SDN Modern Networking Fundamentals

Motivations and Background

Future Internet Laboratory
School of Electronics and Telecommunications, Hanoi University of Science and Technology
Contact: thanh.nguyenhuu@hust.edu.vn

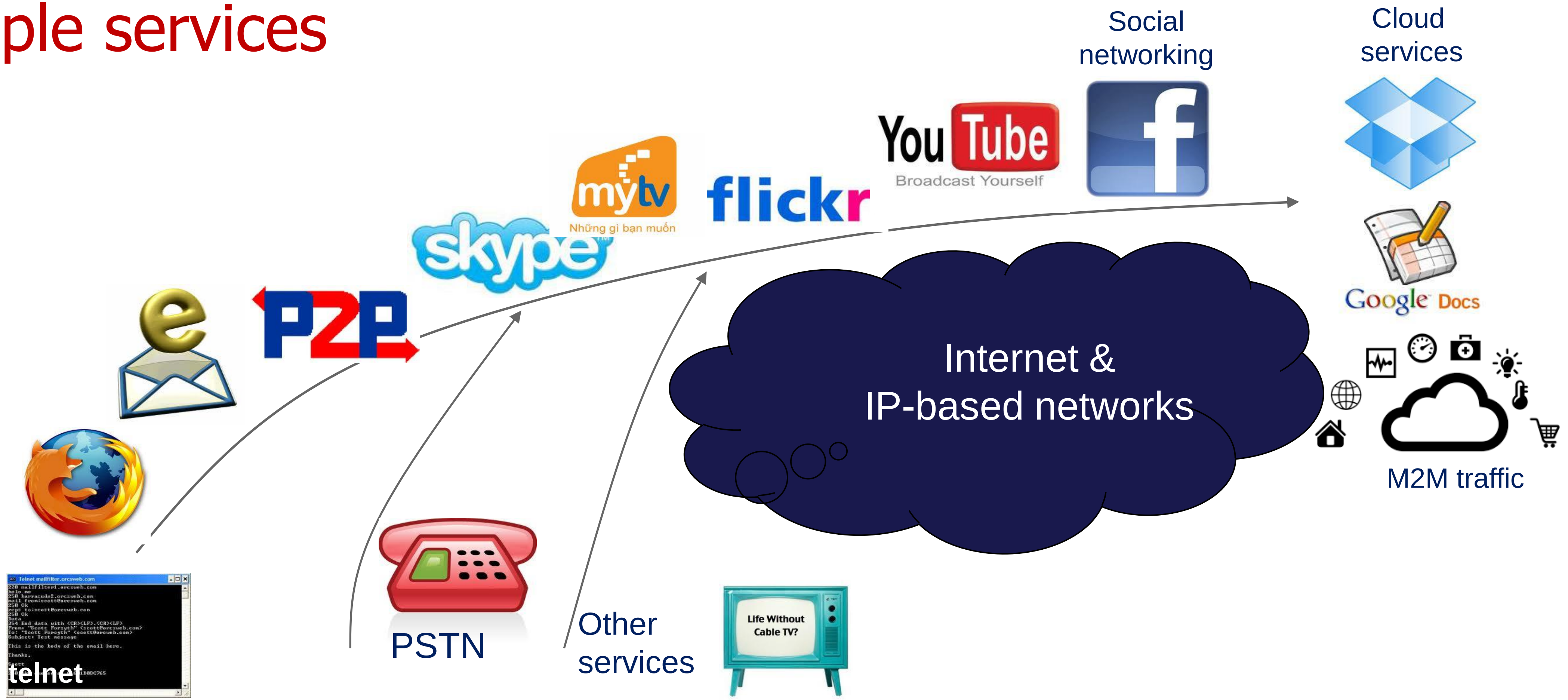
Contents

- Motivations
- SDN - Introduction
  - Why SDN?
  - What is SDN?
- Basic concepts

- Motivations
- SDN - Introduction
  - Why SDN?
  - What is SDN?
- History of programmable networks and SDN
- Basic concepts

