Software-Defined Networking and Advanced Network Control Programming

1 Introduction

Kai Gao

kaigao@scu.edu.cn

School of Cyber Science and Engineering
Sichuan University

Logistics

Course Information

Course Expectations

Grading

Course Information

Software-Defined Networking and
Advanced Network Control Programming
Dr. Kai Gao

Associate Research Scientist
kaigao@scu.edu.cn
Room 1003, School of Cyber Science and Engineering, Jiangan

软件定义网络与高级网络控制编程
高凯

副研究员
四川大学江安校区网络空间安全学院 10 楼 1003

Course Information

Software-Defined Networking and
Advanced Network Control Programming
Dr. Kai Gao

Associate Research Scientist
kaigao@scu.edu.cn
Room 1003, School of Cyber Science and Engineering, Jiangan

软件定义网络与高级网络控制编程
高凯

副研究员
四川大学江安校区网络空间安全学院 10 楼 1003

Discussion

Online Discussion

QQ: 627100173

Offline Discussion

Time: To be determined

Place: Room 1003, SCSE, SCU@Jiangan

https://surl.amap.com/23ixjfjm3PX

情景演绎

2023 年 x 月 x 日

情景演绎

请说一说你的软件定义网络这门课学了些什么？

情景演绎

请说一说你的软件定义网络这门课学了些什么？

我学习了软件定义网络的基本概念

. . .

情景演绎

请说一说你的软件定义网络这门课学了些什么？

我学习了软件定义网络的基本概念 . . . SDN 的核心思想是 . . .

情景演绎

请说一说你的软件定义网络这门课学了些什么？

我学习了软件定义网络的基本概念...SDN 的核心思想是...SDN 的体系结构中包含...

情景演绎

(嗯，可能看过百度百科)

情景演绎

(嗯，可能看过百度百科)

我学习了SDN 数据平面的代表技术 OpenFlow . . .

情景演绎

(嗯，可能看过百度百科)

我学习了SDN数据平面的代表技术
OpenFlow . . . SDN控制器. . .

情景演绎

(让我来看看你是不是真的学过) 你知道 OpenFlow 的转发原理吗？

情景演绎

(让我来看看你是不是真的学过）你知道 OpenFlow 的转发原理吗？

OpenFlow 采用基于优先级控制的匹配行为表，…

情景演绎

(看来理论是真的学过,能上手做项目么？) 你有没有什么 SDN 的实践经历呢？

情景演绎

(看来理论是真的学过，能上手做项目么？) 你有没有什么 SDN 的实践经历呢？

我使用 mininet 进行了 SDN 网络的模拟，

情景演绎

(看来理论是真的学过，能上手做项目么？) 你有没有什么 SDN 的实践经历呢？

我使用 mininet 进行了 SDN 网络的模拟，用 OpenDaylight 控制器对网络进行控制，

情景演绎

(看来理论是真的学过，能上手做项目么？) 你有没有什么 SDN 的实践经历呢？

我使用 mininet 进行了 SDN 网络的模拟，用 OpenDaylight 控制器对网络进行控制，我还学习了如何利用 SDN 实现流量工程，...

情景演绎

(有工程实践经历了，能不能做科研呢？）你了解 SDN 有哪些研究方向呢？

情景演绎

(有工程实践经历了，能不能做科研呢？）你了解 SDN 有哪些研究方向呢？

SDN 的各层次中都有很多研究问题，

情景演绎

(有工程实践经历了，能不能做科研呢？）你了解 SDN 有哪些研究方向呢？

SDN 的各层次中都有很多研究问题，我主要了解的是 SDN 网络控制编程这个方向，

情景演绎

(有工程实践经历了，能不能做科研呢？）你了解 SDN 有哪些研究方向呢？

SDN 的各层次中都有很多研究问题，我主要了解的是 SDN 网络控制编程这个方向，这个方向上我比较熟悉的比较重要的研究包括 . . .

学习内容

- 理解 SDN 基本概念：定义、核心思想和体系结构

学习内容

- 理解 SDN 基本概念：定义、核心思想和体系结构
- 熟悉 SDN 关键技术：OpenFlow、控制器

学习内容

- 理解 SDN 基本概念：定义、核心思想和体系结构
- 熟悉 SDN 关键技术：OpenFlow、控制器
- 掌握 SDN 工程工具：Mininet、OVS、OpenDaylight、YANG

学习内容

- 理解 SDN 基本概念：定义、核心思想和体系结构
- 熟悉 SDN 关键技术：OpenFlow、控制器
- 掌握 SDN 工程工具：Mininet、OVS、OpenDaylight、YANG
- 掌握 SDN 基础理论工具：自动机理论、（整数）线性规划

学习内容

- 理解 SDN 基本概念：定义、核心思想和体系结构
- 熟悉 SDN 关键技术：OpenFlow、控制器
- 掌握 SDN 工程工具：Mininet、OVS、OpenDaylight、YANG
- 掌握 SDN 基础理论工具：自动机理论、（整数）线性规划
- 了解 SDN 网络控制编程代表性研究及其原理

学习内容

- 理解 SDN 基本概念：定义、核心思想和体系结构
- 熟悉 SDN 关键技术：OpenFlow、控制器
- 掌握 SDN 工程工具：Mininet、OVS、OpenDaylight、YANG
- 掌握 SDN 基础理论工具：自动机理论、（整数）线性规划
- 了解 SDN 网络控制编程代表性研究及其原理
- 综合应用：利用 SDN 实现特定场景中的特定网络控制要求

Grading

- Attendance/考勤 (10%)
- Quiz/课堂测试 (20%)
- Lab/实验 (30%)
- Team Project/团队项目 (40%)
- Participation/课程参与（加分）(5%)

Grading

- Attendance/考勤 (10%)
- Quiz/课堂测试 (20%)
- Lab/实验 (30%)
- Team Project/团队项目 (40%)
- Participation/课程参与（加分） (5%)

除了团队项目之外，其它考察内容大部分同学可以在课内完成

Quiz

Answer the questions after the lectures.

- Quiz 1: How does OpenFlow tables work?
  Week 2/September 12
- Quiz 2: How does YANG language define a data tree?
  Week 4/September 26
- Quiz 3: How to compute the product of two automatons?
  Week 7/October 17
- Quiz 4: What is trace tree and how to generate rules from it?
  Week 10/November 7

Lab

Complete 3 labs:

- Lab 1: OpenFlow, OpenVSwitch, and Mininet
  Week 3/September 19
- Lab 2: OpenDaylight Deployment and Development
  Week 6/October 10
- Lab 3: Traffic Engineering with SDN
  Week 9/October 31

Team Project

Form a team of 3-4 students and work on a course project:

- identify a networking scenario and a problem
- set up simulation environments
- generate traffic or replay real traces
- design and implement a prototype using SDN technologies and tools
- write a project report

Deadline: Week 16/December 12

Course Overview

Introduction

The Road to SDN

Time: Week 1 (September 5)

- Why SDN: Revisit the historical development of SDN
- What is SDN: Scope, definition and characteristics of SDN
- What can SDN do: Applications and deployment

SDN Data Plane

Where Packets Go

Time: Week 2 (September 12)

- From OpenFlow to P4: the historical development of SDN data plane (数据平面) (consider it as the data path in the computer architecture)
- Software data plane: Open VSwitch (OVS), DPDK, bmv2
- Hardware data plane: OpenFlow switches, P4 chips, SmartNIC

Lab 1: SDN Emulation with Mininet

Build Your Own SDN Network

Time: Week 3 (September 19)

- Build virtual SDN networks with Mininet
- Control OVS from command line
- Understand Linux name space
- Complete a list of exercises

SDN Control Plane

Where Routing is Done

Time: Week 4 (September 26)

- The historical development of SDN control plane (控制平面)
- Yet Another Next-Generation (YANG)
- The OpenDaylight controller

Lab 2: SDN Development on OpenDaylight

Be Your Own Master

Time: Week 6 (October 10)

- Background: software development environment for SDN (OSGi, Karaf, Maven, RESTful)
- Deployment: Connecting Mininet to OpenDaylight
- Define data model & API with YANG
- Development: Build SDN applications for OpenDaylight
- Complete a list of exercises

Basis for SDN Programming

All that Math Jazz

Time: Week 7-8 (October 17/24)

- Automata theory (自动机理论) (formal definitions, product construction)
- Optimization techniques (linear programming/线性规划, mixed integer linear programming/混合整数线性规划)
- Boolean satisfiability problem (布尔可满足问题) (satisfiability modulo theories)
- Algebra (代数系统) (Kleen Algebra, routing algebra)

Lab 3: Network Design with SDN Technologies

Let's Solve some Real Problems

Time: Week 9 (October 31)

- Background: solving optimization problems with PuLP
- Background: solving SMT problems with z3
- Exercise: Realize SDN traffic engineering (流量工程) with Mininet and OpenDaylight

SDN Programming Languages

Make it Easy

Time: Week 10-11 (November 7/14)

- Cover a wide range of SDN programming languages
- Introduce the key technique behind these languages

Summary

In this lecture, we cover the following topics:

- A brief history of SDN
- SDN: features, architecture, layers
- Applications and deployment of SDN

Summary

In this lecture, we cover the following topics:

- A brief history of SDN
- SDN: features, architecture, layers
- Applications and deployment of SDN

You should

- roughly understand what is SDN

Summary

In this lecture, we cover the following topics:

- A brief history of SDN
- SDN: features, architecture, layers
- Applications and deployment of SDN

You should

- roughly understand what is SDN
- have a basic sense of why it is designed the way it is

Summary

In this lecture, we cover the following topics:

- A brief history of SDN
- SDN: features, architecture, layers
- Applications and deployment of SDN

