Network Security Virtualization NETSECVISOR

Mrudula Borkar

AISSMS College of Engineering
Department of Computer Engineering

Guided by : Prof. J.S. Chinchole

Outline

Introduction

Social Issue:

To make proper use of resources in order to avoid e-waste.

Introduction

Problem Statement:

To develop a system which can effectively use the available resources for securing a network and other network related to issues in order to avoid idleness of resources.

Introduction

Objective:

- To minimize the need of security devices.
- To maximize the utilization of resources.
- To maintain security flow in the network whenever necessary.

Introduction

Motivation:

Utilization of existing resources
Abstraction of security resources is necessary to provide simple interface
Provide dynamic,flexible and on-demand security services to the users.

Introduction

Need

- Complicated network architectures.
- Complex network management
- Scalability and Cost issues.
- Inefficient use of resources and middle-boxes.
- Need of simple User Interface.

Introduction

New System

- Maximize utilization of resources.
- Redirection of network flow.
- Dynamically enabling security response function.
- Providing flow policies on demand.

Introduction

- Leveraging pre-installed security devices.
- Basic security response functions are enabled such as network isolation.
- Things included:
  - User Interface.
  - Routing Algorithms.
  - Response Strategies.

Literature Survey

Previous Research Approaches

Sr.No	Paper Name	Technique Used	Merits	Demerits
1	"New opportunities for load balancing in network-wide IDS"	NIDS	Traffic Replication, In depth analysis of routing algorithms.	Not dynamic implementation.
2	"Making Middleboxes Someone Else's Problem:Network Processing as a Cloud Service"	APLOMB	Network Redirection.	Relatively complex.
3	"ETSI. Network function virtualization. [Online]. Available: http://portal.etsi.org/NFV/"	Network Function Virtualization	Converts middle-boxes into virtual machine and relocation to centralized place.	Overhead of relocation.

Proposed System

Architecture

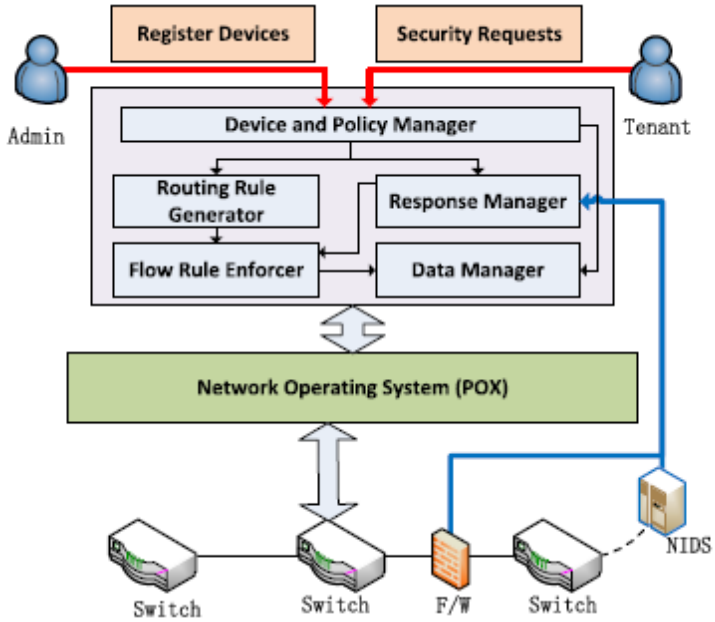

Figure 1: Overall Architecture[1]

Proposed System

Working

- Register security devices.
- Security requests are submitted.
- Parse request and write security policy.
- Routing path and corresponding flow rules are created.
- Flow rules to each security device.
- On detection, corresponding security response function is enabled.

Proposed System

Registeration Of Security Devices

- Simple script language.
- Device ID,Type,Location etc.
- Example:
  Device ID-1
  Device Name-IDS
  Data Path-121
  Device Mode-Passive
  Function- Protect network from DNS attacks.
  Script
  [1,IDS,121,passive,detect DNS attack]

Proposed System

Creating Security Policies

- Tenants define security requests.
- NETSECVISOR describes it with script.
- Request is translated in security policy.
- Functions are mapped.