Current Trends

- Internet Status and Trend: Converged networks, integrated, multiple services



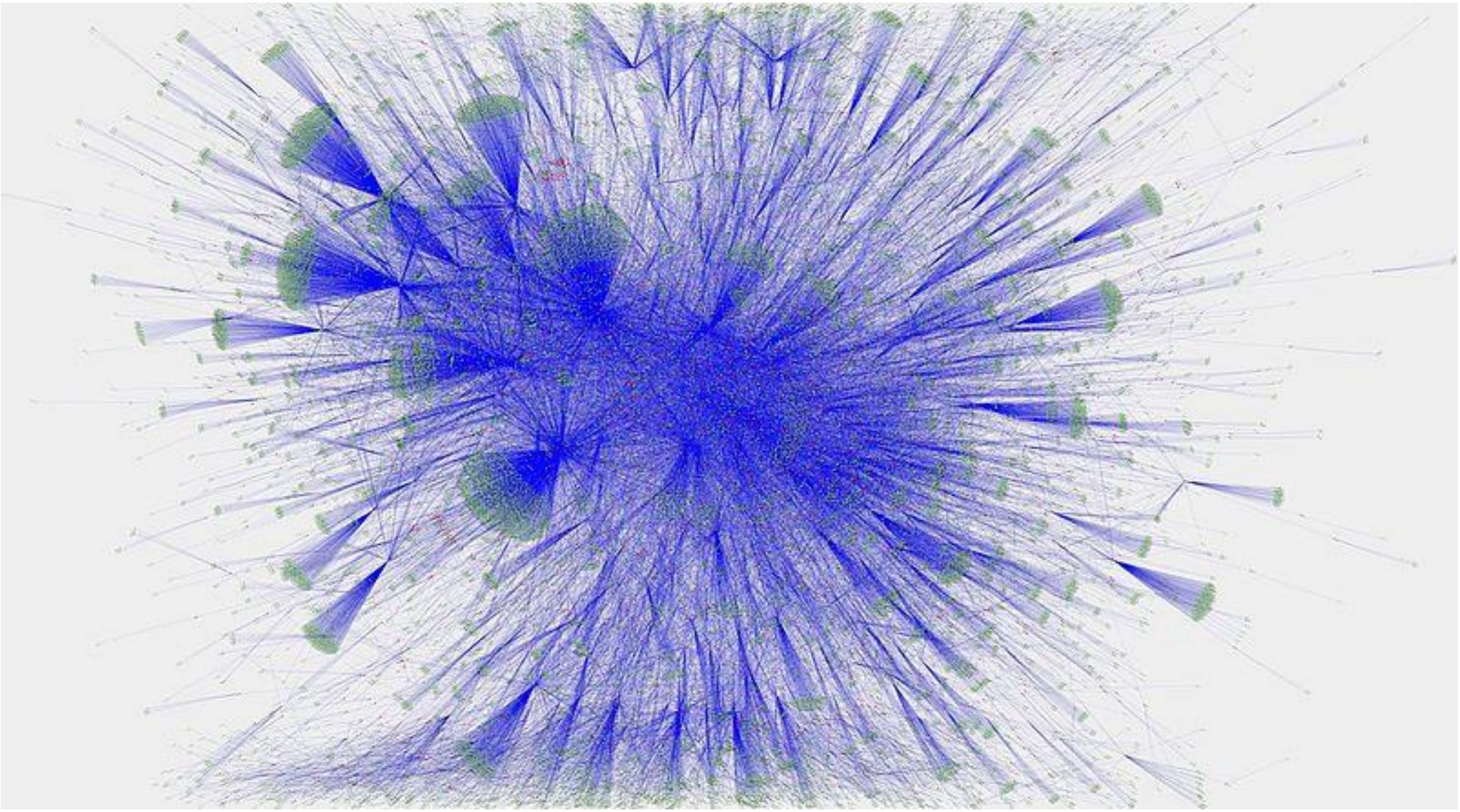
Traffic and Network Services

- 40% per annum growth in network traffic
- 10% per annum growth in number of users
- New applications lead to increased traffic volume in the core of the network
  - Cloud applications and services:
    - Software as a Service: Search, email, social networking, data mining, utility computing
      - Google Docs etc.
      - Social networking: Facebook, Twitter etc.
      - Flickr, YouTube
    - Storage as a Service:
      - Dropbox, Sky Drive
    - Processing as a Service
  - P2P applications: Bittorrent
  - Machine-to-machine traffic is on the rise

