UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Department of Computer Science
Computer Networks Research Group

Project Plan

Authors

Arkajit Ashwin
Bhargavi Deeksha
Harshitha Sanket
Suheel Vivek

Supervisors:
Sevil Mehraghdam Hadi Razzaghi
Holger Karl

Paderborn, November 28, 2018

Contents

1 Motivation

1.1 Motivation

The rapid growth of mobile data services driven by mobile internet has led to a substantial challenge of high availability , low latency, high bit rate and performances in networks. The recent development of NFV and SDN has emerged as a key enabler for 5g networks. There has been a paradigm shift in networking with the recent developments in Network Virtualization technology. Network Function Virtualization (NFV) involves decoupling of the hardware components from the software components of a network function. These NFVs require some central management and orchestrations framework (MANO) in order to fully deliver end to end services of an application. There are multiple open source as well as industrial implementation of a MANO framework in the market.

End to end network service delivery with high avaliability , scalability and low latency requires chaining of the Network Functions across different Internet Service Providers (ISPs) which in turn have their own MANO frameworks. In order to seamlessly create a Network Sevice by utilizing the Virtual Network Funtions within each of the MANO frameworks, a common ground needs to be setup based on which the VNFs could be optimally linked to create a virtual network service.

1.1.1 Problem Description

European Telecommunicaitons Standards Institute (ETSI) defines the reference architecture for a MANO framework. Each network service is composed of multiple network funtions virtually linked and orchestrated by a MANO framework. The network services require a descriptor which contains the details of all the VNFs , virtual Links , Forwarding Graph of VNFs are defined. Each of the Virtual Network Funtions contain its own descriptors namely the number of virtual machine it needs etc. Different MANO frameworks have its own Descriptor Schemas pertaining to the standard defined by ETSI. This framework specific Network Service Descriptor hinders the orchestration and management of NFVs between different MANO frameworks. Our project aims at tackling this problem by implementing a method to translate and split the NFVs from different MANO frameworks in order to create a framework independent Network Service chain.

2

Goals and Use Cases

2.1 Goals

In this section, we describe the goals of the project.

The structure of the project consists of a component that manages and orchestrates the network (Management and orchestration,MANO) and Network Service Descriptor (NSD) that specify the network service to be created and it is a collection of configuration documents which determines how the network service is comprised in terms of Virtual Network Functions (VNFs).

The main goal of the project is to build three work packages:

1. Service Descriptor Translator(SDT) : SDT translates Network Service descriptor (NSD) of one MANO framework (mano1) to another MANO framework (mano2). In the structure given, translator is required if mano2 needs to deploy the services described by NSD of mano1.

2. Service Descriptor Splitter(SDS) : In order to deploy VNFs over different Point of presence (POPs), splitter is required.

3. MANO scalability support : Adaptor is necessary if more than one instance of MANO framework is needed, i.e, adaptor allows interaction between different instances of MANO frameworks. It also exposes the underlying MANOs service instantiation interfaces and retrieves monitoring information about the service status.

Figure 2.1: This figure visualizes the structure of the project.

2.2 Use Cases

«««< Updated upstream

2.2.1 Cross MANO Framework interaction

As a matter of fact , the MANO frameworks used by every network service provider varies from one another. Network service Descriptor translation enables the deployment of network services that is in accordance with the intended framework

For instance : If an operator uses Sonata framework and another operator uses OSM operator, the NSD schemas for both the frameworks will be different. Using the solutions of translation and splitting, network services can be deployed and orchestrated across MANO implementations.

2.2.2 Hierarchical orchestration

By implementing MANO adaptor , dynamic instantiation and inter-operability between different MANO frameworks can be achieved. As a result of this goal, operators will be able to scale up and scale down the resources as and when required. Also the operator will be able to handle the resources in an efficient manner. When there is a high demand for network service , the operator can explore options to include additional MANO instances to mitigate the traffic load on a single MANO instance.The resources can be provisioned based on the amount of requests. This helps the operator in extending profitability. =======

»»»> Stashed changes

3

Non Functional Requirements

The important non-functional requirements for our work packages are defined here. Below is the overview over some requirements and describe their meaning for our project in particular.

1. *Scalability*: One of the main requirements of any cloud-based networking, is its ability to adapt its throughput to varying load. Here, the load of particular network functions, load on the translator, splitter is also considered as an add on to the overall load of the system. One of the main work packages mentioned in this project is the scalability support for the MANO framework. The adapter plays a major role in scaling up the defined system.

