Virtual Network Urbanization

Othmen Braham, Ahmed Amamou, Guy Pujolle

Virtual Network Urbanization

Othmen Braham, Ahmed Amamou, and Guy Pujolle

Pierre and Marie Curie University,
Place Jussieu. 4, 75005 Paris, France
{othmen.braham,ahmed.amamou}@virtuor.fr
guy.pujolle@lip6.fr
http://www.lip6.fr/phare

Abstract. Adapting virtualization concepts to satisfy network telecommunication challenges receives more and more attention. The virtual network environment is formed by the amount of bounded virtual resources provided by physical network equipments. Deploying virtual network infrastructure has recently caught more research interests due to its flexibility and manageability. However, virtual network deployment is not evident as it should be. Therefore, it is now necessary to provide virtual environment design to coexist virtual networks and their respective operators. In this article, we describe our virtual network environment architecture. We explain the proposed organization strategies to improve virtual network urbanization. The initial results and quantitative analysis of the architecture deployment are exposed.

Keywords: Virtualization, Network, Management, Urbanisation

1 Introduction

It is widely accepted that virtualization is taken more and more importance for IT platform. The virtualization is a way for sharing physical resources into separate and isolated virtual resources. It permits simultaneous multiple machines to run within one physical computer. The virtual network is based on physical network equipment virtualization. The connected virtual machines composing the virtual network share currently the amount of provided virtual resources.

Advances in virtualization technologies have created new opportunity for network operators to take advantage of network resources more efficiently. There is growing need for an architecture that allows network managers to instantiate virtual networks in an organized and easy-way. Related works, which describe virtual network [1], are more interested on network communication mechanism behaviour due to virtualization overhead. This research explores the design and strategy to be considered to fulfil virtual network operator needs. And in this context is where we want to implement the proposed architecture and test the performance and availability of virtual network.

In our approach, each virtual network could propose a different network service with isolation guarantees. The virtual machine could be any kind of network

equipment used within real network: Routers, Label Switch router, firewall, access point, SIP router, IP PBX... Virtual machine could implement any protocol stack such as: IPv4, IPv6, MPLS... A virtual network is created by the instantiation of each virtual machine that compose its topology. These virtual machines are linked through virtual link. A virtual network router could use any routing protocol such as: OSPFv2, OSPFv3, Rip, RIPng, BGP...

This paper describes an architecture that facilitates the management of high-evolutive virtual network environment. Although many virtual management applications have been developed to offer their special features and tools, few works has been reported on providing support for developing virtual network urbanization application. It permits the instantiations of virtual networks within a virtualized environment and the ability to configure each part.

Aiming to provide an environment for building virtual networks, we have developed our architecture which combines the following features:

- Virtual network isolation;
- Defining a gap between physical and virtual resources management;
- The virtual machines that compose virtual networks could be any network equipment;
- Easy configurable and easy manageable.

To present an overview of our virtual network management application, this paper focuses on our architecture and its key elements. The rest of this paper is structured as follows. Section 2 describes the virtual network environment platform. Section 3, describes the overall virtual network application architecture. Section 4 presents the design and mechanisms underlying the virtual network platform. Section 5 presents the experimental evaluations. Finally, Section 6 concludes the paper with the description of our future work.

2 Virtual network environment platform

The physical nodes are connected through physical network link. Each physical node offers its virtualized resources via a hypervisor. The hypervisor guarantees concurrent virtual machine running isolation conforming to their allocated resources. The Physical Node is the component upon which all virtual machines reside. It corresponds to the physical machine with a real location and hardware specification. Its resources are characterized in term of CPU, memory and network I/O as we are focusing on network equipments. We have used in our platform physical equipment with 4Go RAM, C2D-2.4 Ghz CPU and six 1 Gbyte network interface.

We have adopted XEN [2] as hypervisor because it is open source and increasingly popular among virtual network infrastructure researches. Xen is also a high performance resource-managed virtual machine monitor (VMM) comparing to other technologies as mentioned in literature [3, 4]. The VMM provides isolation and safety for running virtual machines. Despite the advances in virtualization technology and technique, the overhead of network I/O through Xen

virtualization still have a negative impact on the performance of virtual network applications. Though, the performances are close to those of non virtualized software routers if the need for processing overhead is satisfied and allocated resources are oversized to support unfairness in resources share.