Proposed System

Routing Path

- Network flow w.r.t. security policy.
- Optimized routing path.
- Two modes
  - Passive
  - Inline
- New algorithms are proposed with the help of SDN.

Proposed System

Enable Security Response Function

- After detection, action to be taken.
- On detected packets or infected host.

Algorithm

Overview

- Different types of network .
- Combination of inline and passive.
- SDN technologies such as OpenFlow.
- Terms:
  - Start node
  - End node
  - Security node
  - Security link

Algorithm

Layout

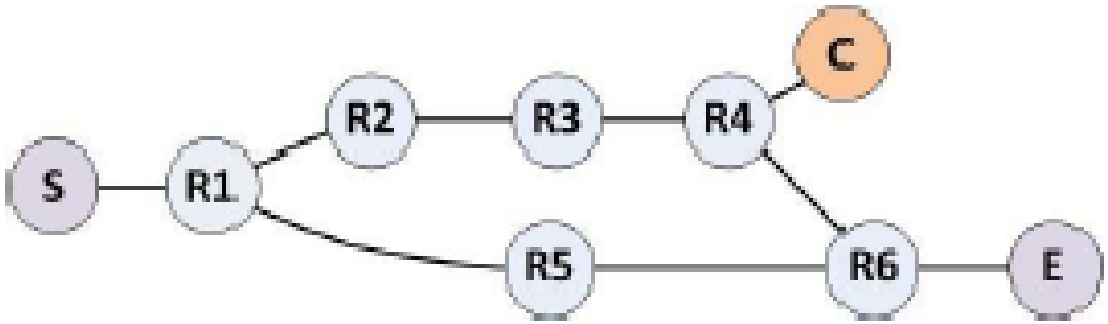

Figure 2: Layout[1]

Algorithm

Shortest Path

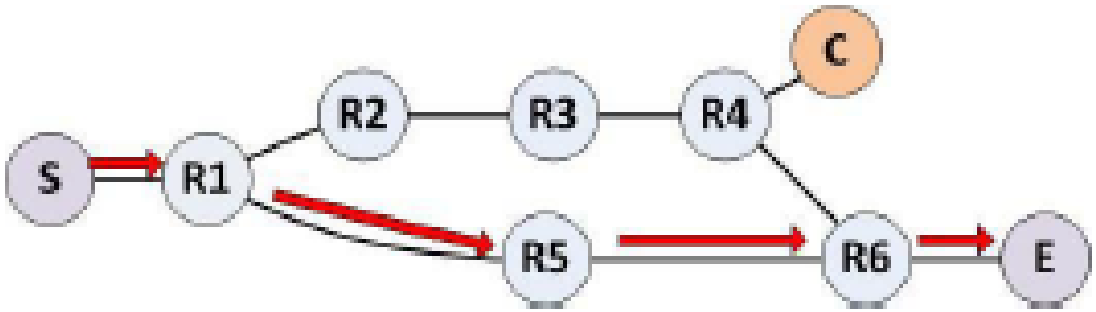

Figure 3: Shortest Path[1]

Algorithm

A1:Multipath-Naive

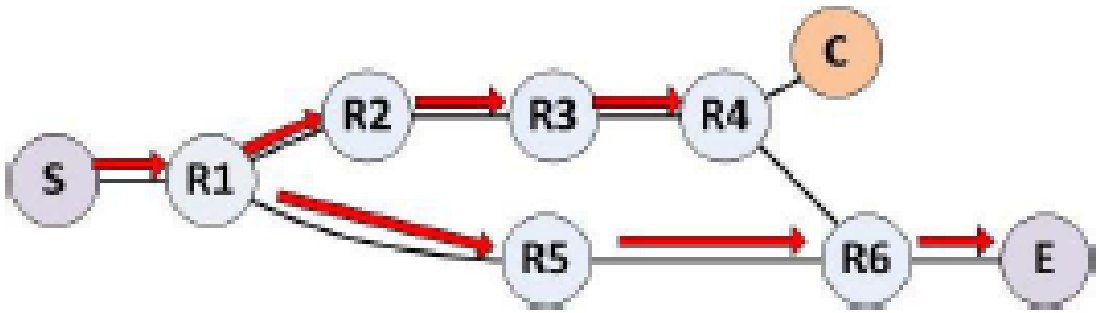

Figure 4: A1:Multipath-Naive[1]