2. *Reliability*: The requirement of reliability aims at the ability of the system to provide continuous correct service. It is the continuation of the service in compliance with the service specification. It is one of the weak requirements that has to be met at any case. In this project, this can also be viewed as providing service for the required duration and then terminating.

3. *Availability*: This term can be paraphrased as *readiness for correct service*, i.e., delivery of service in compliance with the service specification. Here, the work packages should be available for translation or for splitting of the NSDs.

4. *Flexibility*: This terms defines the ability to accomodate the changing requirements. Here in this project, the system need to accomodate the MANO framework and its instances.

5. *Performance*: This term summarises the possible aspects of the system. In this network scenario , we will basically consider throughput and latency of all the network functions involved, as a measurement of its efficiency.

4 Technologies

4.1 Orchestrator

4.1.1

4.1.2

4.1.3

4.1.4

4.2 Virtualized Infrastructure Manager (VIM)

VIM is one of the three functional blocks specified in the Network Functions Virtualization Management and Orchestration (NFV-MANO) architecture. VIM is responsible for controlling and managing the NFV infrastructure (NFVI), by provisioning and optimizing the allocation of physical resources to the virtual resources in the NFVI. Performance and error monitoring is also a key role of the VIM. Popular VIMs are discussed in the following sections.

4.2.1 OpenStack

OpenStack[1] is a community-driven open source cloud resource management platform. Compute, storage, and networking resources in a datacenter can be provisioned using application program interfaces (APIs) or web dashboard provided by OpenStack component, for instance, NOVA is a component which can provide access to compute resources, such as virtual machines and containers. Network management is enabled by NEUTRON component which handles the creation and management of a virtual networking infrastructure like switches and routers. SWIFT component provides a storage system. By making use of many such components, OpenStack can deliver complex services by utilizing an underlying pool of resources.

4.2.2 Amazon Web Services (AWS)

AWS[2] is a cloud computing platform, It offers (1) Infrastructure as a Service (IaaS) – provides resources that can be utilized for custom applications (2) Platform as a Service (PaaS) –

[1] https://www.openstack.org/
[2] https://aws.amazon.com/

provides services such as database and email which can be used as individual components for building complex applications (3) Software as a Service (SaaS) – provides user applications with customization and administrative capabilities that are ready to use. AWS is specially preferred by small companies for the flexibility and ease of use of its cloud infrastructure.

4.2.3 Kubernetes

Kubernetes[3] (K8s) is an open-source platform for automation and management of containerized services, it manages computing, networking, and storage infrastructure. Kubernetes was initially developed by Google and now under Cloud Native Computing Foundation. Kubernetes Architecture consists of (1) Master server components – it is the control plane of the cluster and act as the gateway for administrators (2) Node Server Components – servers which are performing work by using containers are called nodes, they communicate with the master component for instructions to run the workload assigned to them. Kubernetes provides comprehensive APIs which are used to communicate between the components and with the external user.

[3]https://kubernetes.io/

5 Related Work

In this chapter we discuss the relevant research efforts that can be used to achieve our goals. First we discuss the standards and specifications for orchestration and management of NFV in the section 5.1. The fundamental aspect of a service deployment is the Network Service Descriptor (NSD), in the section 5.2 we discuss the trends and options of NSDs. In section 5.3, we discuss research papers that try to mitigate the interoperability challanges between different MANO frameworks. Section 5.4 will be a brief account on the MANO scalability problem.

As this is our initial project proposal, the state-of-the-art could change as we progress and we will update our approach accordingly.

5.1 Standards and Specifications

[?] [?] [?]

5.2 Service Description

[?]

5.3 MANO Interoperability

5.4 Hierarchical Orchestration

MANO framework face significant scalability challenges in large scale deployments, the amount of infrastructure a single instance of MANO framework can manage is limited. Abu-Lebdeh et al. [?] explores the effects of placement of MANO on the system performance and scalability and conclude by suggesting hierarchical orchestration architecture to optimize them, they formally define the scalability problem as an integer linear programming and propose a two-step placement algorithm. We also intend to answer the MANO scalability challenges, by exploring the optimal number of MANO deployments in a system, the optimal hierarchical level and how to manage the state of such a system dynamically.