You should

- roughly understand what is SDN
- have a basic sense of why it is designed the way it is
- feel excited to learn more

An Introduction to SDN

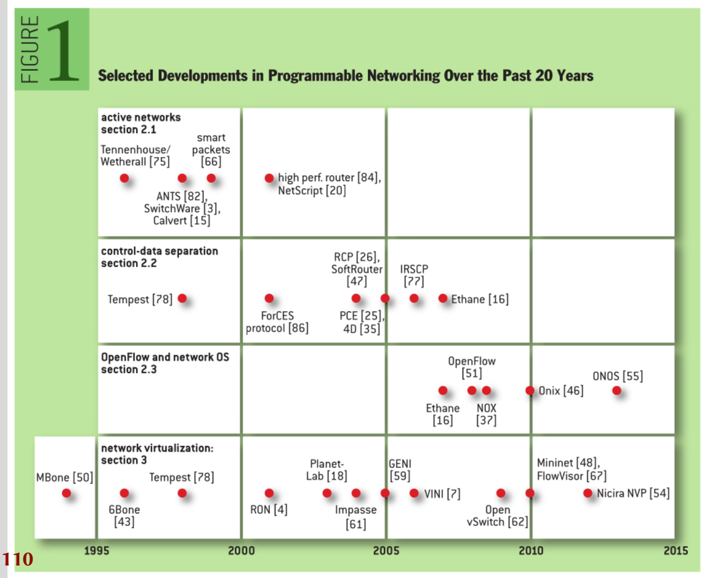
A Brief History

Scope, Definition and Characteristics

Deployment and Applications

A Brief History

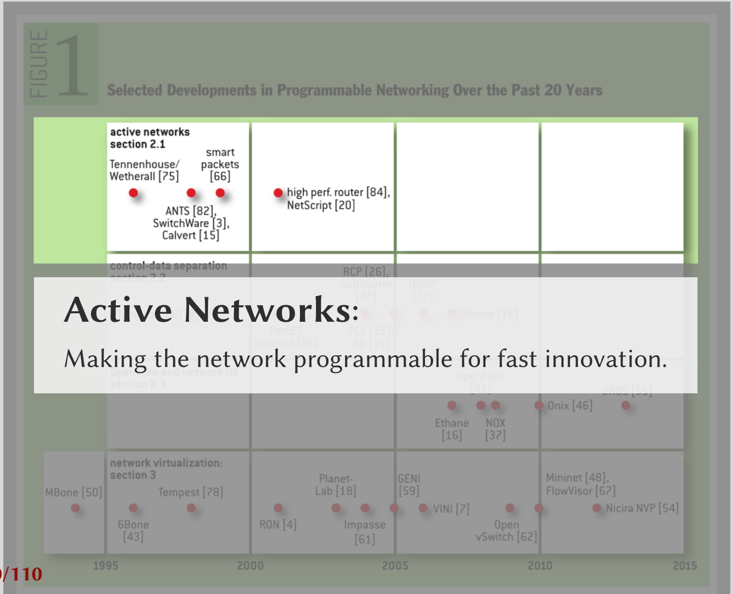
What makes the SDN today



Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

A Brief History

What makes the SDN today



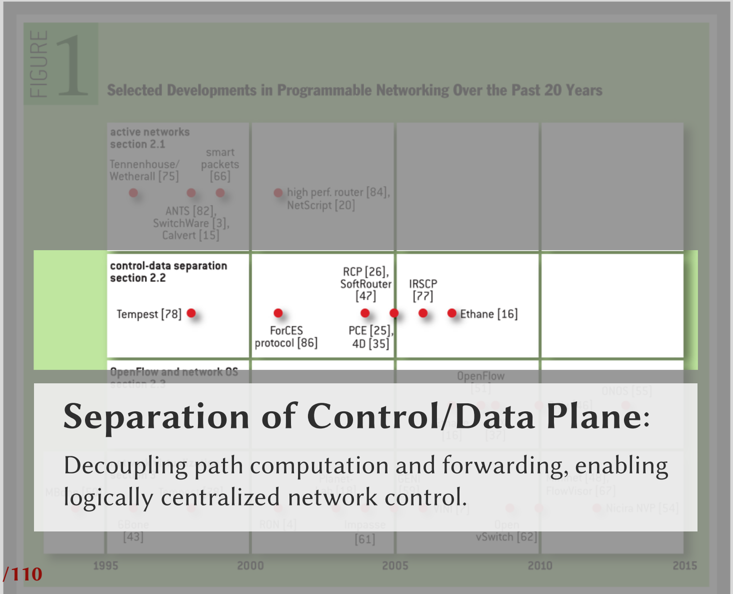
Active Networks:

Making the network programmable for fast innovation.

Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

A Brief History

What makes the SDN today



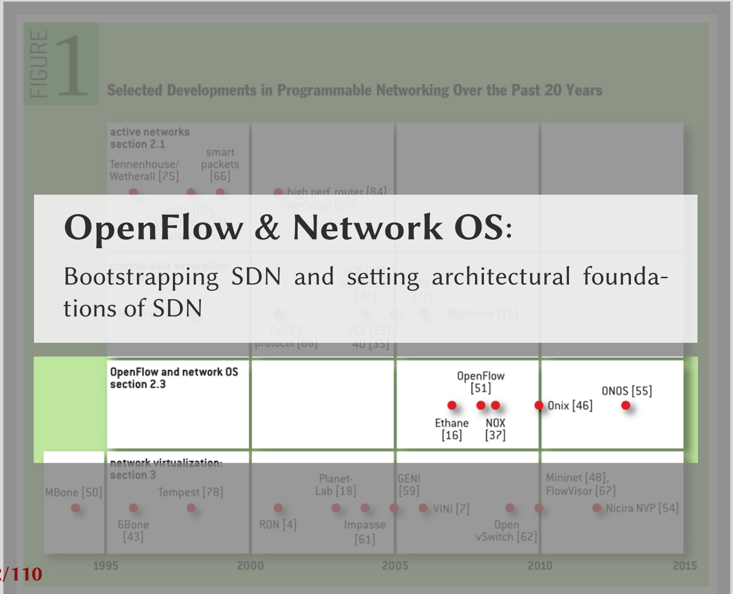
Separation of Control/Data Plane:

Decoupling path computation and forwarding, enabling logically centralized network control.

Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

A Brief History

What makes the SDN today



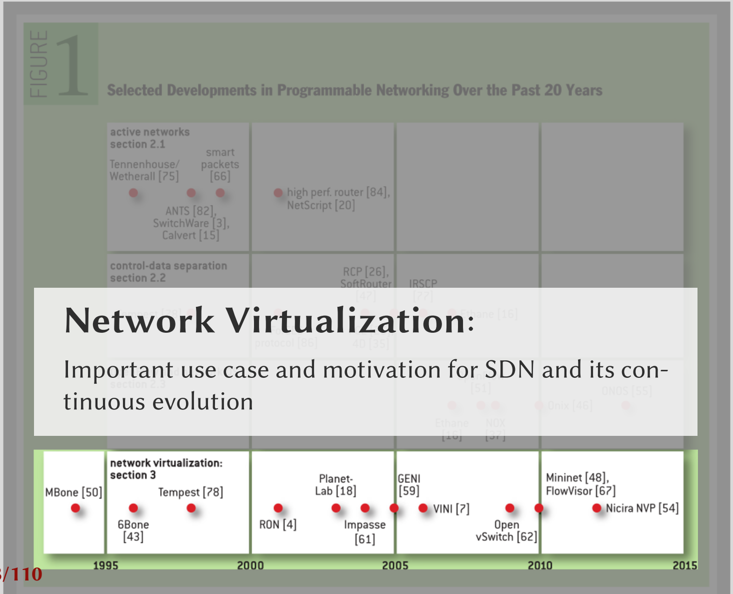
OpenFlow & Network OS:

Bootstrapping SDN and setting architectural foundations of SDN

Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

A Brief History

What makes the SDN today



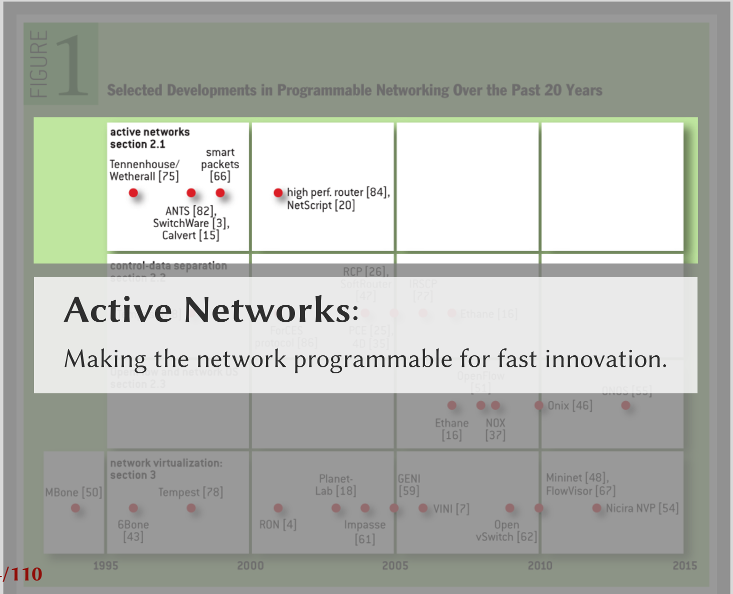
Network Virtualization:

Important use case and motivation for SDN and its continuous evolution

Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

A Brief History

What makes the SDN today



Active Networks:

Making the network programmable for fast innovation.