The virtual machines share physical host resources. An instantiation of a virtual machine needs: kernel, files system, network application and configuration description of resources that will be allocated. We have used small size virtual machines that can support different network application. We have integrated some available open source project inside virtual machines such as: Xorp [5], Quagga [6], Asterisk [7], MPLS-Linux [8], Opensip [9]. We have elaborated a virtual machine list that covers different kind of network equipment. We have adapted virtual machine operating system to support several type of protocol stack. We have deleted unused operating system modules to eliminate their effect on network traffic. Using light weight virtual machine increases routing performance and managing action such as instantiation, migration and backup.

The virtual network environment as shown in Fig. 1 consists of several virtual networks. Each virtual network can guarantee different kind of service to satisfy the requested Service Level Agreement (SLA). The virtual network creation is achieved by the instantiation of each virtual machine that compose it. The virtual machines must correspond to the offered service in term of protocol stack and network application.

As exposed in Fig. 1, we have different protocol stack IPv4, IPv6 and MPLS network running on the same physical nodes and isolated down to hypervisor isolation. The IPv4 network is extended by IP PBX functionalities to support video conference service. MPLS network is used for video streaming service due to its minimal delay grantee. The IPv6 network guarantees data exchange service between its users. The clients must have the same protocol stack as their virtual network and configure them self to be connected to it. They can be connected to several virtual networks at the same time to take advantages of the different proposed services.

This platform could be used by several operators. They can share the deployed physical infrastructure by allocating available virtual resources. The amount of allocated resources could be used to create different instantiation of virtual networks. Depending on resources needed by each machine that composes the network, virtual machines have to be placed appropriately to an available resources location that satisfies the virtual network topology.

The network is distributed by nature. Due to virtualization level capability, it becomes also dynamic. A new operator can allocate available virtual resources and instantiate a new virtual network with a new kind of service. Another operator could delete a virtual network that is not any more in use to free the allocated resources. The virtual machine migration could be used for different purpose in this kind of platform to ensure high availability and load-balancing. In the next section, we describe our architecture to manage virtual resources and establish heterogeneous virtual networks that guarantee different qualities of service.

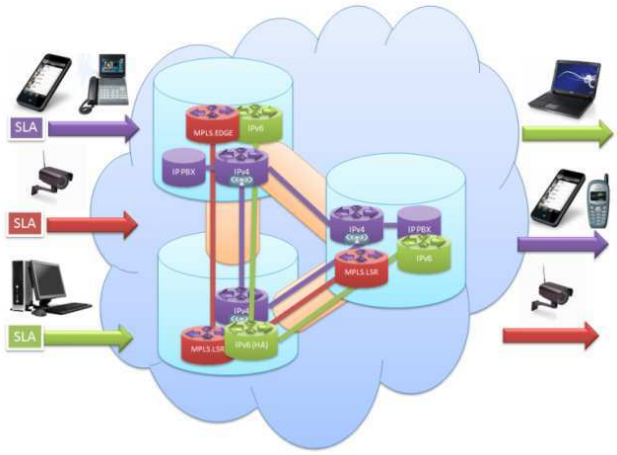

Fig. 1. Virtual network environment platform.

3 Application Architecture Overview

The architecture, as shown in Fig. 2, consists of three principal abstraction layers. The lowest layer is the Infrastructure Provider Domain (IPD). It is the management domain of the infrastructure provider. The IPD includes all physical network resources. Each Physical Node has its own unique identity within the Infrastructure Provider Domain.

The second layer is the Network Provider Domain (NPD). The amount of virtual resources in terms of processor, memory and network interfaces represent the NPD. The IPD administrator defines the namespace boundary for these virtual resources as a NPD. The Network Provider Domain is an aggregation of all the different types of virtual objects. The NPD administrator could be represented as a virtual operator.

Each physical node offers an amount of virtual resources in term of CPU, Memory and network I/O. This amount is shared between virtual operators. To separate between concurrent NPDs in the same physical node, we propose in our architecture a logical border assimilated to a container of virtual resources. The container corresponds to a logical resources amount in term of CPU, memory and network I/O. It defines the limit of allocated resources for a virtual operator. A virtual operator can allocate on each physical node the estimated container capacity depending on its client proportion and location. The request of resources by the NPD could not exceed container capacity. The container will enclose as many virtual machines as its capacity allows that. It guarantees the availability of resources amount allocated by operators. The service level agreement of virtual operators must be satisfied in any case unlike the operator subscribers SLA which are more flexible.