Algorithm

A2:Shortest Through

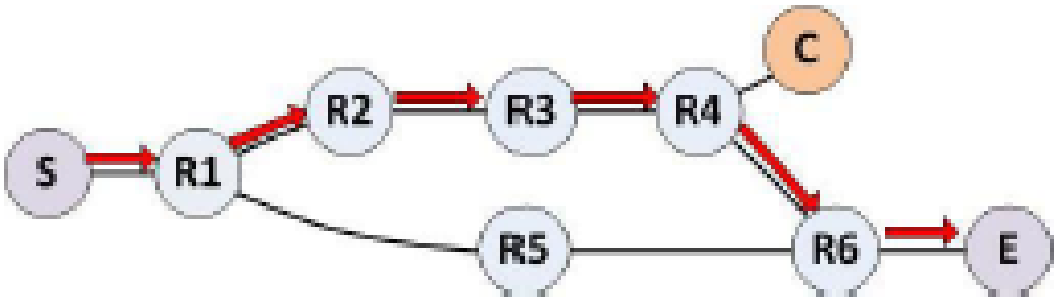

Figure 5: A2:Shortest Through[1]

Algorithm

A3:Multipath Shortest

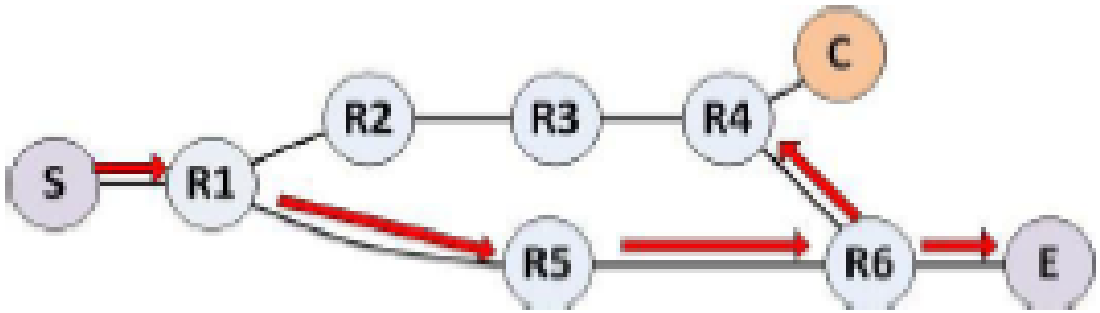

Figure 6: A3:Multipath Shortest[1]

Algorithm

A4:Shortest Inline

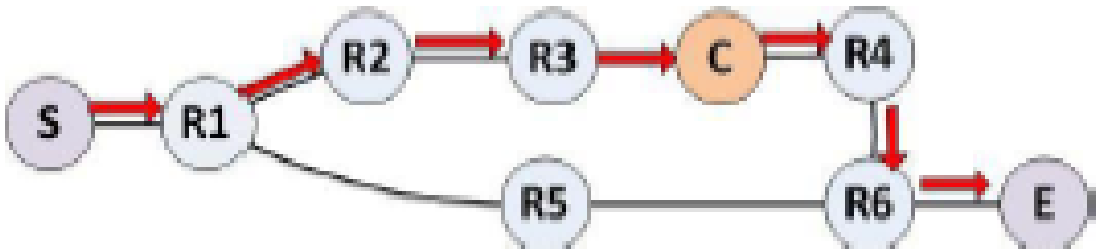

Figure 7: A4:Shortest Inline[1]

Algorithm

Comparison

Algorithm	Pros	Cons	When to Use
A1:Multipath-Naïve	Simple and fast	Redundant flows	Enough network capacity, delay is important
A2:Shortest-Through	No redundant path	Computation overhead, when multiple devices	Not enough network capacity, delay is not so important
A3:Multipath-Shortest	Efficient routing path	Computation overhead, when multiple devices	Not many hops(e.g.,communication between inside Vms)
A4:Shortest-Inline	Guarrantee passing through a specific link	Computation overhead, when multiple devices	For an inline security device(e.g.,IPS)

Figure 8: Comparison between algorithms[1]

Experimental Study

Estimation of performance.

- Generation Time.
- Network Cost.
- CPU and Memory Overhead.
- Response Time.

