# Project Report

Liu Xu

2016/6/6

# 1 Group

Doctor Tian' s NFV Group

# 2 Environment

Ubuntu 14.04.1 LTS
GNU/Linux 3.19.0-28-generic x86_64

# 3 Introduction

## 3.1 NFV Introduction

Network functions virtualization (NFV) architecture consolidates network functions over standard IT hardware by leveraging virtualization technologies, enables quick and cost-effective response of mobile network operators to the new market demands, where the crux is to effectively and efficiently allocate resources to virtual network functions (VNFs). However, a systematical approach for resource allocation in virtualized mobile core network is still unavailable. That is what we propose.

We propose a synthetic approach based on analysis of both network processing procedures and users behaviors. Inspired by the static user behavior modeling, we construct a more practical user behavior model by analyzing over 20TB real data from an operator. With the model, we propose a matrix mapping based dynamic resource allocation mechanism for the virtualized core.

The NFV architecture is shown as figure1. The left is the whole NFV architecture. And in the right is the case of mobile core network, hardware resources in the NFV infrastructure are virtualized to support virtual network functions in mobile core network.

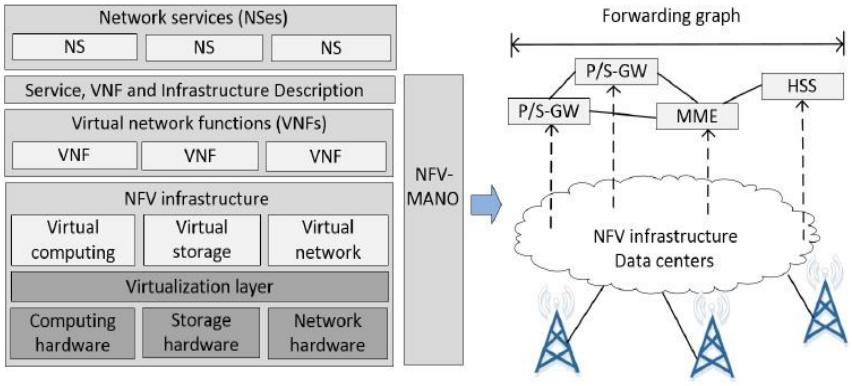


Figure1

The interactions between mobile users and the VMs in the virtual core is more implicit compared with that in the multi-tenant cloud: which VNF to be used depends on the procedure the user is currently in. The following two pictures show the characters. If the mobile user is in the attach procedure, the VNF to be involved are as shown in figure2. And during a calling session, the user could transit to another procedure after attached to the network, where figure3 shows such an example.

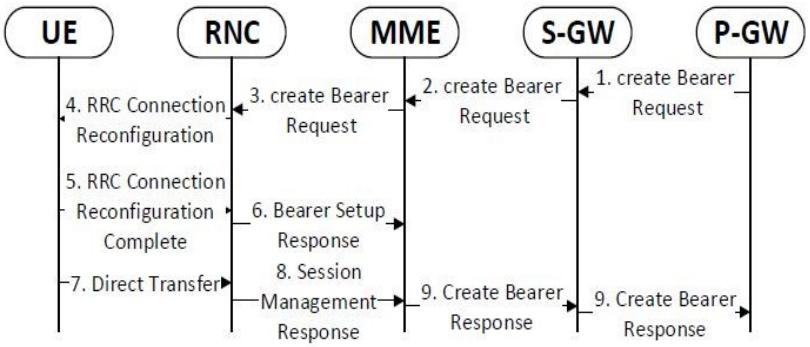


Figure2

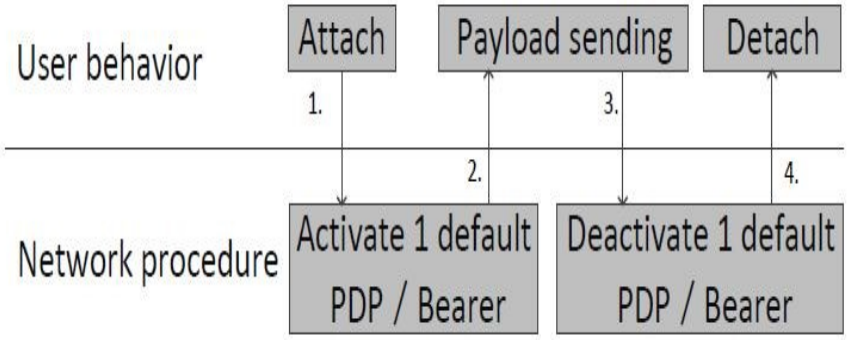


Figure3

## 3.2 Testbed Introdution

To demonstrate the effectiveness of our approach, we develop a testbed that performs resource allocation over a simplified UMTS core network. We conduct experiments on the testbed using application and signaling records of millions of real users.