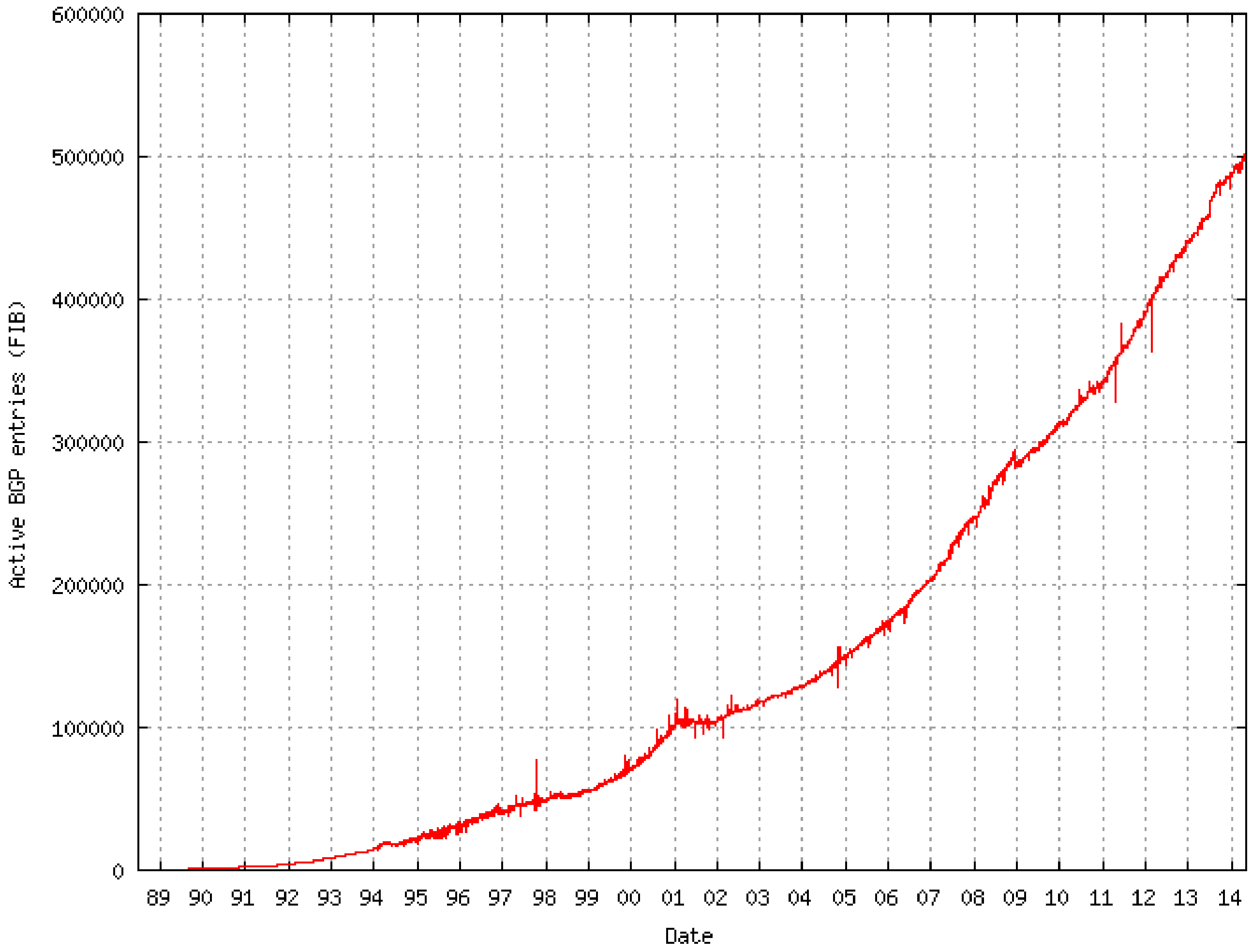
Global Internet Map - 2016

- Internet BGP peering map 2016

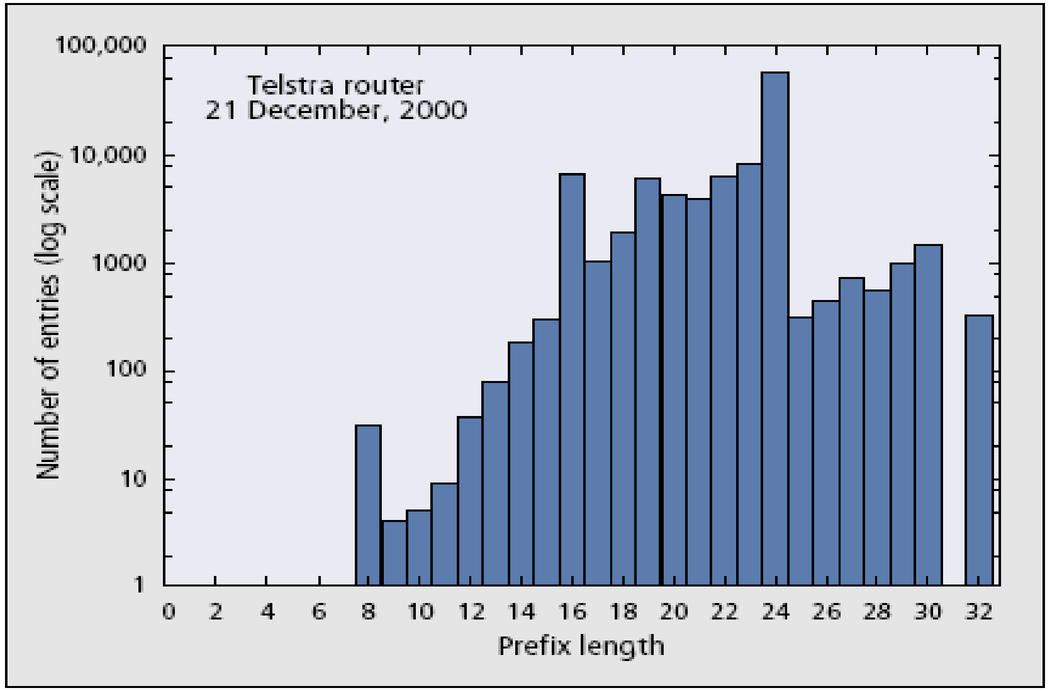
Source: network mapping - Wikipedia



