Software Defined Architecture for Secure Video Streaming: CORD Based Secure Video Streaming

Home Based Healthcare using Multimodal Sensors

Aman Maldar
Department of Electrical and Computer Engineering
University of Massachusetts, Lowell
Lowell, MA, USA
Aman_Maldar@student.uml.edu

Priyanka N. Murthy
Department of Electrical and Computer Engineering
University of Massachusetts, Lowell
Lowell, MA, USA
Priyanka_Murthy@student.uml.edu

Abstract— **Real-time home based healthcare monitoring system requires having very less latency, privacy of patient's data as well as easy installation at the home. This project focuses on implementing patient monitoring service to stream the data from patient's home to the monitoring station. At the same time, project focuses on reducing the latency by processing the data at the central office (TELCO) itself rather than sending the data to the cloud based services. In order to achieve these objectives, project makes use of central office which is Rearchitectured as data center. This architecture ensures that, most of the critical requests are served at the central office and eliminates accessing the cloud base services.**

The project aims at implementing home based healthcare using multimodal sensors (wearable, webcams, etc.)This project focuses on understanding and implementing advanced concepts in the subject Software Defined Networks. This project plans to give extensive knowledge of Central Office Rearchitectured as Datacenter (CORD architecture), installation of CORD on CloudLab, running streaming service on the CORD. Project also intends to provide general understanding of Mist Server/Red5, XOS and media streaming service. This project involves designing Hello World application on CORD, using CloudLab to instantiate CORD, designing a video streaming service on CORD by running Red5 in one virtual machine, performing experiments to evaluate the performance of video streaming (latency, CPU utilization, etc.)

***Index Terms*— Central Office Re-Architectured as Data Center, CORD, XOS, Ansible Playbook, ONOS, Home care, Openflow, SDN, NFV**

I. INTRODUCTION

Home care services are growing up in the past years. Contemplating the patient/family pair, it represents a solution to the medical problems of the modern life. With the social trends, the senior population has been increasing in the last years. However, as living is more stressful than ever, there are more cases of chronic diseases. The difficulties of transport in the big cities and the scarcity of hospital streambeds turn the home care an attractive solution. However, its routines can be switched by telemedicine.[1] A system can be implemented at home which will facilitate remote monitoring of the patient. The system consists of volume of data which is transferred over the internet. System could collect data associated with patient movement, or data from sensors mounted on patient's body. System should provide a mechanism to process the data on priority. This medical data should be processed with less latency. This can be achieved by using available infrastructure at the central office. Consider an example of stroke rehabilitation/monitoring application. The cameras are installed to send stream to the monitoring station. The security and privacy of patient data are important. If the stream processing is done at the central office, rather than processing the stream in the cloud, it makes it more fast to trigger response in case of mishap. The project leverages a home based computer (for sensor data aggregation) , the CORD services at the edge of the carrier network, and the software defined infrastructure at a cloud data center (e .g. CloudLab). The CORD provides many opportunities to apply the security and privacy protection policies. All the mentioned objectives are focusing on using the infrastructure at the central office.

II. CORD

CORD (an acronym for Central Office Re-architected as Datacenter) is an end-to-end solution POC for the ONOS project spanning Telco Central Office, Access (GPON, G.Fast), Home/Enterprise that combines SDN, NFV, Cloud with commodity infrastructure and open building blocks to deliver data center economies of scale and cloud-style agility to service provider networks.[2] CORD is a service provider driven (AT&T) collaborative effort to build an end-to-end SDN/NFV/Cloud solution POC with best-of-breed, open building blocks.

CORD enables service providers to build an underlying common infrastructure with white boxes, ONOS, OpenStack, and XOS with a diversity of organizations building the services and solutions that ride above. CORD solution POC is driven by a partnership comprised of AT&T, ONOS, ON.Lab, PMC Sierra and Sckipio. [2] CORD addresses two overarching problems.