We conduct the experiment on a cluster of 7 Dell Poweredge R420 servers. The cluster adopts the star topology with one master and 6 slaves as shown in figure4. All servers have 4 CPU and 32GB memory. We further run two kernel based virtual machines on each slave. With a total of 12 virtual machines, we perform all the functions a network service provided.

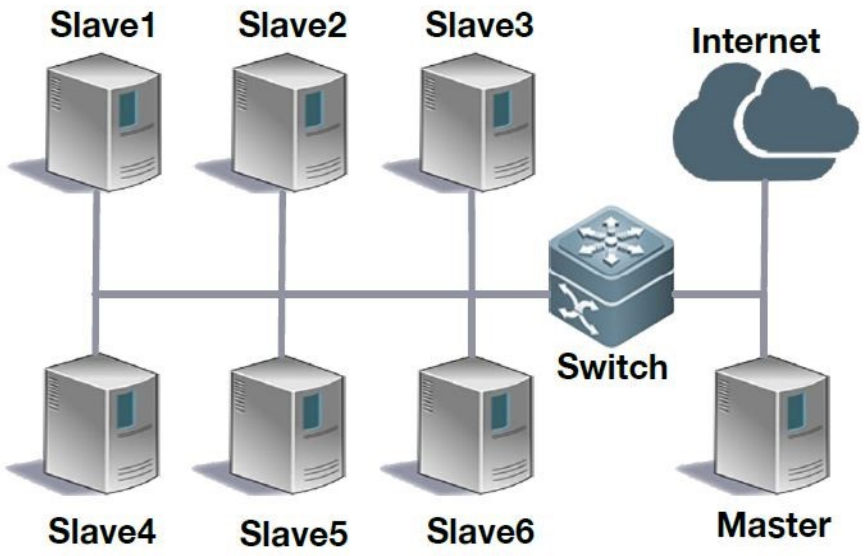


Figure4

# 4 Content

The Testbed Task is assigned to me. The target is by different means to optimize the testbed to make it more effective, more robust and further simulate the real network environment. Now, I will tell you what I do during the process.

## 4.1 Learning

I am mainly responsible for the migration section: to realize the migration and apply to the testbed process to make it better. So the first step is to learn KVM virtual machines and find methods to realize the migration.

What we need for is the live migration, that is, to ensure the normal operation of the virtual machine service, and, at the same time, migrate a virtual machine system from a physical host to another host. Its different from static migration, because the services are available during the migration process, and has only a short time.

In the actual process, I find two methods, one is based on storage sharing, and another based on data block, and after realized both of them, I choose the method based on data block, because its more convenient.

### 4.2 Preparation

After found method, I start my work in the cluster. Firstly, I standardize the virtual machines naming mode, because every virtual machine is individually working before, so the names are loose. But the migration process need the connection between hosts, so I standardize the naming mode to make it easy to manage.

And then, although the method based on data block neednt storage sharing, but it has a demand: the virtual machines environment on the both hosts must be the same, especially the image file, so I have to create same image file for every VM in every host. Although the whole process is no technic, but it is necessary and time-consuming, I spent a lot on it.

### 4.3 First optimization

I adapt the last layer data, find that the resource allocation in every host is different, some hosts have heavy load, but some have just a little. So I think, if we timely migrate the all virtual machines between hosts, such as migrate vm1 from slave1 to slave2, vm2 from slave2 to salve3cycle in turn.

This way can significantly reduce the use in some hosts, in one period, every host has similar load, so as to increase the robustness. And because is in live, the work executed in the virtual machine cannot be affected.

### 4.4 Second optimization

At last, I attend to add migration process to the resource allocation algorithm, aimed to make correlative functions in several virtual machines on a host. Why I attend to do this? because when we implemented the cycle migration, we found a latent problem. The virtual machine is individually working at the first, so we cannot sure the integrity of correlative functions.

But after adding migration process to the resource allocation algorithm, we can make correlative functions in several virtual machines on a host, to guarantee the integrity of functions.

## 5 Conclusion

We compare our model with the blank model under the condition that both models will process the same incoming workload. Specifically, the incoming workload is time-varying and will change according to the number of people accessing the Internet. The blank model will divide the workload equally to 12 sub-jobs and allocate one of them on each VM. In contrast, our dynamic allocation model will assign more jobs to the VM with more resources.

The results show that the new approach significantly increases the utility and capacity of the mobile core network, and adding the cycle migration, the management of the hosts and the virtual machines will be more convenient. The robustness also has an improvement.

And the final optimization hasnt been test, so what I will do next is to test it as soon as possible, I think it will have a good effect. And then, we can find another way to optimize, such as to allocate with memory rather than the numbers of CPU.

# 6 Reflection

Due to this course, I am lucky enough to join in Doctor Tians NFV Group. During the whole term, I have learned a lot, such as servers use, KVM virtual machine management. When I learning, the seniors give me much help, I am very grateful to them. And then, I also try my best to attribute to this group, to make the work better. Finally, I finish my work with my classmate, this process also improves our teamwork, its useful for finishing anything.