Number of BGP Entries in Core Routers



Source: http://www.cidr-report.org/



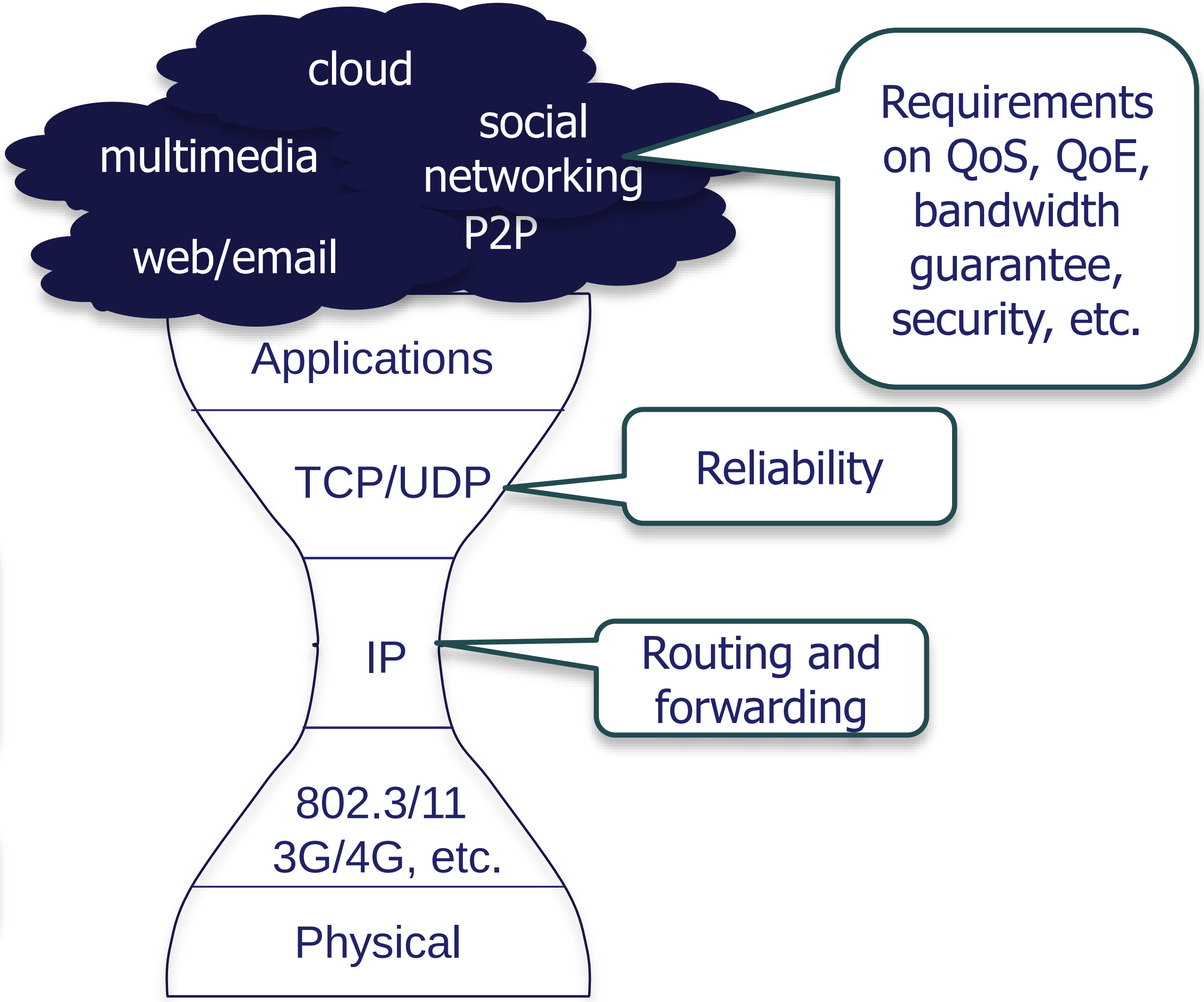
Prefix length distribution in core routers