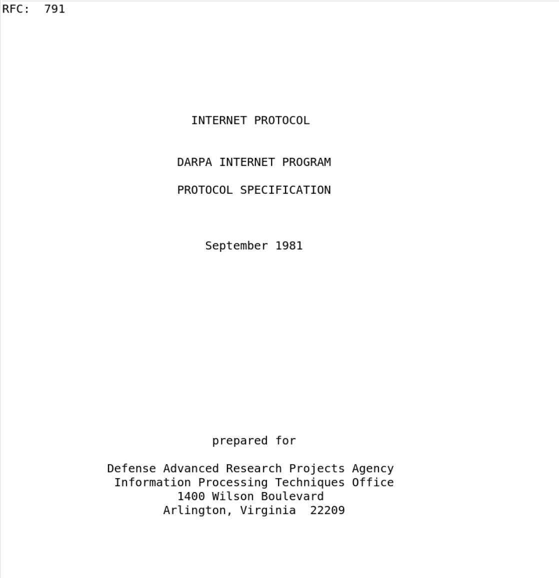
Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

Motivation

Network Infrastructure Must Evolve

In the early 1990s, the Internet starts to take off with the success of Internet Protocol (IP) but the war of protocols was not over

Evolution is critical to infrastructure innovation: reduce the cost and time to try and deploy new ideas



RFC: 791

INTERNET PROTOCOL

DARPA INTERNET PROGRAM

PROTOCOL SPECIFICATION

September 1981

prepared for

Defense Advanced Research Projects Agency
Information Processing Techniques Office
1400 Wilson Boulevard
Arlington, Virginia 22209

Internet Protocol. RFC 791. Sept. 1981. URL: https://rfc-editor.org/rfc/rfc791.txt

Motivation

Different Networks for Different Applications

The IP narrow waist and the end-to-end argument (Saltzer, Reed, and Clark 1984) boosted the rapid development of applications but makes network difficult to customize

Customizable application-specific networking may further enable application innovation

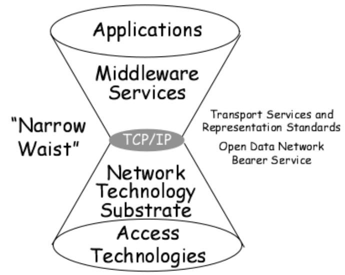


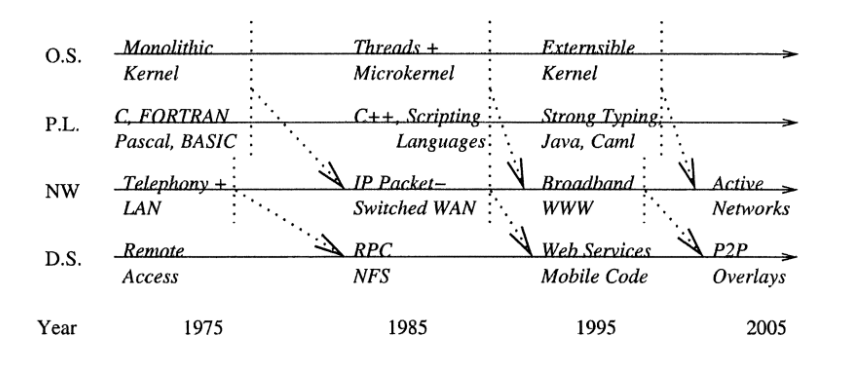
Figure 1. Kleinrock's "Narrow Waist" Model

Randy H Katz, Eric A Brewer, and Steven McCanne. "The Evolution of Internet Services". In: (2015), p. 12

Philosophy

Bring Programmability to the Network

The development of distributed systems had largely influenced the design philosophy of active networking (AN, 主动网络)



J.M. Smith and S.M. Nettles. "Active Networking: One View of the Past, Present, and Future". In: *IEEE Trans. Syst., Man, Cybern. C* 34.1 (Feb. 2004), pp. 4–18. URL: http://ieeexplore.ieee.org/document/1262565/ (visited on 08/15/2021)

Architecture

A Top-down View

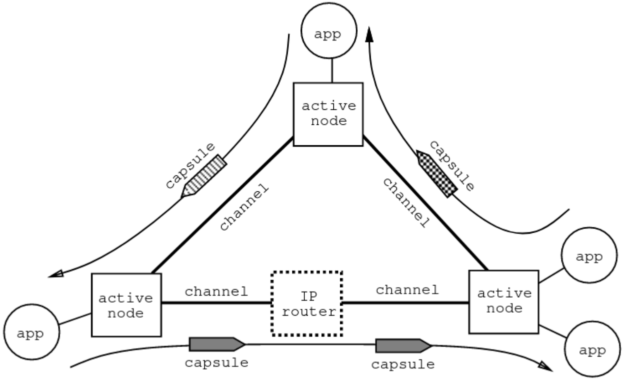


Figure 1. Entities in an ANTS active network

D. Wetherall. "Active Network Vision and Reality: Lessons from a Capsule-Based System". In: *Proceedings DARPA Active Networks Conference and Exposition.* DARPA Active Networks Conference and Exposition. San Francisco, CA, USA: IEEE Comput. Soc, 2002, pp. 25–40. URL: `http://ieeexplore.ieee.org/document/1003482/` (visited on 08/26/2021)

Architecture

A Top-down View

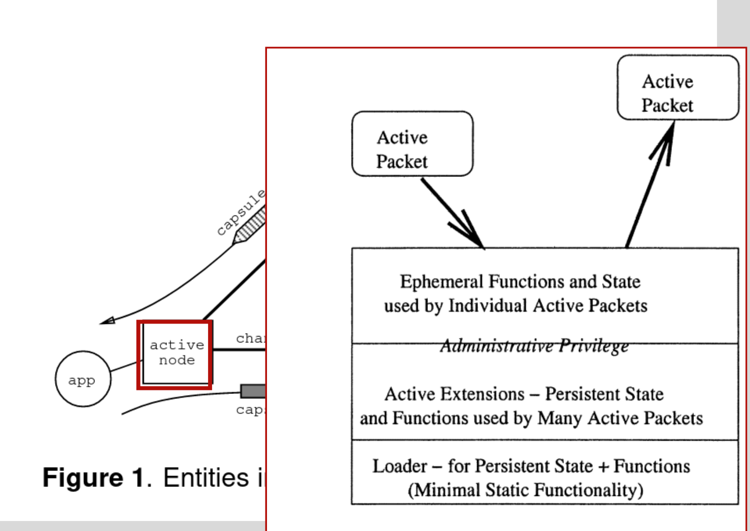


Figure 1. Entities i

Fig. 7. Abstract roles of SwitchWare subsystems.

J.M. Smith and S.M. Nettles. "Active Networking: One View of the Past, Present, and Future". In: *IEEE Trans. Syst., Man, Cybern. C* 34.1 (Feb. 2004), pp. 4–18. URL: http://ieeexplore.ieee.org/document/1262565/ (visited on 08/15/2021)

Architecture

A Top-down View

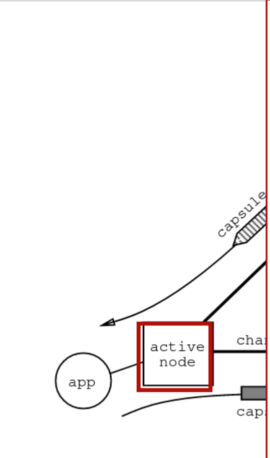


Figure 1. Entities i

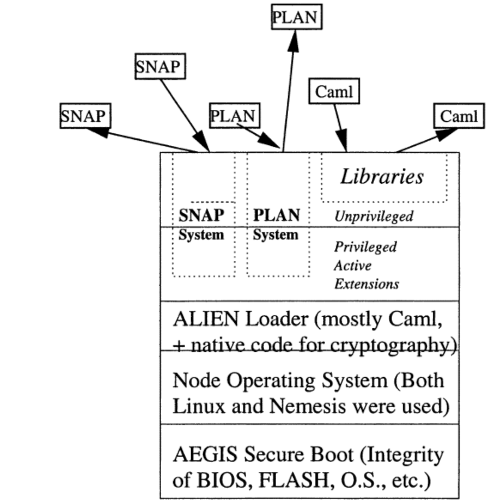


Fig. 9. SwitchWare system architecture.

J.M. Smith and S.M. Nettles. "Active Networking: One View of the Past, Present, and Future". In: *IEEE Trans. Syst., Man, Cybern. C* 34.1 (Feb. 2004), pp. 4–18. URL: http://ieeexplore.ieee.org/document/1262565/ (visited on 08/15/2021)

Multiplexing the Active Node

Virtualization of Active Applications/Extensions

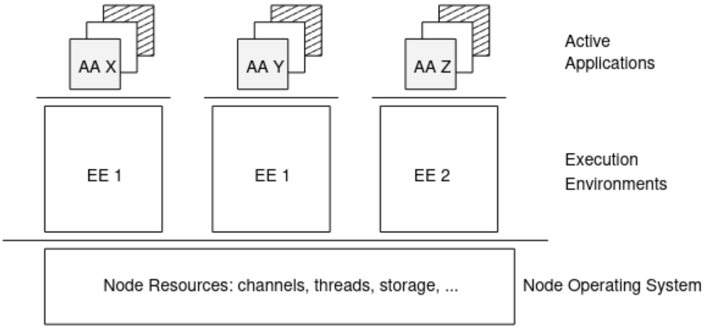


Figure 1: Functional Components of Active Node Architecture

Ken Calvert. “Reflections on Network Architecture: An Active Networking Perspective”. In: *ACM SIGCOMM Computer Communication Review* 36.2 (2006), p. 4

Active Networking

Summary

Active networking is an approach that

- allows dynamic forwarding of packets on a network device

Active Networking

Summary

Active networking is an approach that

- allows dynamic forwarding of packets on a network device
- allows packets to dynamically update the behavior of a network device