Compared with Dijkstra's Algorithm.

Experimental Study

Generation Time

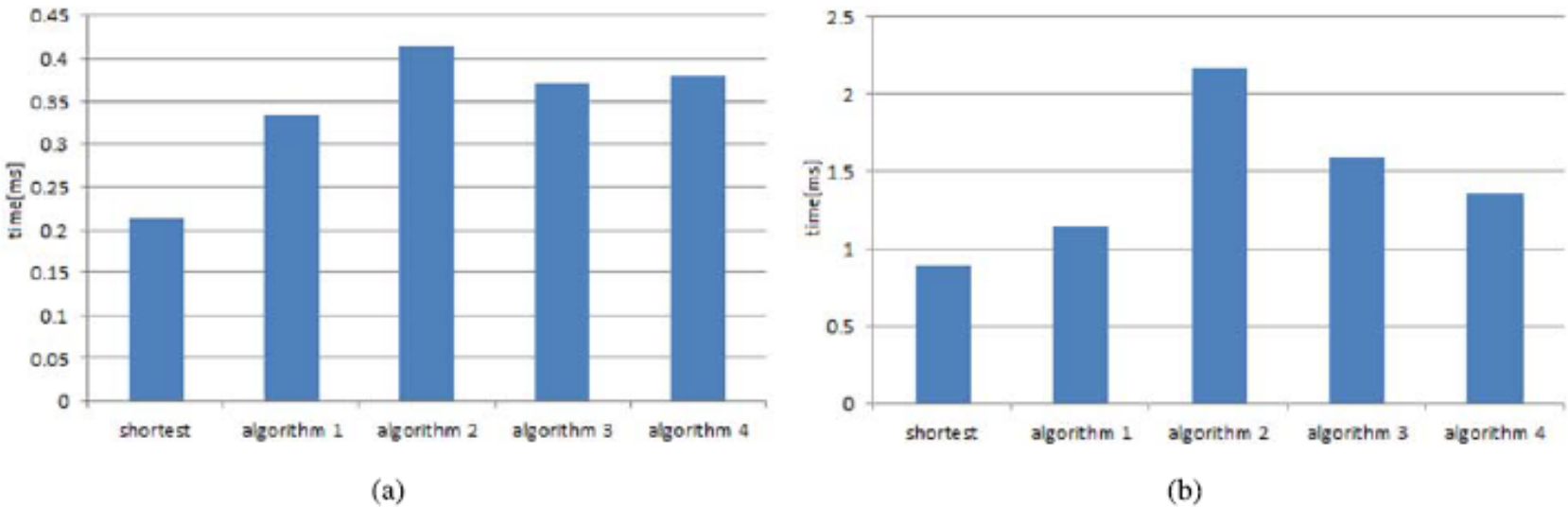

Figure 9: Flow Rule Generation Time Measurement(a)16 routers (b)64 routers [1]