Today's Telco Central Offices (COs) are a huge source of CAPEX/OPEX and their design/infrastructure is not geared towards programmability and flexibility. This is due to fragmented, non-commodity infrastructure in today's Central Offices where each site hosts more than three hundred unique

deployed appliances, each requiring a physical install and specialized management. The question then is how do you bring in the cost-performance leadership and agility of, say, Google or Facebook to Telco networks? This is the first problem CORD aims to address. [2]

Secondly, a number of existing SDN/NFV solutions have virtualized the CPE or virtualized the appliances or a subset of functionality/devices. However CORD aims to look at things holistically in order to deliver maximum benefits of SDN, NFV and Cloud to Telcos instead of providing piecemeal virtualization. CORD delivers an end-to-end SDN/NFV/Cloud solution POC that encompasses the subscriber CPE, a diversity of access technologies (including GPON and G.fast) and the Telco Central office. It is the first end-to-end solution POC to include virtualized OLT along with Openflow-enabled G.fast [2]

III. CORE BUILDING BLOCKS OF CORD

CORD enables building an underlying common infrastructure with white boxes, ONOS, OpenStack, and XOS and enables a variety of solutions above this. A diversity of organizations can leverage this and build the services that ride on this common infrastructure. CORD solution uses open, best-of-breed software and hardware building blocks as shown below,

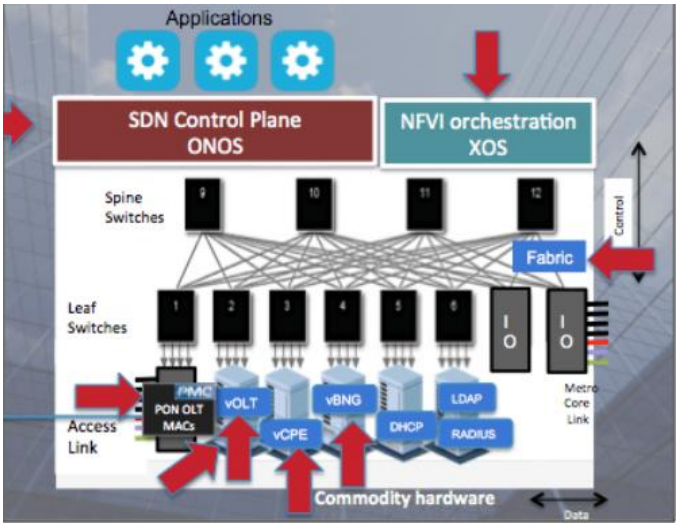


Figure 1: Basic Building Blocks

CORD key components include commodity hardware, SDN control plane (ONOS), NFVI orchestration (XOS, Openstack), Open Leaf spine fabric, Simple on plane CPE+ vCPE, Virtualized access (PON OLT Mac + vOLT), Virtualized functions, Virtualized BNG

CORD enables building an underlying common infrastructure with white boxes, ONOS, OpenStack, and XOS and enables a variety of solutions above. A diversity of organizations can leverage this and build the services that ride on this common infrastructure. This is aligned with ONOS's goal of bringing the benefits of real SDN to service provider networks. [2]

XOS:

XOS builds on two open source projects. The first is OpenStack, which manages virtual resources on a cluster of commodity servers. OpenStack is responsible for creating and provisioning virtual machines (VMs) and virtual networks (VNs), while XOS defines a service abstraction on top of those virtual resources. The second is ONOS, which manages the cluster's switching fabric. ONOS hosts a collection of network control applications, while XOS incorporates those applications into the overall portfolio of CORD services. [3]

XOS provides explicit support for multitenant services, making it possible to create, name, operationalize, manage and compose services as first class operations. Commodity multi-tenant clouds are designed to host applications, but they usually treat these as single tenant services that run on top of the cloud, for the benefit of the user. In contrast, XOS provides a framework for implementing multitenant services that become part of CORD, thereby lowering the barrier for services to build on each other. [3]