Active Networking

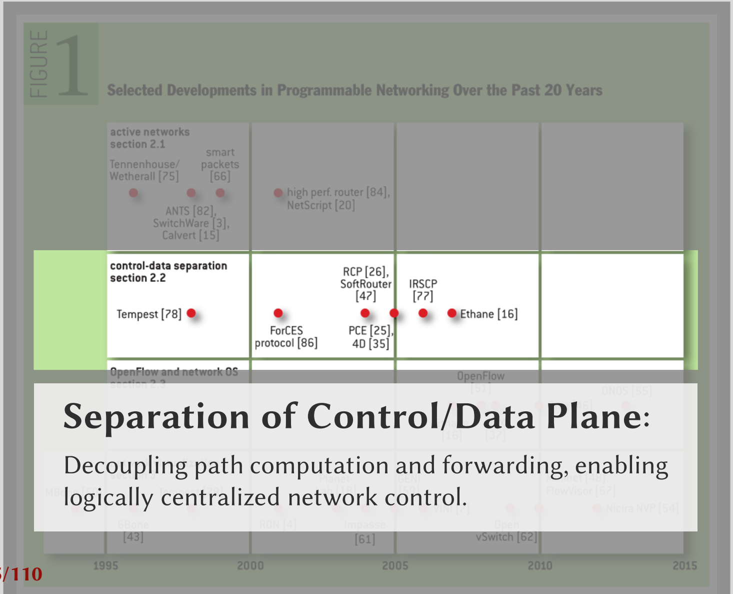
Summary

Active networking is an approach that

- allows dynamic forwarding of packets on a network device
- allows packets to dynamically update the behavior of a network device
- enables continuous evolving of the infrastructure with the bidirectional in-network programmability

A Brief History

What makes the SDN today



Separation of Control/Data Plane:

Decoupling path computation and forwarding, enabling logically centralized network control.

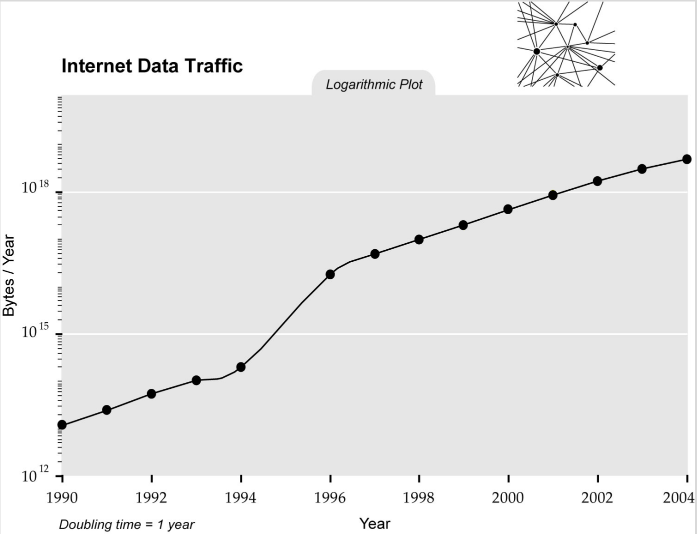
Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

Motivation

Rapid Growth of Internet Traffic

The Internet data traffic grew exponentially in the early 2000s

Managing the traffic with distributed control became difficult



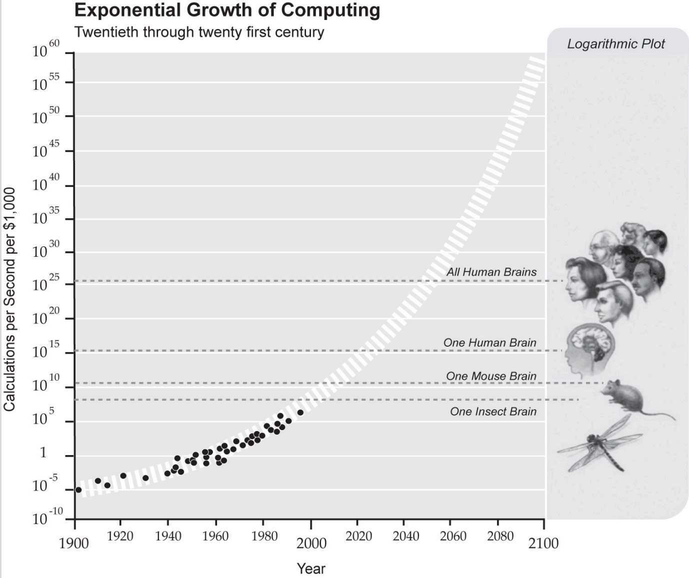
Ray Kurzweil. *The Singularity Is near: When Humans Transcend Biology*. Penguin, 2005

Motivation

Rapid Growth in Computation Power

CPU power also grew rapidly

Decoupling the control plane and data plane (forwarding plane) allow technologies to develop independently



Ray Kurzweil. *The Singularity Is near: When Humans Transcend Biology.* Penguin, 2005

Forwarding and Control Element Separation (ForCES)

Overview

IETF standardized the ForCES framework in 2004 (RFC 3746) and the ForCES protocol in 2010 (RFC 5810)

```
Network Working Group                                          L. Yang
Request for Comments: 3746                                 Intel Corp.
Category: Informational                                       R. Dantu
                                                  Univ. of North Texas
                                                           T. Anderson
                                                           Intel Corp.
                                                              R. Gopal
                                                                 Nokia
                                                            April 2004


     Forwarding and Control Element Separation (ForCES) Framework

Status of this Memo

   This memo provides information for the Internet community.  It does
   not specify an Internet standard of any kind.  Distribution of this
   memo is unlimited.

Copyright Notice

   Copyright (C) The Internet Society (2004).  All Rights Reserved.

Abstract

   This document defines the architectural framework for the ForCES
   (Forwarding and Control Element Separation) network elements, and
   identifies the associated entities and their interactions.
```

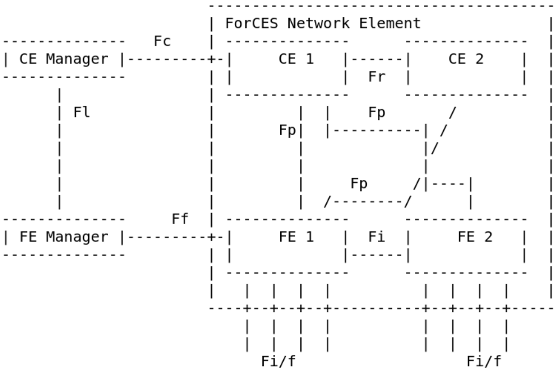
Lily Yang et al. *Forwarding and Control Element Separation (ForCES) Framework.* RFC 3746. Apr. 2004. URL: https://rfc-editor.org/rfc/rfc3746.txt

ForCES

Architecture

The ForCES protocol defines the entities and interfaces for the ForCES framework

FE (Forwarding Element) and CE (Control Element) constitute the data plane (forwarding plane) and control plane respectively




```
Fp: CE-FE interface
Fi: FE-FE interface
Fr: CE-CE interface
Fc: Interface between the CE manager and a CE
Ff: Interface between the FE manager and an FE
Fl: Interface between the CE manager and the FE manager
Fi/f: FE external interface
```

Figure 1: ForCES Architectural Diagram

Joel M. Halpern et al. *Forwarding and Control Element Separation (ForCES) Protocol Specification.* RFC 5810. Mar. 2010. URL: https://rfc-editor.org/rfc/rfc5810.txt

Route Control Platform

Centralized Routing Control

Route control platform (RCP)

- collects network-wide topology information
- makes per-router routing decisions for an entire network

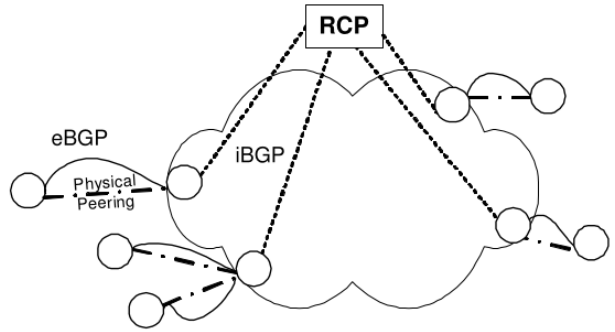


Figure 1: Routing Control Platform (RCP) in an AS

Matthew Caesar et al. “Design and Implementation of a Routing Control Platform”. In: *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2.* NSDI’05. USA: USENIX Association, 2005, pp. 15–28

Path Computation Element (PCE)

Overview

IETF standardized the PCE framework in 2006 (RFC 4655) and the base PCEP in 2009 (RFC 5440)

PCE is mainly used for traffic engineering in commercial label switching networks. The Working Group is still active.

```
Network Working Group                                          A. Farrel
Request for Comments: 4655                           Old Dog Consulting
Category: Informational                                   J.-P. Vasseur
                                                     Cisco Systems, Inc.
                                                                  J. Ash
                                                                    AT&T
                                                             August 2006


           A Path Computation Element (PCE)-Based Architecture


Status of This Memo

   This memo provides information for the Internet community.  It does
   not specify an Internet standard of any kind.  Distribution of this
   memo is unlimited.

Copyright Notice

   Copyright (C) The Internet Society (2006).

Abstract

   Constraint-based path computation is a fundamental building block for
   traffic engineering systems such as Multiprotocol Label Switching
   (MPLS) and Generalized Multiprotocol Label Switching (GMPLS)
   networks.  Path computation in large, multi-domain, multi-region, or
   multi-layer networks is complex and may require special computational
   components and cooperation between the different network domains.

   This document specifies the architecture for a Path Computation
   Element (PCE)-based model to address this problem space.  This
   document does not attempt to provide a detailed description of all
   the architectural components, but rather it describes a set of
   building blocks for the PCE architecture from which solutions may be
   constructed.
```