Experimental Study

Network Cost

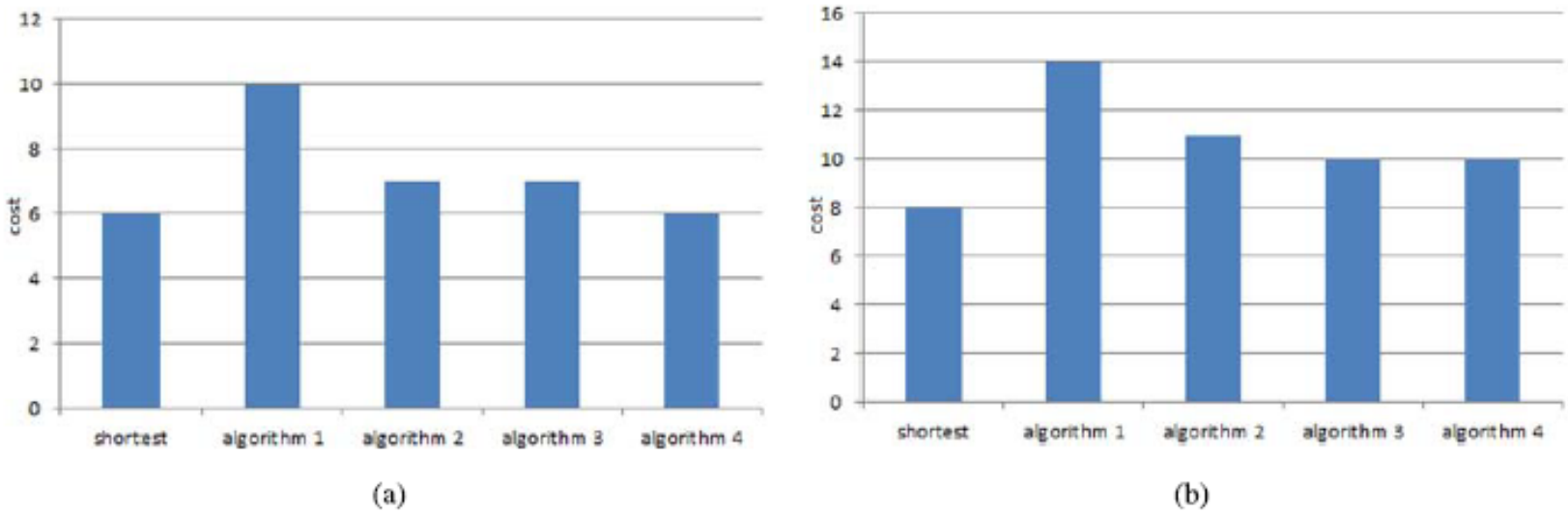

Figure 10: Network cost measurement (a)12 routers (b)64 routers [1]

Experimental Study

CPU and Memory Overhead

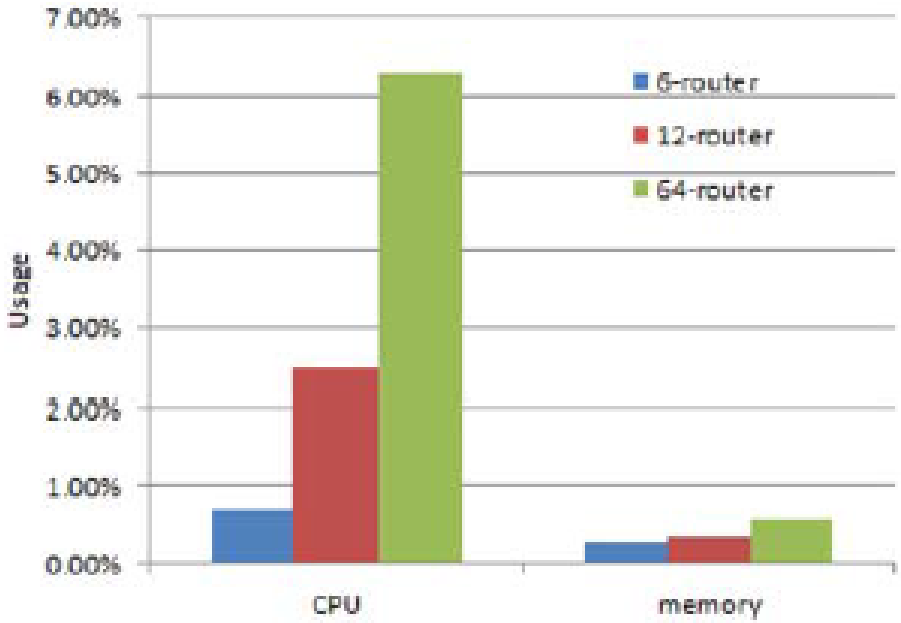

Figure 11: CPU and Memory Overhead[1]