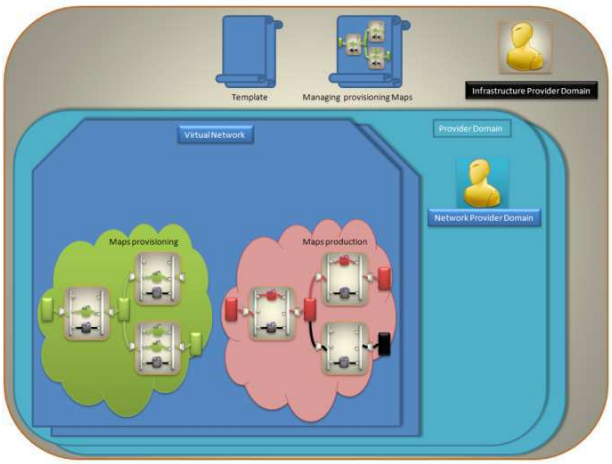

Fig. 2. Application Architecture.

The third layer is composed by virtual networks (VN). The Virtual Network provides the ability to monitor and manage logical groups of resources within the Network Provider Domain. A Virtual Network could be represented by physical location, a subnet range, or a logical grouping of virtual machines. The NPD Administrator creates Virtual Networks for its clients to satisfy their service level agreement request. Each VN has two maps view to bring management simplicity for virtual network resources. The map production shows the virtual resources allocated and already in use within virtual network. It provides a representation of the network topologies. It corresponds to unavailable amount of resources within the map of containers. The Map Production shows the actual running virtual machine and current used virtual resources inside each virtual network in the Network Provider Domain. The monitored objects includes virtual machines, network interfaces, and templates. The NPD Administrator can manage running virtual machine through the map production. The Map Provisioning represents allocated virtual resources not yet in use. It corresponds to the available amount of resource in the map container. The Map Provisioning provides views and capabilities to understand the relationships between the available amount of virtual resources and their physical location within the infrastructure provider domain. It indicates the available virtual resources that could be used by VN to be extended. The NPD Administrator has a view on the available virtualized resources that it could request from IPD through map provisioning.

We have defined a template for each virtual machine that describes its configuration. A template is a picture of a virtual machine that can be used as a master copy to create and provision new virtual machines. This image typically includes a specified operating system and configuration that provides virtual counterparts to real hardware equipments. The template is a combination of three level abstractions: physical template, operating system template and vir-

tual domain template. Physical template describes hardware resources that will be allocated to the virtual machine. This description includes hardware specification that will be seen by the virtual machine such as: CPU, memory, virtual network interfaces. Operating system template includes the operating system specification and system files that will be used on the boot and during the execution of the virtual machine. Virtual domain template indicates the type and functionality that would offer the virtual machine. The NPD Administrator requests an instance of virtual machines according to the defined templates.

The roles are separated into two groups that correspond to domain action. The infrastructure provider domain administrator manages the physical resource level and allocates them to network provider domain. The IPD could accept several NPD depending on its resources offers. The IPD Administrator has the ability to manage and organize Physical Nodes. He can also create new Network Provider Domain and manages role memberships throughout the entire Infrastructure Provider Domain for NPD Administrators.

Unlike an Infrastructure Provider Domain, which is used to organize physical object type, a Network Provider Domain is an aggregation of all virtual resources level which is composed by virtual machines. The network provider domain administrator interacts with its IPD to allocate available virtual resources on each physical node. The NPD clients ask for specific network in term of protocol routing, protocol stack, and network equipment type. VN are then created and configured by their NPD. The NPD Administrator can manage running virtual machine through the map production. The NPD aims to optimize the use of its containers on each physical node to satisfy maximum virtual heterogeneous network. An NPD Administrator has read-only access to the available resources within Infrastructure Provider Domain level shown by provisioning map and no access to other Network Provider Domains. Access or visibility to Maps provisioning depends on the NPD Administrators administrative privileges. He cant manage any physical node because he has permissions on virtual resources only. The NPD Administrator has full management on the owned resources shown by the maps production.

4 Deployment Architecture Platform

We have implemented a management tool as described in the proposed architecture to control virtual resources and establish heterogeneous virtual networks that guarantee different qualities of service. After authentication step, the tool offers multiple functionalities that help to manage physical network resources for IPD administrators and virtual network resources for NPD administrators. It gives them an amalgam of options to control different instantiations of virtual networks and the ability to configure every virtual object within the virtualized environment.

