
Ángel Leonardo Valdivieso
Lorena Isabel Barona
{angevald, lorebaro}@ucm.es

Group of Analysis, Security and Systems
Department of Software Engineering and Artificial Intelligence
Complutense University of Madrid

Evolution and Challenges of Software Defined Networking
Outline

1. Introduction
2. Evolution of Software Defined Networking
3. SDN Research Initiatives
4. SDN Challenges
5. Conclusions
Global Internet IP Traffic by Device Type

Exabytes per Month

Source: Cisco VNI, 2013
The percentages within parenthesis next to the legend denote the relative traffic shares in 2012 and 2017.
Introduction

Global Consumer IP Traffic

Exabytes per Month

23% CAGR 2012-2017

- Web/Data (24.2%, 18.9%)
- File Sharing (15.7%, 8.1%)
- Managed IP Video (21.8%, 21.0%)
- Internet Video (38.3%, 52.0%)

Source: Cisco VNI, 2013
The percentages within parenthesis next to the legend denote the relative traffic shares in 2012 and 2017.
New requirements:

- Easy support to implement custom network services
- Changing traffic patterns dynamically
- Support to public and private cloud services
- Intelligent network management of increasing bandwidth – Big data support

The networks are ready to support this new requirements?
Introduction

Limitations of Current Networking Technologies:

- Protocols solve a specific problem (ACL, VLAN, QoS, Firewall, NIDS, authentication…) → complexity
- Need to configure network devices individually (using CLI)
- Static nature of networks in contrast to the today´s dynamic environment (server virtualization, distributed applications)
- Vendor dependence to support new capabilities and services
2. Evolution of SDN

Active Networks:
- Smart Packets
- SwitchWare
- Calvert...

Separation Control-Data:
- Tempest
- ForCES
- RCP
- Ethane...

Openflow and NOS:
- Openflow Protocol
- NOX, Beacon
- Procera...

(1990 – 2000)
- Open the control network through a programming interface (network API)
- Programming models: Node Operating System (NodeOS)
- Not offer practical performance and security

(2000-2010)
- Open interface between control and data planes
- Logically centralized control of the network
- Distributed state management
- Difficult implementation in production networks (compatible hardware)

(2010→)
- Interface build on already supported technology (flowtables)
- Vision of a Network Operating System (NOS)
- Difficult to enable sophisticated middlebox functionality
Software Defined Networking is an emerging network architecture where network control is decoupled from forwarding and is directly programmable.
Software Defined Networking is an emerging network architecture where network control is decoupled from forwarding and is directly programmable.

Source: Open Networking Foundation (ONF)
SDN Research Initiatives

1. Introduction
2. Evolution of SDN
3. SDN Research Initiatives
4. SDN Challenges
5. Conclusions

Virtualization & NOS

Simulation

Testbeds

Multimedia
Virtualization

Evolution and Challenges of Software Defined Networking
Virtualization: FlowVisor
Evolution and Challenges of Software Defined Networking

### 3. SDN Research Initiatives

**Network Operating System NOS**

1. **Introduction**

2. **Evolution of SDN**

3. **SDN Research Initiatives**

4. **SDN Challenges**

5. **Conclusions**

- **Switch**, Virtual Switch, Router

- **Procera**

- **OpenFlow Protocol**

- **Northbound API**

- **Southbound API**

- **QoS**, Routing, Firewall
Simulation: Mininet - An Instant Virtual Network on your Laptop

Source: B. Lantz, B. Heller, and N. McKeown, “A Network in a Laptop: Rapid Prototyping for Software-Defined Networks”
SDN Research Initiatives

Testbeds: Global Environment for Network Innovation (GENI)
SDN Research Initiatives

Testbeds: OpenFlow in Europe
Linking Infrastructure and Applications (OFELIA)
Multimedia: SDN helps to improve the QoE experience

SDN Path Optimization Systems Architecture: QoS Matching and Optimization Function (QMOF) and Path Assignment Function (PAF)

If the quality of the network is degraded, the system can react and dynamically modify the path parameters of the network prioritizing the users configuration.


Figure 2 – The proposed architecture for QoE-driven service optimization and path assignment

Figure 3 – An example of a session Media Degradation Path
Multimedia: Fast Recovery of a link failure

SDN improves the recovery time from a link failure using an alternative path

The system uses the restoration and protection mechanism.

In restoration system, the controller calculates and establishes an alternative path when a path failure signal is received.

In protection system, the controller anticipates a failure and calculates two disjoint paths (working and protected)

The experiments show a recovery time of less than 50 ms for carrier-grade network.

Modeling and Performance:

How many controllers needed and the localization of these controllers?

- SDN establishes the separation of control and data planes, but doesn’t necessarily order the existence of a single controller.
- Control plane may have a single controller, a set of controllers of even any dynamic topology.
- The number and position of the controllers depends on desired reaction bounds, metric choice and the network topology.
SDN Challenges

Modeling and Performance:

How to select the appropriate NOS and how to measure its performance?

- The performance of the controller depends directly of programming language (C++, Java, Python) and the development environment
- It is necessary to establish a balance between high performance and the productivity of developers (a developer friendly language)
SDN Challenges

NOS: How to measure its performance?

How many datapath request can the controller handle per second? How fast the controller responds to datapath request?

Source: D. Erickson, “The Beacon Openflow Controller,”
Resilience and Recovery:

A failure of the controller can negatively compromise resilience of the whole network.

- The failure can be fired by abruptly aborting of a NOS process, an unexpected error of the application or a DDoS attack.
- Openflow establishes the configuration of one or more backup controllers, but doesn’t provide a coordination mechanism between them.

The project CPRecovery permits the replication between primary and secondary SDN controller in two phases.

The replication phase acts during normal functioning of the controller and send regular updates to the backup controller. The recovery phase acts during a failure state of the primary controller and starts the backup controller as a main controller.

Convergence and Integration:
Integration with legacy networks (actual Internet infrastructure)

- The equipment could be upgraded or even replaced to support Openflow → High costly network re-engineering
- OpenFlow is unable to handle or manage legacy equipment → it is difficult to connect two-Openflow environments with legacy networks.
- It is necessary a coordination between SDN equipment and legacy infrastructure

5. Conclusions

- SDN can be expanded beyond actual match/action paradigm. For example, it could integrate middleboxes or programmable custom packet processors.

- This integration could offer new services like on the fly encryption, transcoding, or traffic classification. This require coordination, consensus and vendor support.

- In control plane, the composing and coupling of heterogeneous components are still difficult. For example, compose application using Beacon, POX or Floodlight simultaneously.

- Finally, remark that SDN is a tool. The research community can use this tool to create new innovative services and applications.
Thanks for your time and attention!

Ángel Leonardo Valdivieso
angevald@ucm.es

Group of Analysis, Security and Systems
Department of Software Engineering and Artificial Intelligence
Complutense University of Madrid