JP Vasseur, Adrian Farrel, and Gerald Ash. *A Path Computation Element (PCE)-Based Architecture.* RFC 4655. Aug. 2006. URL: https://rfc-editor.org/rfc/rfc4655.txt

PCE

Architecture

There are multiple ways to deploy a PCE including

- PCE in a router (right)
- external PCE with intercommunication (bottom)
- others

JP Vasseur and Jean-Louis Le Roux. *Path Computation Element (PCE) Communication Protocol (PCEP)*. RFC 5440. Mar. 2009. URL: https://rfc-editor.org/rfc/rfc5440.txt

PCE

Architecture

There are multiple ways to deploy a PCE including

- PCE in a router (right)
- external PCE with intercommunication (bottom)
- others

```
                    ---------------
                   |   ---------   | Routing   ----------
                   |  |         |  | Protocol |          |
                   |  |   TED   |<-+----------+->        |
                   |  |         |  |          |          |
                   |   ---------   |          |          |
                   |      |        |          |          |
                   |      | Input  |          |          |
                   |      v        |          |          |
                   |   ---------   |          |          |
                   |  |         |  |          | Adjacent |
                   |  |   PCE   |  |          |   Node   |
                   |  |         |  |          |          |
                   |   ---------   |          |          |
                   |      ^        |          |          |
                   |      |Request |          |          |
                   |      |Response|          |          |
                   |      v        |          |          |
                   |   ---------   |          |          |
          Service  |  |         |  | Signaling|          |
          Request  |  |Signaling|  | Protocol |          |
           ------+->|  Engine   |<-+----------+->        |
                   |  |         |  |          |          |
                   |   ---------   |           ----------
                    ---------------

              Figure 1.  Composite PCE Node
```

JP Vasseur and Jean-Louis Le Roux. *Path Computation Element (PCE) Communication Protocol (PCEP).* RFC 5440. Mar. 2009. URL: https://rfc-editor.org/rfc/rfc5440.txt

PCE

Architecture

There are multiple ways to deploy a PCE including

- PCE in a router (right)
- external PCE with intercommunication (bottom)
- others

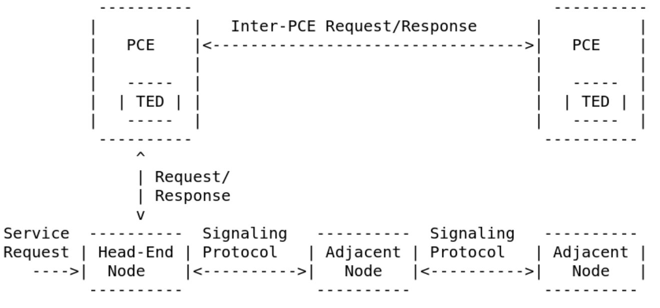


Figure 4. Multiple PCE Path Computation with Inter-PCE Communication

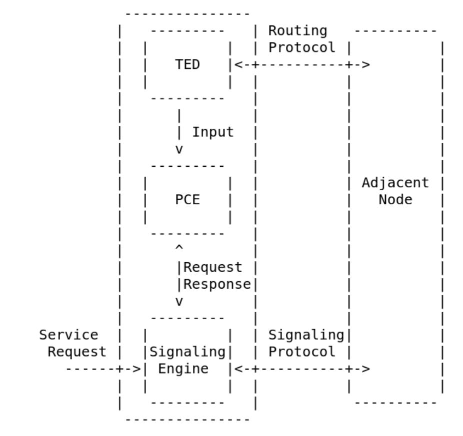


Figure 1. Composite PCE Node

JP Vasseur and Jean-Louis Le Roux. *Path Computation Element (PCE) Communication Protocol (PCEP)*. RFC 5440. Mar. 2009. URL: https://rfc-editor.org/rfc/rfc5440.txt

Summary

Various technologies have explored the direction of separating the control plane and the data plane to

- enable independent evolving of forwarding capabilities and computation power;

Summary

Various technologies have explored the direction of separating the control plane and the data plane to

- enable independent evolving of forwarding capabilities and computation power;
- enable centralized routing for an entire network;

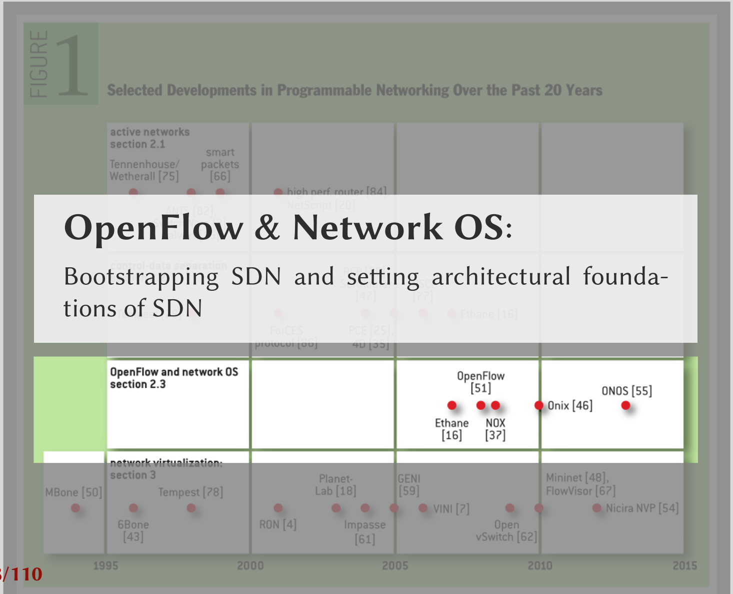
Summary

Various technologies have explored the direction of separating the control plane and the data plane to

- enable independent evolving of forwarding capabilities and computation power;
- enable centralized routing for an entire network;
- realize complex routing based on optimization problems.

A Brief History

What makes the SDN today



OpenFlow & Network OS:

Bootstrapping SDN and setting architectural foundations of SDN

Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

OpenFlow

SDN 1.0

OpenFlow is first introduced in 2008 and then standardized by Open Networking Foundation (ONF) in 2009

The protocol specifies

- forwarding table structure
- interface between switch and controller

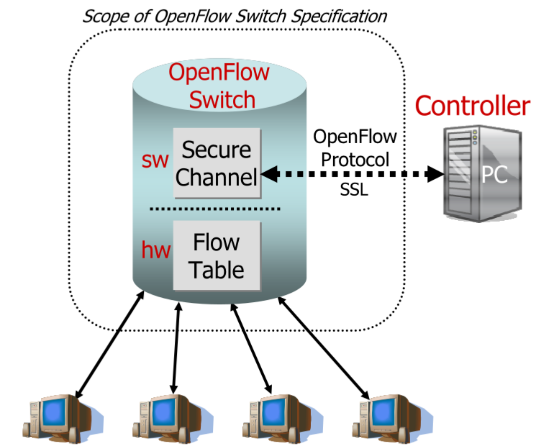


Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

Nick McKeown et al. "OpenFlow: Enabling Innovation in Campus Networks". In: *SIGCOMM Comput. Commun. Rev.* 38.2 (Mar. 2008), pp. 69–74. URL: http://doi.acm.org/10.1145/1355734.1355746

Network Operating Systems

NOX is the first network operating system for SDN (published in 2008)

Many more have emerged in the following years, most notably Onix (Koponen, Casado, et al. 2010), ONOS (Berde et al. 2014) and OpenDaylight (Medved et al. 2014)

Learn more details in Week 4

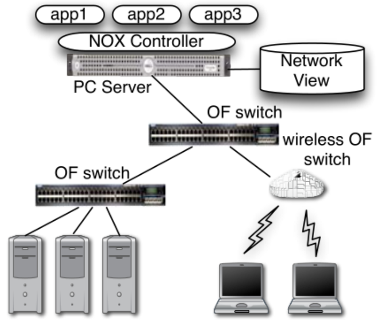
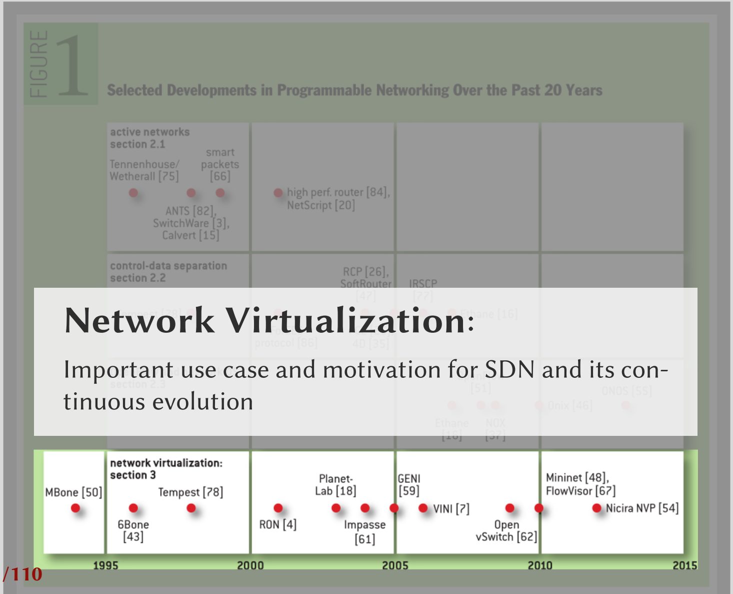


Figure 1: Components of a NOX-based network: OpenFlow (OF) switches, a server running a NOX controller process and a database containing the network view.