As shown in Fig. 3, we opted for a central decision server with distributed agents on each physical node. The management server is based on web server application and uses a data base for virtual network environment information

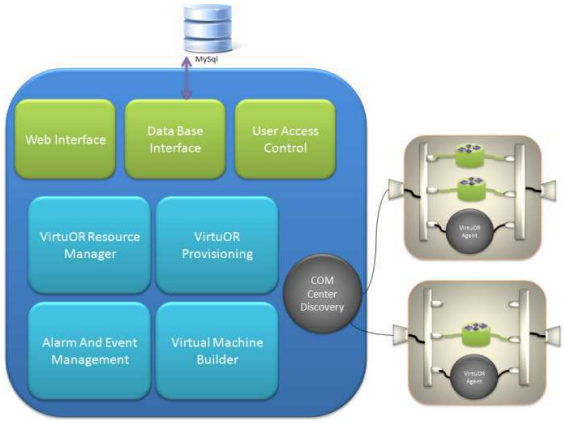

Fig. 3. Platform architecture deployment.

persistence. Each actor must be identified by the server before it could act on its resources and interact with agents. Each agent runs continuously in the background, performing monitoring and executing managing activities even when no actor is logged on the physical node where it resides. It is installed on the privilege domain to permit the execution of privilege instructions on virtual resources and to have a local view of them. The agent has access to the entire physical node resources and can manage from outside the running virtual machine.

All the running virtual machines on managed physical nodes are automatically discovered and shown into correspondent Map Production within their Network Provider Domain. The migrating process describes the action of moving a virtual machine from one physical node to another. The virtual machine migration is a life transferring of a virtual machine snapshot image to another physical node without service interruption. In our case, we can easily migrate from one container to another if resources of destination container permit that.

To manage virtual network, we have created a management network. The management network is formed by the different agents that control each physical machine. In response to NPD needs, IPD administrator creates a container which is a logical resources border on the selected physical node. The NPD administrator instantiates virtual machines of its virtual network through map provisioning. When virtual network instantiation is completed, the NPD can access and configure each virtual machine. He could reach virtual machines using the network that he has created or through the management network.

A virtual operator would not share routing information tables inside virtual machines with any other actor. This information could give indications about its clients activities and location. To guarantee virtual network isolation, our deployment allow only for NPD administrator to configure the inside of its virtual machine instantiations such as routing table, network application and other

configuration. The NPD must be recognized by the identification system of the virtual machine to configure it. The IPD administrator can manage the instantiation of virtual machine but cannot access inside or interfere with the execution behaviour of the running network application.

To guarantee feather extensions to our tool, we have used REST [10] as web service architecture because there is no need to keep session. We have used XML data exchange format between server, agents and actors. Each virtual network environment entity has its own Uniform Resource Identifiers (URI) [11] that permit to execute on it all REST standard action like: GET, POST, DELETE, PUT and the other related resource type actions.

5 Experimental setup

Our experimental testbed consists of 5 physical machines. The witness virtual network connects two client laptops. Each one has C2D 1.6 GHz CPU, 2Go RAM, 100Mb network interface. The same configuration is deployed for the charged virtual network. The Physical node that will host all virtual Network has 4Go RAM, C2D-2.4 GHz CPU, six 1 GByte network interface and XEN 3.4 installed ont it. All instantiated Virtual Routers have this configuration: one x86 virtual CPU, two 100 MB virtual interfaces, 20 Mb image disk size, 80Mb RAM and Quagga router as network application.

As shown in Fig. 4, the physical node will host 2 virtual networks: the witness virtual network and the charged one. The witness network will support all performance tests. The charged virtual network aims to charge the physical node in term of CPU and memory use. It is composed from several virtual routers that are connected to each other in cascade. The network traffic generated by clients keeps the virtual resources in use. The two networks are not connected to the same physical interfaces to avoid influence between the two virtual networks packets. For our experiments, we try to only charge the virtual resources. Connecting the two networks with the same interfaces would produce false results.