→ The Internet is becoming very complex

VIỆN ĐIỆN TỬ - VIỄN THÔNG
School of Electronics and Telecommunications

Current Network's Problem Statements

Diversity in applications and requirements

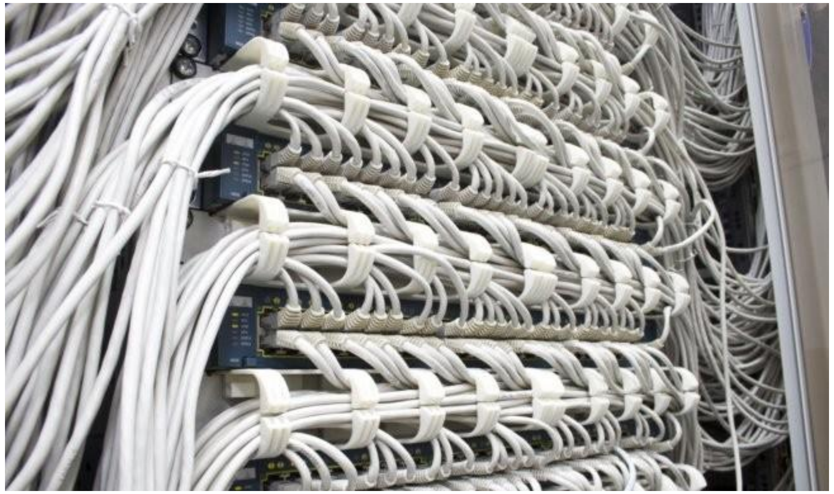


"IP bottle neck": everything running on top of IP

Technology dependent

Current Network's Problem Statements

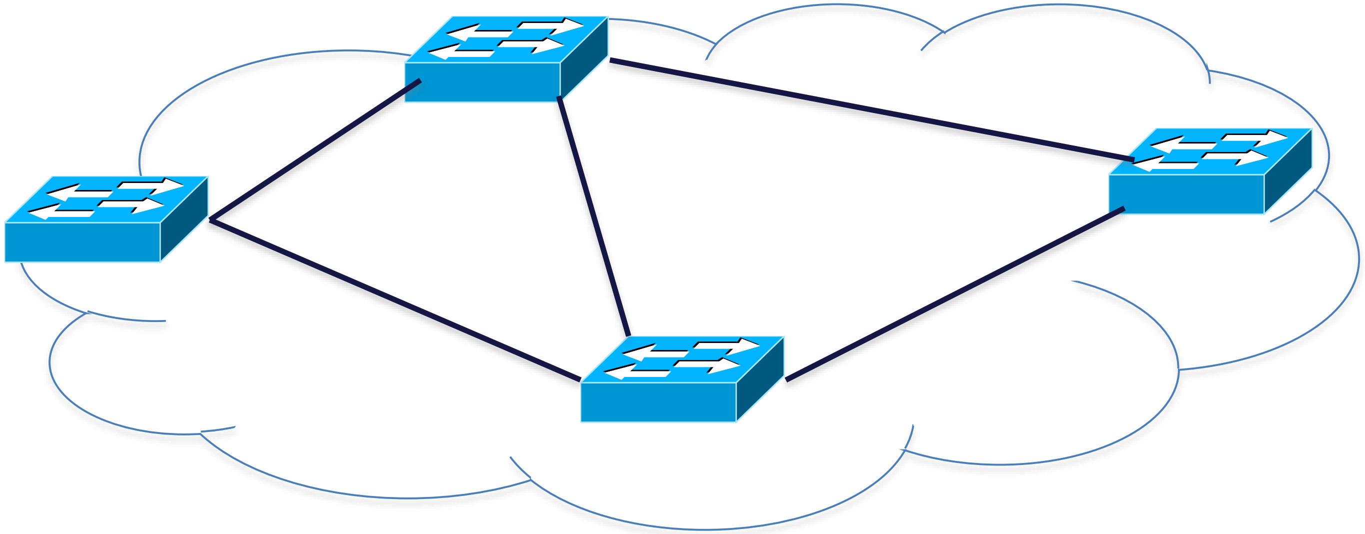
- Traditional networking approaches: closed, and proprietary
- Networks are too complex and hard to manage
  - Highly complex as the computation and storage have been virtualized
  - Network administrators large share of sysadmin staff
- Networks are hard to evolve
  - Difficult to optimize: How to optimize routes, traffic, services and functions of the networks?
  - Difficult to customize: How to add new functionalities depending on individual's needs?
- Costly
  - Operating costs grow with the complexity