Natasha Gude et al. "NOX: Towards an Operating System for Networks". In: *SIGCOMM Comput. Commun. Rev.* 38.3 (July 2008), pp. 105–110. URL: http://doi.acm.org/10.1145/1384609.1384625

A Brief History

What makes the SDN today



Network Virtualization:

Important use case and motivation for SDN and its continuous evolution

Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327

Network Virtualization

Different Scope of Network Virtualization

Network virtualization covers a large scope: from virtualization of hardware devices such as NIC (LEFT), router, etc., to network overlay (RIGHT) or slicing

SDN is mostly used for the latter

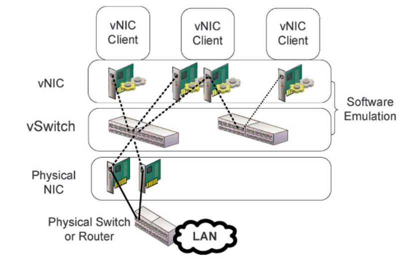


Fig. 1. General NIC Virtualization Architecture.

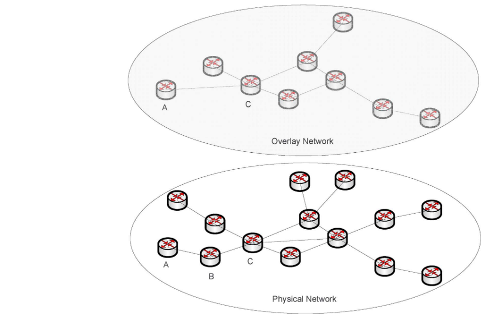


Fig. 3. Overlay Network.

Anjing Wang et al. “Network Virtualization: Technologies, Perspectives, and Frontiers”. In: *J. Lightwave Technol.* 31.4 (Feb. 2013), pp. 523–537. URL: http://ieeexplore.ieee.org/document/6272301/ (visited on 08/27/2021)

Network Virtualization

How Network Virtualization boosts SDN

- Cloud computing makes network virtualization an important industrial requirement
- Increasing numbers of cloud tenants make it difficult to manage virtualized network with conventional networking technologies
- SDN reduces the complexity and substantially improves the efficiency of network virtualization

> **Dynamic infrastructure requirements make SDN a critical component of private cloud development and external service consumption**
>
> A key component of any cloud computing strategy is making IT more agile and, therefore, more responsive to the needs of the business. Historically, it could take two months to provision physical servers and two weeks to provision the network. With the adoption of server virtualization, server provisioning is measured in hours. This makes the two weeks needed to provision the network a key bottleneck and a barrier to business agility.

David Cearley et al. *Gartner Note: Top 10 Technology Trends, 2013: Cloud Computing and Hybrid IT Drive Future IT Models.* 2013. URL: https://www.cisco.com/c/dam/en_us/solutions/trends/cloud/docs/future_of_it.pdf (visited on 08/27/2021)

So what is SDN?

SDN as a Network Architecture

- Separated control and data planes
- Fine-grained forwarding capabilities
- Logically centralized network OS with standard cross-layer abstractions and protocols
- Programmable control plane

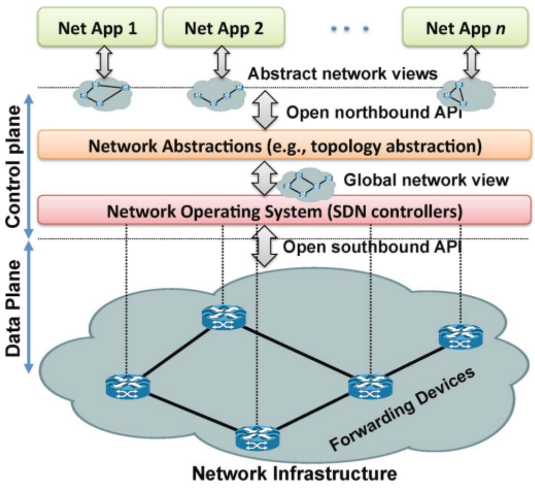


Fig. 4. *SDN architecture and its fundamental abstractions.*

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

SDN as a Network Architecture

- Separated control and data planes
- Fine-grained forwarding capabilities
- Logically centralized network OS with standard cross-layer abstractions and protocols
- Programmable control plane
- Programmable data plane

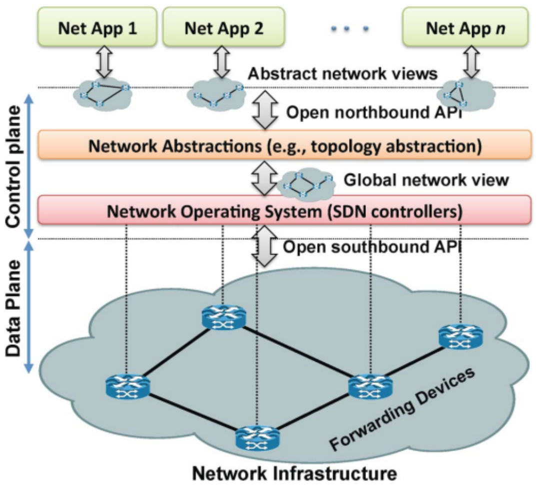


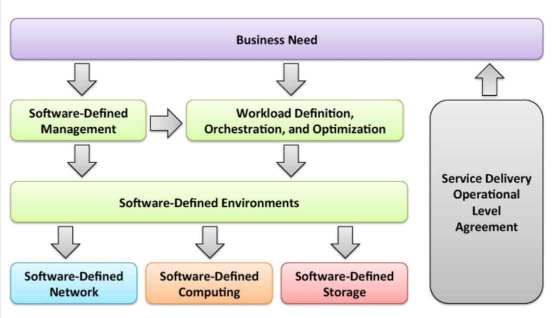
Fig. 4. ***SDN architecture and its fundamental abstractions.***

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

SDN as a Technology Trend

SDN is also identified as a technology trend that

- brings programmability to networking (Nadeau and Gray 2013; Feamster, Rexford, and Zegura 2013)
- enables tighter application-network integration
- achieves fully automated integrated service and resource orchestration (Li et al. 2014; ETSI 2019)



D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

SDN in the Course

What We Talk about When We Talk about SDN

We focus on the network architecture aspect of SDN

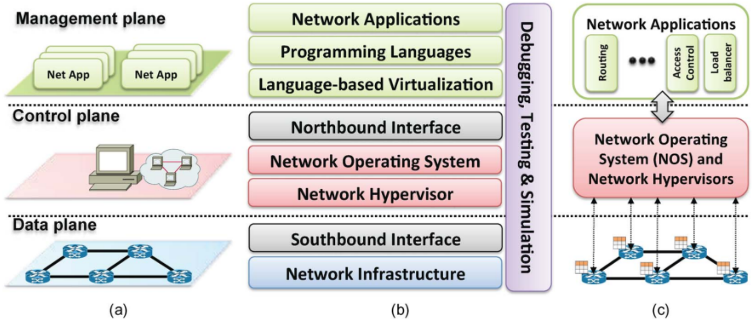


Fig. 6. *Software-Defined Networks in (a) planes, (b) layers, and (c) system design architecture.*

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

SDN in the Course

What We Talk about When We Talk about SDN

We focus on the network architecture aspect of SDN

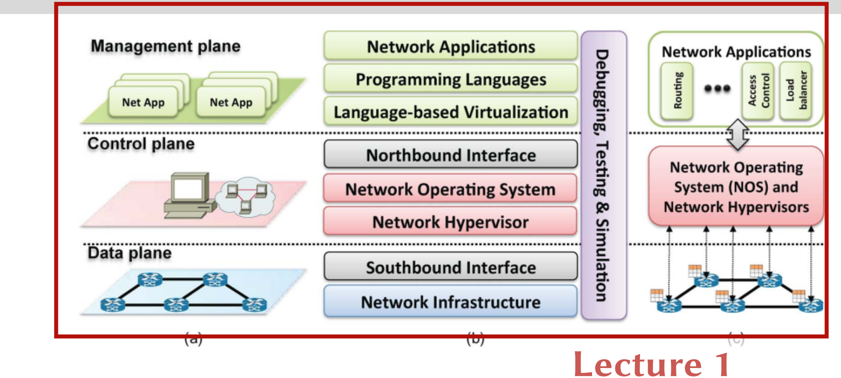


Fig. 6. *Software-Defined Networks in (a) planes, (b) layers, and (c) system design architecture.*

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

SDN in the Course

What We Talk about When We Talk about SDN

We focus on the network architecture aspect of SDN

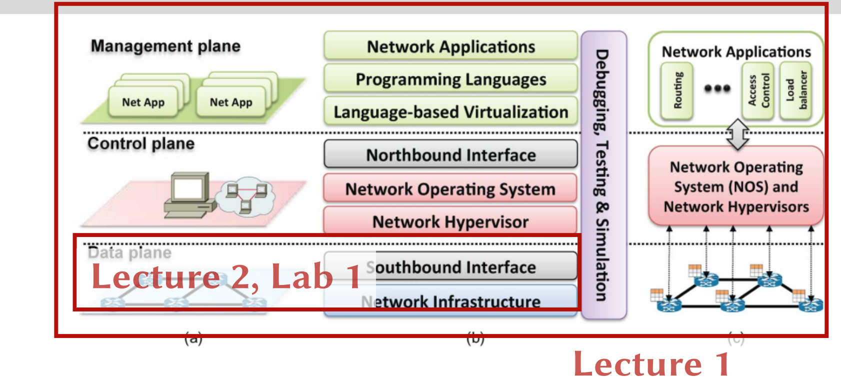


Fig. 6. *Software-Defined Networks in (a) planes, (b) layers, and (c) system design architecture.*