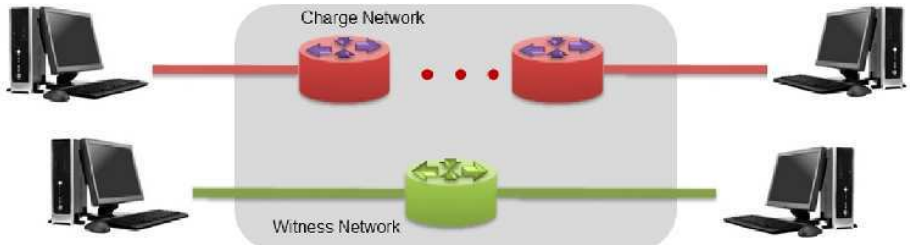

Fig. 4. Testbed description.

The test consists of measuring witness network performance while incrementing routers number in the other one. Test purpose is to prove that a virtual router

can work with acceptable performance when competed by several other routers. On charged network all routers are placed in cascade to be sure that all of them work in the same time. We will use Iperf with default parameters to charge the second virtual network. In these tests will not present the performance of charged Network. We remarked that it has equivalent performance to witness network.

First, we will present throughput measurement result. We will use Mgen to generate series of 100000 TCP packets with Poisson distribution of 1Kb each to evaluate throughput and to more generic. We have chosen this number because on client we have 100Mb network interfaces and to reach the max routing number of packet. Each scenario test was repeated 30 times for standard point and 100 for interesting point (where throughput have a big variation).The duration of each test was fixed to 5 minutes.

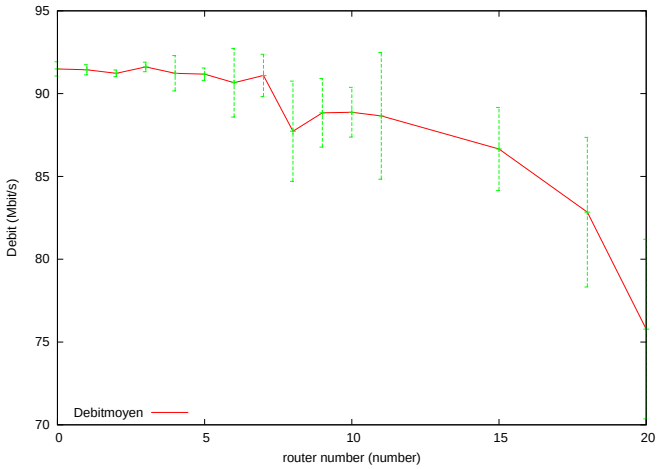

Fig. 5. Throughput measurement results.

We noticed after measurements that until eight machines on charged network we have a relative constant throughput. Starting from the 9th router throughput began to decrease gradually to reach 75 Mb per second with 20 routers as shown in Fig. 5. Throughput's decrease is due to the router scheduling. We estimate that system overload with context switching between routers is the causes of this throughput decrease. Despite the decrease these results demonstrate that even with extreme charge on physical node the throughput remain almost acceptable to satisfy local networking application.

Second, we will present the Round Time Trip (RTT) delay measurement. For delay measurement, we used the Ping RTT result. Ping is generally the most used way to evaluate delay as in [12]. It can be inexact if routers give low priority to ping packets. For our testes as we haven't changed ICMP packet priority inside

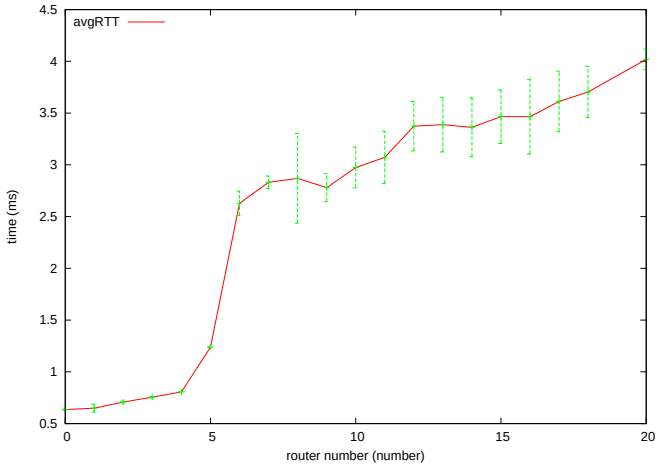

Fig. 6. RTT delay measurement results.

virtual router. Ping packet has the same priority as all other packets. Also for this test, duration is 5 min and each test is repeated 30 times for irrelevant point and 100 times for critical one. As shown in Fig. 6, the RTT in witness network remains under 1ms for a charged network composed from less than 5 routers which is a relatively good value for RTT. For the case of 8 Router we have values under 3ms. Then the RTT begins to increment until reaching 4ms in the worst case with 20 routers. The virtual environment context switching delays the router response and affects RTT variation. However, this delay is acceptable for VoIP communication.