Experimental Study

Response Time

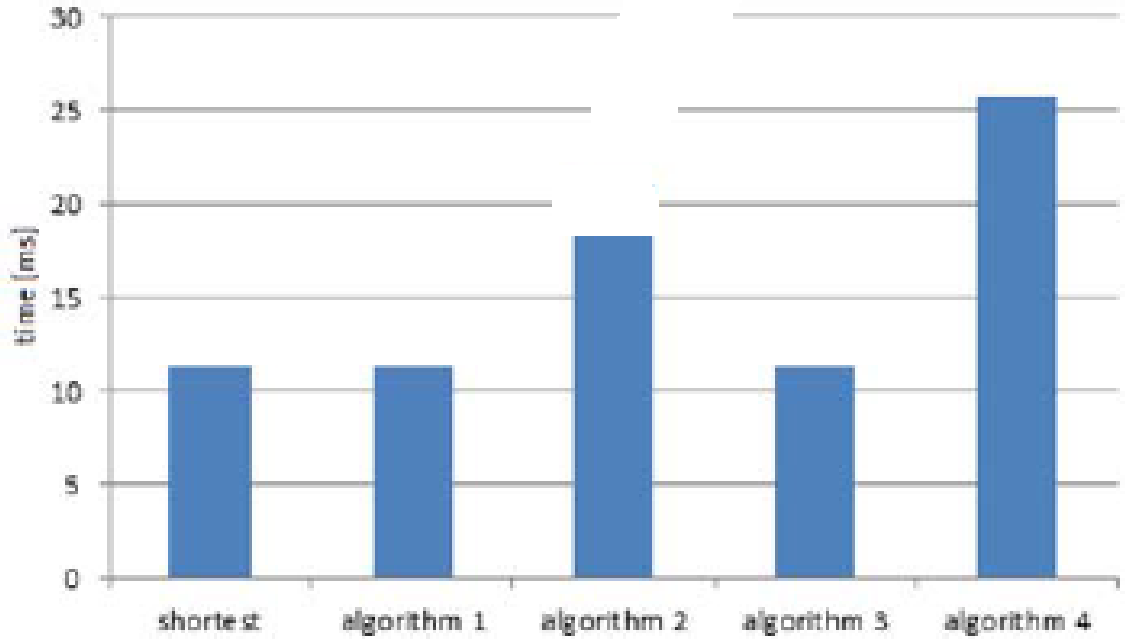

Figure 12: Response Time[1]

Evaluation

Advantages

- Easy, flexible and efficient.
- Proper utilization
- Abstraction of security resources.
- On demand, flexible and dynamic security service.

Evaluation

Disadvantages

- May not generate routing paths in some cases
- May suffer from mistakes of tenants.

Conclusion

This approach builds secure, extensible and dynamic network environment by virtualizing pre-installed resources and providing response function whenever necessary.

Future Scope

- Overcome failure in path generation.
- Identify misconfigured policies.
- Large scale environment.

References

- [1]Seungwon Shin,Haopei Wang and Guofei Gu, "A First Step Toward Network Security Virtualization : From Concept To Prototype", in IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,VOL. 10,NO.10,OCTOBER 2015.
- [2]Philip Porras, Seungwon Shin, Vinod Yegneswaran, Martin Fong, Mabry Tyson, Guofei Gu, "A Security Enforcement Kernel for OpenFlow Networks", in HotSDN'12,August 13,2012,Helsinki,Finland.
- [3]Sayed Kaveh Fayazbakhsh, Vyas Sekar, Minlan Yu, Jerey C. Mogul, "FlowTags:Enforcing Nework-Wide Policies in the presence of Dynamic Middlebox Action", in HotSDN'13,August 16,2013,Hong Kong, China.
- [4]Zafar Ayyub Qazi, Cheng Chun Tu, Luis Chiang, Rui Miao, Vyas Sekar, Minlan Yu, "SIMPLE-fying Middlebox Policy Enforcemenr using SDN", in SIGCOMM'13 August 12-16,2013, Hong Kong, China.
- [5]OpenFlow."Open Networking Foundation. [online]. Available:https://www.opennetworking.org/sdnresources/open flow"

References

- [6]V. Heorhiadi, V. Sekar, and M. K. Reiter, New opportunities for load balancing in network-wide intrusion detection systems, in Proc. ACM CoNEXT,2012, pp. 361372.
- [7]J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V.Sekar, Making middleboxes someone elses problem: Network processing as a cloud service, in Proc. ACM SIGCOMM, 2012, pp. 1324
- [8]N. Foster, M. J. Freedman, R. Harrison, J. Rexford, M. L. Meola, and D.Walker, Frenetic: A high-level language for OpenFlow networks, in Proc. ACM Workshop Program. Routers Extensible Services Tomorrow (PRESTO), 2010, Art. ID 6.
- [9]ETSI. Network Function virtualization. [Online]. Available: http://portal.etsi.org/NFV/, accessed Mar. 10, 2016.

Thank You!

Questions