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

SDN in the Course

What We Talk about When We Talk about SDN

We focus on the network architecture aspect of SDN

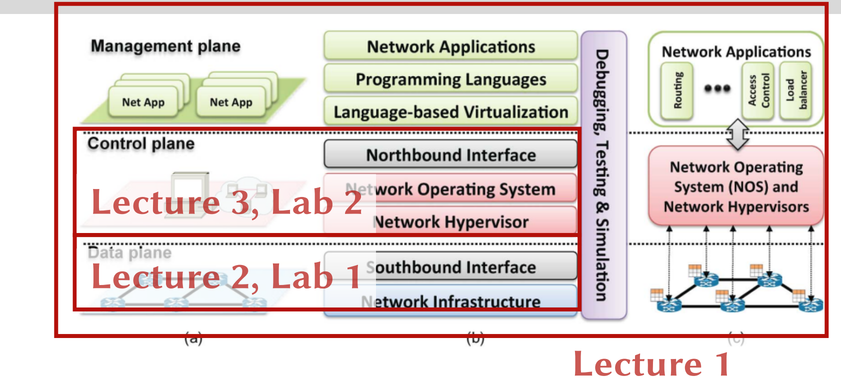


Fig. 6. *Software-Defined Networks in (a) planes, (b) layers, and (c) system design architecture.*

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

SDN in the Course

What We Talk about When We Talk about SDN

We focus on the network architecture aspect of SDN

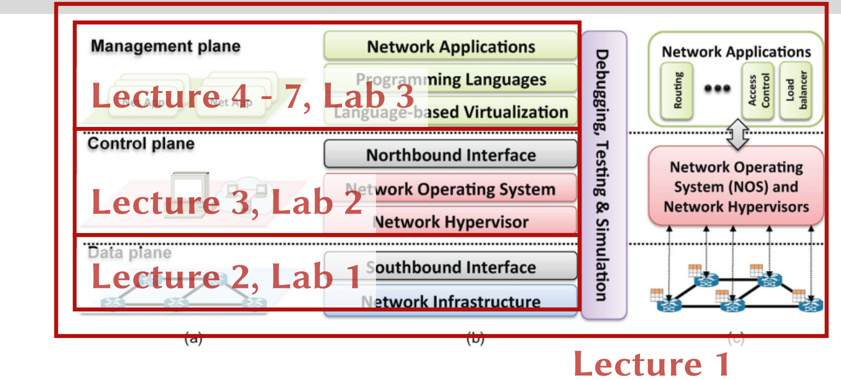


Fig. 6. Software-Defined Networks in (a) planes, (b) layers, and (c) system design architecture.

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

Network Infrastructure

What is it?

- OpenFlow-enabled forwarding devices
- Middleboxes or Virtualized Network Functions (VNF)

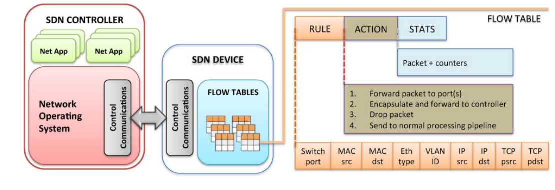


Fig. 7. *OpenFlow-enabled SDN devices.*

D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76

What you are expected

- Understand the concept of match-action table (匹配行为表) and how they are realized
- Understand common match and action defined in the OpenFlow specification
- Learn about common OpenFlow software and hardware devices and understand how to use Open VSwitch (OVS)
- Learn about more programmable network devices

Southbound Interface

What is it?

- API and protocols between the physical or virtual device to the controller

What you are expected

- Understand the basic message types of OpenFlow
- Learn about other southbound interfaces (NETCONF, SNMP, etc.)

Network Hypervisor

What is it?

- Resource provisioning and management, including devices, address space, bandwidth, etc.
- Compose virtual FIBs into physical FIB

What you are expected

- Understand common goals in network hypervisor

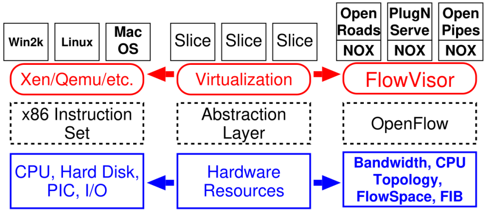


Figure 1: Similar to computer virtualization, FlowVisor is a network virtualization layer that resides bet the hardware and software architectural components. OpenRoads, PlugNServe, and OpenPipes are exa of virtual network controllers built on NOX(§ 6).

Rob Sherwood et al. "FlowVisor: A Network Virtualization Layer". In: *OpenFlow Switch Consortium, Tech. Rep* (2009). URL: http://sb.tmit.bme.hu/mediawiki/images/c/c0/FlowVisor.pdf (visited on 06/26/2014)

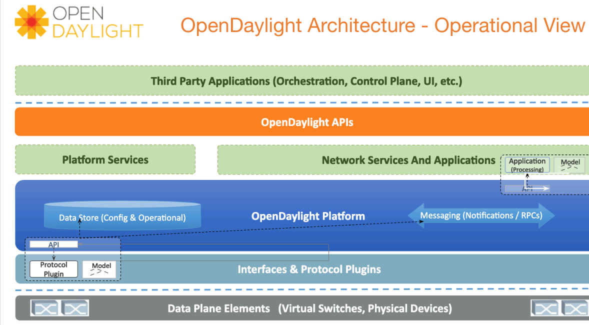
Network Operating System

What is it?

- Platforms that provide abstractions, essential services, and common APIs to application developers
- Resource management and execution environment for applications

What you are expected

- Learn about design choices of NOS
- Learn about common NOS and how to use and develop for OpenDaylight

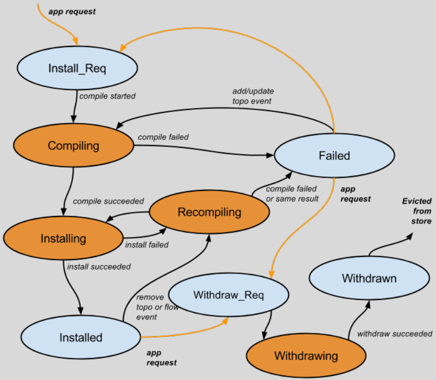


OpenDaylight. *OpenDaylight Architecture*. 2021. URL: https://www.opendaylight.org/what-we-do/current-release/neon/attachment/opendaylight-architecture

Northbound Interface

What is it?

- API and protocols between the NOS and applications



What you are expected

- Understand the idea of intents
- Understand RESTful API and how to use the YANG modeling language
- Learn about basic standard interfaces in OpenDaylight

ONOS. *Intent Framework*. 2016. URL: https://wiki.onosproject.org/display/ONOS/Intent+Framework

Language-based Virtualization

What is it?

- Abstractions and virtualization support offered by programming languages

What you are expected

Programming Languages

What is it?

- Syntax, semantics and execution environments of programming languages that are designed for SDN

Syntax

Fields	$f ::=$	$f_1 \mid \cdots \mid f_k$	
Packets	$pk ::=$	$\{f_1 = v_1, \cdots, f_k = v_k\}$	
Histories	$h ::=$	$pk::\langle\rangle \mid pk::h$	
Predicates	$a, b ::=$	1	*Identity*
	$\mid$	0	*Drop*
	$\mid$	$f = n$	*Test*
	$\mid$	$a + b$	*Disjunction*
	$\mid$	$a \cdot b$	*Conjunction*
	$\mid$	$\neg a$	*Negation*
Policies	$p, q ::=$	a	*Filter*
	$\mid$	$f \leftarrow n$	*Modification*
	$\mid$	$p + q$	*Union*
	$\mid$	$p \cdot q$	*Sequential composition*
	$\mid$	p^*	*Kleene star*
	$\mid$	dup	*Duplication*

What you are expected

- Learn about several SDN programming languages
- Understand the science behind the selected set of programming languages (Kleen Algebra, Trace Tree, Product Graph, Mixed Integer Linear Programming)

Carolyn Jane Anderson et al. “NetKAT: Semantic Foundations for Networks”. In: *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL ’14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 113–126. URL: https://doi.org/10.1145/2535838.2535862

Network Applications

What is it?

- Network management applications
- Network optimizers for certain end-to-end applications

What you are expected

- Identify a scenario and try to manage the network the SDN technology

SDN in Action

Industry Application and Deployment

Most successful and influential application of SDN

- VMWare/Nicira: NVP (Koponen, Amidon, et al. 2014, Network virtualization)
- Google: B4 (Hong et al. 2018, WAN traffic engineering)

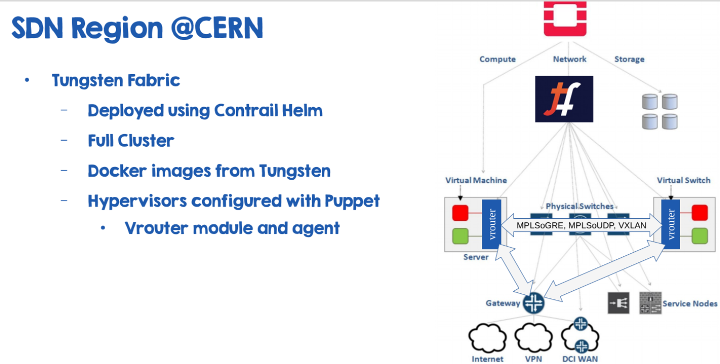
> *... Google's WAN, B4, using Software Defined Networking (SDN) principles and OpenFlow to manage individual switches. ... These features allow many B4 links to run at near 100% utilization and all links to average 70% utilization over long time periods, corresponding to 2-3x efficiency improvements relative to standard practice.*

