기타 다른 라이브러리를 활용하여 VNF 내에 어떠한 타입의 패킷이 송/수신되는지를 실시간으로 모니터링 한 후, 이 정보를 Web GUI로 전달하도록 하는 역할을 한다.

이들 시스템을 통해 기대할 수 있는 서비스는 다음과 같다. 먼저 M-CORD를 구성하는 다양한 노드들의 전반적인 컴퓨팅 및 네트워킹 자원들을 시각화 할 수 있다. 이는 네트워크 운영자로 하여금 M-CORD의 상태를 쉽게 인지할 수 있도록 한다. 이를 확장하여, 만일 M-CORD 내부의 자원이 부족하거나 과도히 사용되는 경우 능동적으로 이 자원을 조절하여 확장성(scalability)을 제공할 수 있는 장점이 있다. 또한, 전반적인 자원의 시각화뿐만 아니라, 현재 VNF들이 어떠한 트래픽을 송/수신하는지를 확인할 수 있다. 이를 통해 외부에서 정상적이지 않은 트래픽이 접근한다거나, 인증 받지 않은 사용자가 EPC 네트워크로 접근을 요구하는 등의 보안을 위한 트래픽 모니터링을 쉽게 할 수 있다. 마지막으로, 위에서 제공하는 서비스들을 정책 기반(Policy-based)이 아닌 머신 러닝(ML; Machine Learning) 및 인공지능 기반으로 수행할 수 있다. 이러한 수행을 위하여 M-CORD 모니터링 시스템은 기저가 되는 역할을 수행 할 것으로 기대한다.

5. Conclusion

본 기술문서에서는 CORD와 M-CORD의 전반적인 내용을 다루고, 특히 M-CORD 환경에서 현재 부재하는 모니터링 시스템을 어떻게 구축할 수 있는지 디자인에 대해 제안하였다. 이를 활용하여, 기존에 M-CORD에서 사용 중인 다양한 각종 물리적인 노드들의 CPU, 메모리, 디스크자원은 물론, 각 VNF들에서 어떠한 트래픽이 오가는지에 대한 모니터링할 수 있는 디자인을 제안하였다. 이를 바탕으로 우리는 실제 M-CORD 버전 5.0 및 최신 버전인 6.0을 위한 M-CORD 모니터링 시스템을 구현하여 공개할 예정이다.

K-ONE Technical Document

- It is not prohibited to modify or distribute this documents without a permission of K-ONE Consortium.
- The technical contents of this document can be changed without a notification due to the progress of the project.
- Refer website below for getting more information of this document.

(Homepage: <http://opennetworking.kr/projects/k-one-collaboration-project/wiki>, E-mail: k1@opennetworking.kr)

Written by K-ONE Consortium

Written in 2018/12