6 Project Time Plan

6.1 TIMEPLAN

In this section we shall discuss the timeplan, which provides an overview of different tasks and what we would be as a team achieving for the course of one year. The intension of this timeline is to provide a rough idea over the tasks and their times, the actual plan shall be decided and defined as we proceed further during the project. For better understanding, we have divide the project into seven main tasks as shown in the below table.

The following list further defines the tasks:

1. **Project organization:** The goal of this task to establish communication between team members so that they can understand each other, in terms of their skills set and expertise. Also decide upon tools for project management, task management, version control and a platform for all further communications.

2. **Preparation and Presentation of mini seminar:** The goal of this task is to get an overview of various technologies and subjects that is required for the project. Select one of the subjects, research the subject in depth and present the importance of the subject towards project to rest of the team.

3. **Developing project plan:** The goal of this task is to combine all the subjects presented by each team member and make a basic sketch of what to achieve in the project group, how to achieve it and by when to achieve it. For our project group, the document is a valuable mean to get an overview over the overall problem, related technologies and required subtasks.

Sr.No.	Tasks	Start	End
1	Project organization	11.10.2018	10.11.2018
2	Preparation and Presentation of mini seminar	11.10.2018	19.11.2018
3	Developing Project Plan	20.11.2018	10.01.2019
4	Reviewing Technologies	20.11.2018	10.01.2019
5	Designing Architecture	06.12.2018	10.03.2019
6	Implementation and Deployment	07.02.2019	15.08.2019
7	Presentation	01.08.2019	20.09.2019

Table 6.1: List of all tasks in the timeplan

4. **Reviewing technologies:** The goal of this task is to list all the technologies relevant for the project and to review pros and cons of each the technology and to decide upon technologies to be used in the project.

5. **Designing architecture:** The goal of this task is to describe in detail and in precise manner on how we are going to achieve the defined tasks and produce end product of the project. This phase is one of the most crucial phase in project group, as it is a core foundation to the project.

6. **Implementation and Deployment:** The goal of this task is to divide the project group into various sub-groups and each sub-group working independently to produce a stable product by the end of the implementation phase. This can the most effort and time, as we might to develop various versions/prototypes until we obtain an end product that is satisfying and stable.

7. **Presentation:** After implementing the system, we will present it to our supervisors and professor. For that, we are going to simulate some use cases and exemplary traffic on a running instance of our software.

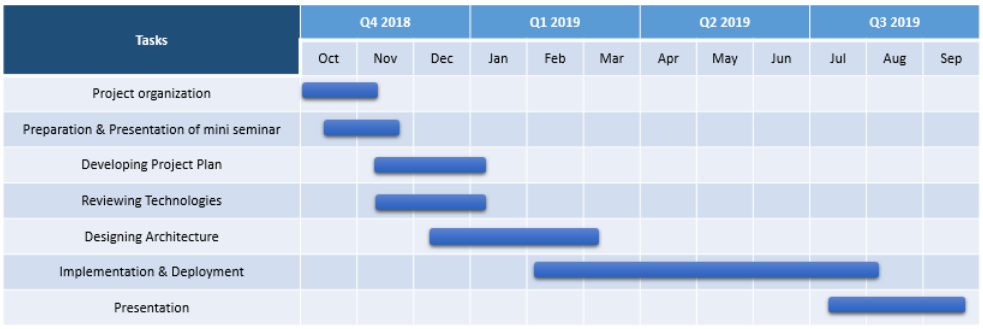

Figure 6.1: A gantt diagram visualising the timeplan.

The sequence of this tasks is visualized in the above diagram. Though the timeline mentioned is fixed and can changes during the course of various phases of project but this is to constantly remind us of the deadlines and to foresee further responsibilities and milestones to be achieved.

7 Conclusion

As we discussed previously, there are many challenges to overcome and milestones to be achieved. At the end our goal is to produce an end product, i.e. OSM adapter, NSD Splitter, NSD translator but we need an in depth knowledge on how we are going to produce it. The next steps for us is to use this document as a base and start reviewing all the technologies and decide upon which ones we have use. Start designing and documenting the architecture which will be the foundation for our implementation. Also in order to be efficient, we will be dividing the project group into sub-groups and independently work on various topics and each week discuss the progress of each sub-group and share knowledge and findings to rest of the team. Collaborate results of sub-groups to produce a stable end product.

Bibliography