Sushant Jain et al. "B4: Experience with a Globally-deployed Software Defined Wan". In: *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*. SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 3–14. URL: http://doi.acm.org/10.1145/2486001.2486019

SDN in Action

SDN in Cloud

SDN has been integrated into cloud platforms and will be used in large-scale scientific networks



Sriram Subramanian and Sreenivas Voruganti. *Software-Defined Networking (SDN) with OpenStack*. Packt Publishing Ltd, 2016

José Castro León. “Introducing SDN in CERN Cloud”. (Melbourne, Australia). Apr. 18, 2019

SDN in Action

SDN in Scientific Research Platforms

SDN has become the basic component of major network research projects

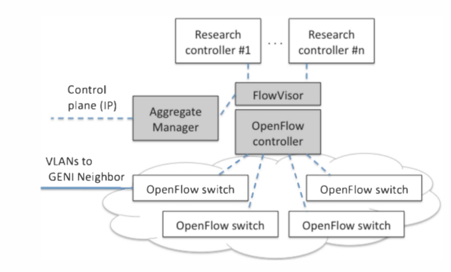


Fig. 1. Schematic of GENI SDN components at a single site

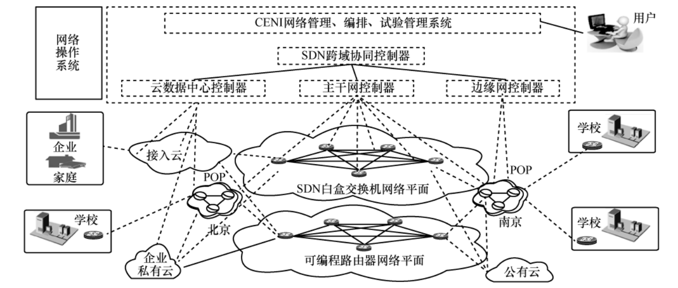


图 1 CENI 的整体架构

Mark Berman, Chip Elliott, and Lawrence Landweber. "GENI: Large-scale Distributed Infrastructure for Networking and Distributed Systems Research". In: *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE).* 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE). July 2014, pp. 156–161

黄韬 et al. "未来网络技术与发展趋势综述". In: 通信学报 42.1 (2021), pp. 130–150, CENI, 未来网络试验设施

The End

Thanks!

kaigao@scu.edu.cn

References I

[1] Carolyn Jane Anderson et al. "NetKAT: Semantic Foundations for Networks". In: *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 113–126. URL: https://doi.org/10.1145/2535838.2535862.

[2] Pankaj Berde et al. "ONOS: Towards an Open, Distributed SDN OS". In: *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*. HotSDN '14. New York, NY, USA: ACM, 2014, pp. 1–6. URL: http://doi.acm.org/10.1145/2620728.2620744.

[3] Mark Berman, Chip Elliott, and Lawrence Landweber. "GENI: Large-scale Distributed Infrastructure for Networking and Distributed Systems Research". In: *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*. 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE). July 2014, pp. 156–161.

[4] Matthew Caesar et al. "Design and Implementation of a Routing Control Platform". In: *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*. NSDI'05. USA: USENIX Association, 2005, pp. 15–28.

[5] Ken Calvert. "Reflections on Network Architecture: An Active Networking Perspective". In: *ACM SIGCOMM Computer Communication Review* 36.2 (2006), p. 4.

[6] David Cearley et al. *Gartner Note: Top 10 Technology Trends, 2013: Cloud Computing and Hybrid IT Drive Future IT Models*. 2013. URL: https://www.cisco.com/c/dam/en_us/solutions/trends/cloud/docs/future_of_it.pdf (visited on 08/27/2021).

[7] ZSM ETSI. *ZSM 002 Draft: Zero-touch Network and Service Management (ZSM); Reference Architecture*. 2019.

References II

[8] Nick Feamster, Jennifer Rexford, and Ellen Zegura. "The Road to SDN: An Intellectual History of Programmable Networks". In: *Queue* 11.12 (Dec. 2013), 20:20–20:40. URL: http://doi.acm.org/10.1145/2559899.2560327.

[9] Natasha Gude et al. "NOX: Towards an Operating System for Networks". In: *SIGCOMM Comput. Commun. Rev.* 38.3 (July 2008), pp. 105–110. URL: http://doi.acm.org/10.1145/1384609.1384625.

[10] Joel M. Halpern et al. *Forwarding and Control Element Separation (ForCES) Protocol Specification.* RFC 5810. Mar. 2010. URL: https://rfc-editor.org/rfc/rfc5810.txt.

[11] Chi-Yao Hong et al. "B4 and After: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google's Software-defined WAN". In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication.* SIGCOMM '18. New York, NY, USA: ACM, 2018, pp. 74–87. URL: http://doi.acm.org/10.1145/3230543.3230545.

[12] *Internet Protocol.* RFC 791. Sept. 1981. URL: https://rfc-editor.org/rfc/rfc791.txt.

[13] Sushant Jain et al. "B4: Experience with a Globally-deployed Software Defined Wan". In: *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM.* SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 3–14. URL: http://doi.acm.org/10.1145/2486001.2486019.

[14] Randy H Katz, Eric A Brewer, and Steven McCanne. "The Evolution of Internet Services". In: (2015), p. 12.

[15] Teemu Koponen, Keith Amidon, et al. "Network Virtualization in Multi-tenant Datacenters". In: *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14).* Seattle, WA: USENIX Association, 2014, pp. 203–216. URL: https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/koponen.

References III

[16] Teemu Koponen, Martin Casado, et al. "Onix: A Distributed Control Platform for Large-scale Production Networks.". In: *OSDI*. Vol. 10. 2010, pp. 1–6. URL: http://static.usenix.org/events/osdi10/tech/full_papers/Koponen.pdf (visited on 06/26/2014).

[17] D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76.

[18] Ray Kurzweil. *The Singularity Is near: When Humans Transcend Biology*. Penguin, 2005.

[19] José Castro León. "Introducing SDN in CERN Cloud". (Melbourne, Australia). Apr. 18, 2019.

[20] C.-S. Li et al. "Software Defined Environments: An Introduction". In: *IBM J. Res. & Dev.* 58.2/3 (Mar. 2014), 1:1–1:11. URL: http://ieeexplore.ieee.org/document/6798712/ (visited on 08/27/2021).

[21] Nick McKeown et al. "OpenFlow: Enabling Innovation in Campus Networks". In: *SIGCOMM Comput. Commun. Rev.* 38.2 (Mar. 2008), pp. 69–74. URL: http://doi.acm.org/10.1145/1355734.1355746.

[22] Jan Medved et al. "Opendaylight: Towards a Model-Driven SDN Controller Architecture". In: *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. 2014.

[23] Thomas D. Nadeau and Kenneth Gray. *SDN: Software Defined Networks*. First edition. Beijing: O'Reilly, 2013. 352 pp.

[24] ONOS. *Intent Framework*. 2016. URL: https://wiki.onosproject.org/display/ONOS/Intent+Framework.

[25] OpenDaylight. *OpenDaylight Architecture*. 2021. URL: https://www.opendaylight.org/what-we-do/current-release/neon/attachment/opendaylight-architecture.

References IV

[26] J. H. Saltzer, D. P. Reed, and D. D. Clark. "End-to-End Arguments in System Design". In: *ACM Trans. Comput. Syst.* 2.4 (Nov. 1984), pp. 277–288. URL: https://dl.acm.org/doi/10.1145/357401.357402 (visited on 08/26/2021).

[27] Rob Sherwood et al. "FlowVisor: A Network Virtualization Layer". In: *OpenFlow Switch Consortium, Tech. Rep* (2009). URL: http://sb.tmit.bme.hu/mediawiki/images/c/c0/FlowVisor.pdf (visited on 06/26/2014).

[28] J.M. Smith and S.M. Nettles. "Active Networking: One View of the Past, Present, and Future". In: *IEEE Trans. Syst., Man, Cybern. C* 34.1 (Feb. 2004), pp. 4–18. URL: http://ieeexplore.ieee.org/document/1262565/ (visited on 08/15/2021).

[29] Sriram Subramanian and Sreenivas Voruganti. *Software-Defined Networking (SDN) with OpenStack*. Packt Publishing Ltd, 2016.

[30] JP Vasseur, Adrian Farrel, and Gerald Ash. *A Path Computation Element (PCE)-Based Architecture*. RFC 4655. Aug. 2006. URL: https://rfc-editor.org/rfc/rfc4655.txt.

[31] JP Vasseur and Jean-Louis Le Roux. *Path Computation Element (PCE) Communication Protocol (PCEP)*. RFC 5440. Mar. 2009. URL: https://rfc-editor.org/rfc/rfc5440.txt.

[32] Anjing Wang et al. "Network Virtualization: Technologies, Perspectives, and Frontiers". In: *J. Lightwave Technol.* 31.4 (Feb. 2013), pp. 523–537. URL: http://ieeexplore.ieee.org/document/6272301/ (visited on 08/27/2021).

[33] D. Wetherall. "Active Network Vision and Reality: Lessons from a Capsule-Based System". In: *Proceedings DARPA Active Networks Conference and Exposition*. DARPA Active Networks Conference and Exposition. San Francisco, CA, USA: IEEE Comput. Soc, 2002, pp. 25–40. URL: http://ieeexplore.ieee.org/document/1003482/ (visited on 08/26/2021).

References V

[34] Lily Yang et al. *Forwarding and Control Element Separation (ForCES) Framework*. RFC 3746. Apr. 2004. URL: https://rfc-editor.org/rfc/rfc3746.txt.

[35] 黄韬 et al. “未来网络技术与发展趋势综述”. In: 通信学报 42.1 (2021), pp. 130–150.