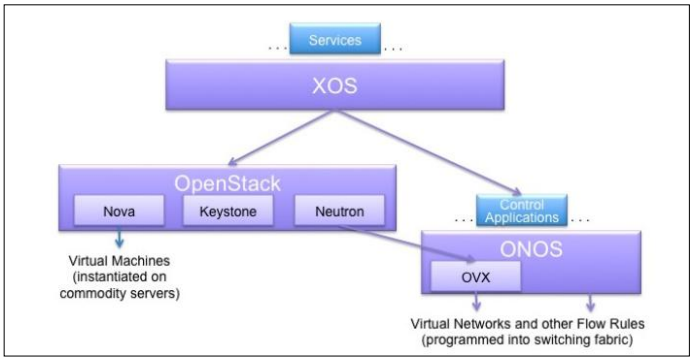


Figure 2: Relationship between XOS, Openstack and ONOS

Figure 2 gives a high-level depiction of XOS, OpenStack, and ONOS, including their critical subsystems. As shown in the figure, XOS supports a set of services and ONOS hosts a set of control applications. OpenStack's Neutron sub system uses ONOS's OVX subsystem to embed virtual networks in the underlying fabric. (In effect, Neutron can be viewed as one of the control apps running on ONOS.) [3]

One of the main values that XOS brings to CORD is a set of abstractions that unify three related but distinct threads: [3]

- The first is SDN, which is about separating the network's control and data planes. This makes the control plane open and programmable, leading to increased innovation. It also allows for simplification of forwarding devices that can be built using merchant silicon, resulting in less expensive white box switches.
- The second is NFV, which is about moving the data plane from hardware appliances to virtual machines. This reduces CAPEX costs (through server consolidation and replacing high margin devices with commodity hardware) and OPEX costs (through software based orchestration). It also has

the potential to improve operator agility and increases the opportunity for innovation.
- The third is the Cloud, which defines the state of the art in building scalable services—leveraging software based solutions, virtualized commodity platforms, elastic scaling, and service composition, to enable network operators to rapidly innovate.

IV. DESIGN

The application comprises of user application sending data to the CORD as well as multiple services running on the cord.

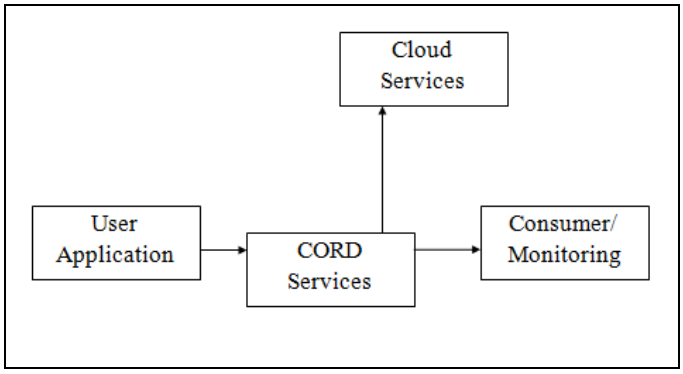


Figure 3: Design Concept

Figure 3above shows the design concept used to implement the project.
User Application: The computer based system located in the user's home will have cameras and sensors connected to it. This data/video stream will be continuously transmitted to the CORD services running at the central office.

Receiving Service: The user application sends video stream packets to the service (S1) running on the CORD. Receiving service at the CORD will receive the data stream and store it for the purpose of processing.

Data Processing: There could be another service created at the CORD which will work on encrypting the incoming stream of data. This service should be capable of extracting features from the incoming stream. For instance, if processed data stream reveals signs of stroke mishap, service should be able to inform emergency team. Furthermore, the encryption of the data stream is needed, so that this stream can be forwarded to monitoring team over the internet.

Transmitting Service: Consumer will request the stream from service (S2) running on CORD. Service 2 will transmit the encrypted data stream over the internet to the consumer.

Cloud Services: Furthermore, the architecture should provide support for further data processing at the cloud in the conventional way.

Above implementation strategies speak about creating the separate services in the CORD to handle all the functionalities. However, a dedicated server, for example Red5 / Mist server can be installed on the CORD which will provide all these functionalities. This server can be used for streaming service.

The project involves following design steps.