Finally, we will present paquet loss measurement. We have changed the flow nature compared to throughput measurement. Iperf measures the loss rate only for UDP Flow. It was also an occasion to view UDP flow variation in virtual network context. We have also changed the charge network flow to UDP. The Test duration is 5 min and each test is repeated as precedent tests. As we have seen with TCP throughput, we have noticed that with the 8th machine we began to have significant problem with a relatively high loss percentage. As shown in Fig. 7, the loss percentage continues to grow gradually to finally reach 70% in a configuration with 20 routers in the charged network. Over this number of machines the performance is decreasing gradually to reach critical level with 21 routing working in the same time. As we have explained before the decrease in performance is due to the increasing number of virtual machine context switching in the Hypervisor. We have also observed during these tests that the most part of resource is used by the privileged domain dom0 which consumes the most significant part of CPU and memory resources.

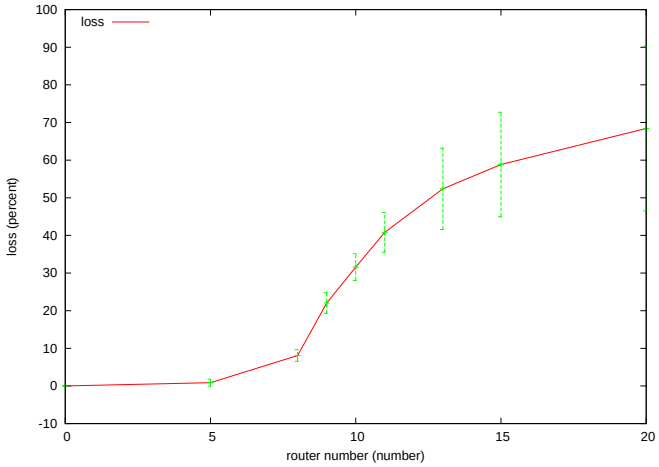

Fig. 7. Paquet loss measurement results.

6 Conclusion

Advances in virtualization technologies have created new opportunities for network operators to take advantage of network resources more efficiently. We try to adapt virtualization concepts to satisfy many of today's network telecommunication challenges.

We have proposed virtual network environment architecture. We have presented the design and mechanisms underlying the virtual network platform. We have started the implementation of this approach with a real prototype. Using experiments with realistic virtual router network applications and benchmarks, we demonstrated the performance/cost benefits and the effortless applicability of our architecture to manage virtual network environment. We have measured the performance of this architecture with a real Testbed. The experimentations show that we have obtained significant result until 10 Virtual routers in the same physical Node.

Our future work will focus on distributed management algorithm of virtual network resources. We will also investigate different means to increment network efficiency by reducing system resources waste in privileged domain, virtual machine size and their resources consumption.

Acknowledgments. We would like to acknowledge VirtuOR start-up to have supported our researches. It helps us to develop our architecture and validate it on a real platform.

References

1. Anhalt, F., Primet, P.: Analysis and evaluation of a XEN based virtual router. HAL-CCSD, (2008)
2. Barham, P.,Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. ACM Symposium on Operating Systems Principles (SOSP), (2003)
3. Deshane, T., Shepherd, Z., Matthews, J.N., Ben-Yehuda, M., Shah, A., Rao, B.: Quantitative Comparison of Xen and KVM. Xen Summit, Boston (2008)
4. Santos, J.R., Janakiraman, G., Turner, Y.: Xen Network I/O Performance Analysis and Opportunities for Improvement. HP Labs, (2007)
5. Xorp, `http://www.xorp.org`
6. Quagga, `ttp://www.quagga.net`
7. Asterisk, `http://www.asterisk.org`
8. MPLS-Linux, `http://sourceforge.net/apps/mediawiki/mpls-linux`
9. Opensips, `http://www.opensips.org`
10. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. Doctoral Thesis, University of Califormia, Irvine (2000)
11. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifiers (URI): Generic Syntax. RFC Editor, (1998)
12. Wang, G., Ng, T.S.E.: The Impact of Virtualization on Network Performance of Amazon EC2 Data Center. IEEE INFOCOM'10, San Diego (2010)