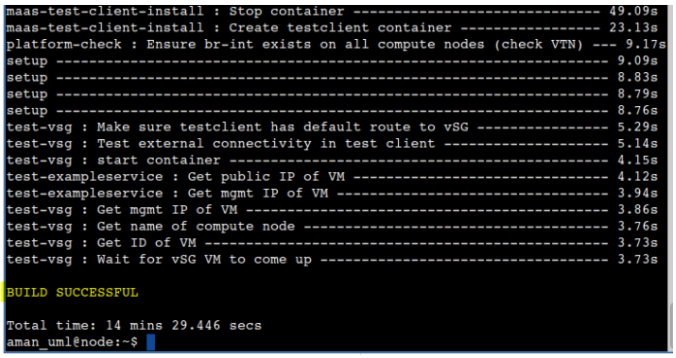
- Setting up CORD Environment on CloudLab.
- Running Hello World Service
- Reusing the template of HelloWorld service to create new tenant service
- Setting up Red5/Mist Server on CORD
- Creating a Video streaming service on CORD
- Writing an application in python to send the stored video stream to CORD service (S1)
- Writing a service (S1) in XOS controller to handle incoming packets
- Writing a service to encrypt the user data stream
- Writing a consumer application to read the video stream from CORD service (S2)
- Creating a service (S2) to serve the incoming consumer request
- Observe Video streaming performance metrics like latency, CPU Utilization.

V. IMPLEMENTATION/ WORK DONE

We completed the first 3 objectives. We used CloudLab to setup the CORD-POD. *There is a separate file in the github repository to show the installation steps and screenshots.[4]*
We executed the example service to show Hello World message. We made changes to create a tenant service, using the template of HelloWorld service. *There is a separate file in the github repository showing the details and screenshots [4][5].*

VI. RESULTS

Screenshot below shows the CORD installation on CloudLab.



Screenshot below shows the execution of example service.

```
vagrant@prod:~$ sudo lxc exec testclient -- /bin/bash
root@testclient:~# curl http://10.6.1.194
ExampleService
 Service Message: "hello"
 Tenant Message: "world"

root@testclient:~#
```

Screenshot below shows the result of changes made in HelloWorld template. The output of the tenant service is shown.

```
vagrant@prod:~/service-profile/cord-pod$ sudo lxc exec testclient -- /bin/bash
root@testclient:~# curl http://10.6.1.194
ExampleService
 Service Message: "hello"
 Tenant Message: "universe"

root@testclient:~# curl http://10.6.1.195
ExampleService
 Service Message: "hello"
 Tenant Message: "world"

root@testclient:~#
```

VII. TROUBLESHOOTING

Most challenging part was to install CORD, which took several hours. Any changes made CORD in the python script of HelloWorld application would break the CORD architecture production environment. We learnt to design and implement HelloWorld Application. We learnt to make changes in HelloWorld application to use as a Service.

Changes made in the Ansible script were not executed properly. We studied logs of the container to understand sequence of operations.

VIII. CONTRIBUTIONS

The contribution part mainly involves understanding different concepts in the CORD architecture and discussing their role in implementation of the application. We made efforts towards developing small parts in the application, we faced problems, we analyzed the issues to understand how things work.

Contributed together on

- Understanding advanced concepts like XOS, ONOS, Service assembly
- Implement the application as a whole.

Individual contribution

Aman Maldar

- Read service assemble documentation and executed HelloWorld application.
- Reused HelloWorld template to create tenant service.
- Executed multiple services to display results
- Created the documentation for example service demonstration.
- Created documentation for CORD environment setup using CloudLab.
- Made initial installation of red5 and mist server on the Linux for testing purpose.
- Wrote a sample python application (uppercase program) to install as a service in CORD.
- Made changes in the Ansible playbook to execute sample python application.

Priyanka Murthy

- Understood the installation process of CORD
- Set up the environment for application development.
- Managed the github repository to keep track of changes in different files/folders.
- Worked on troubleshooting logs of container to understand execution of Ansible script.
- Understood the role of docker container, VM in implementing the services.
- Helped in creating final report by compiling the details.
- Worked on creating presentation slides.

ACKNOWLEDGMENT

We are thankful to Dr. Yan Luo, Dr. Peilong Lee, Mr. Mian Ibrahim and Mr. Chen Xu for all the guidance.

REFERENCES

[1] Mobile Telemedicine System for Home Care and Patient Monitoring - M. V. M. Figueredo, J. S. Dias

[2] XOS Project - http://xosproject.org/wp-content/uploads/2015/06/CORD-FAQs.pdf

[3] XOS Service Orchestration for CORD – White Paper.

[4] Github repository link for all project details- https://github.com/amanmaldar/EECE7290_Project

[5] Github repository link for all project details- https://github.com/priyanka-N-Murthy/EECE-7290-Software-Defined-Networking-Project