Current Network's Problem Statements

- Typical networking operations
  - Management plane
  - Control Plane – The brain/decision maker
  - Data Plane – Packet forwarder

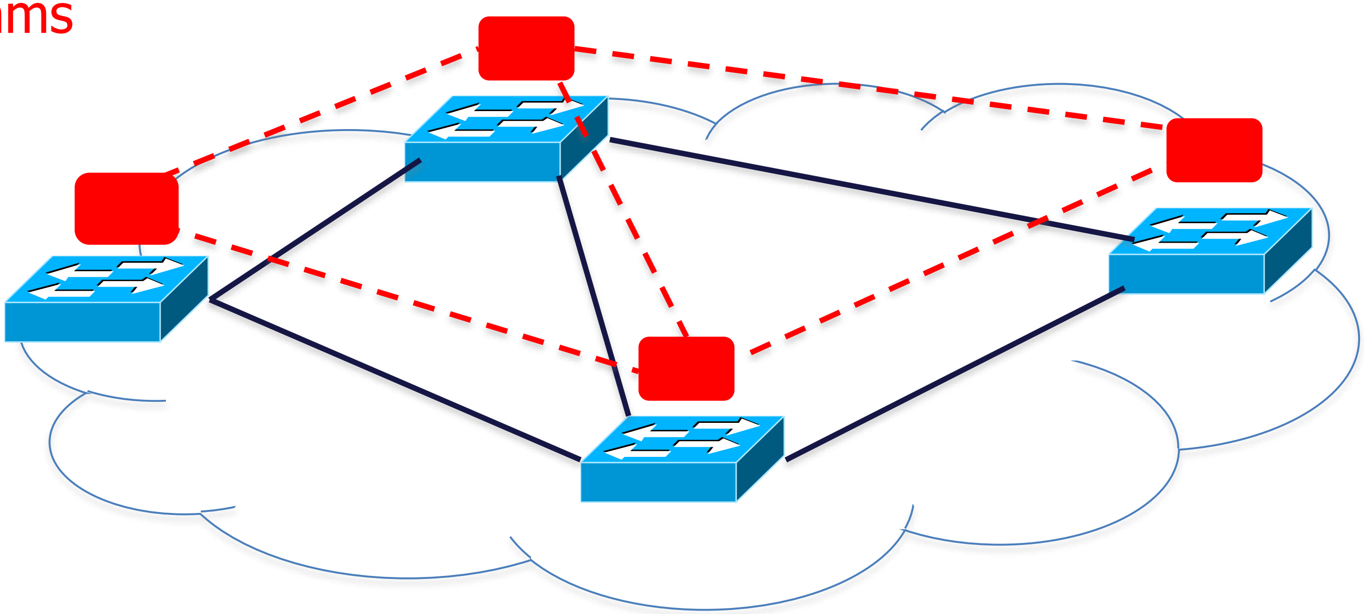
Data plane:
Packet streaming



Forward, filter, buffer, mark, rate-limit, and measure packets

Current Network's Problem Statements

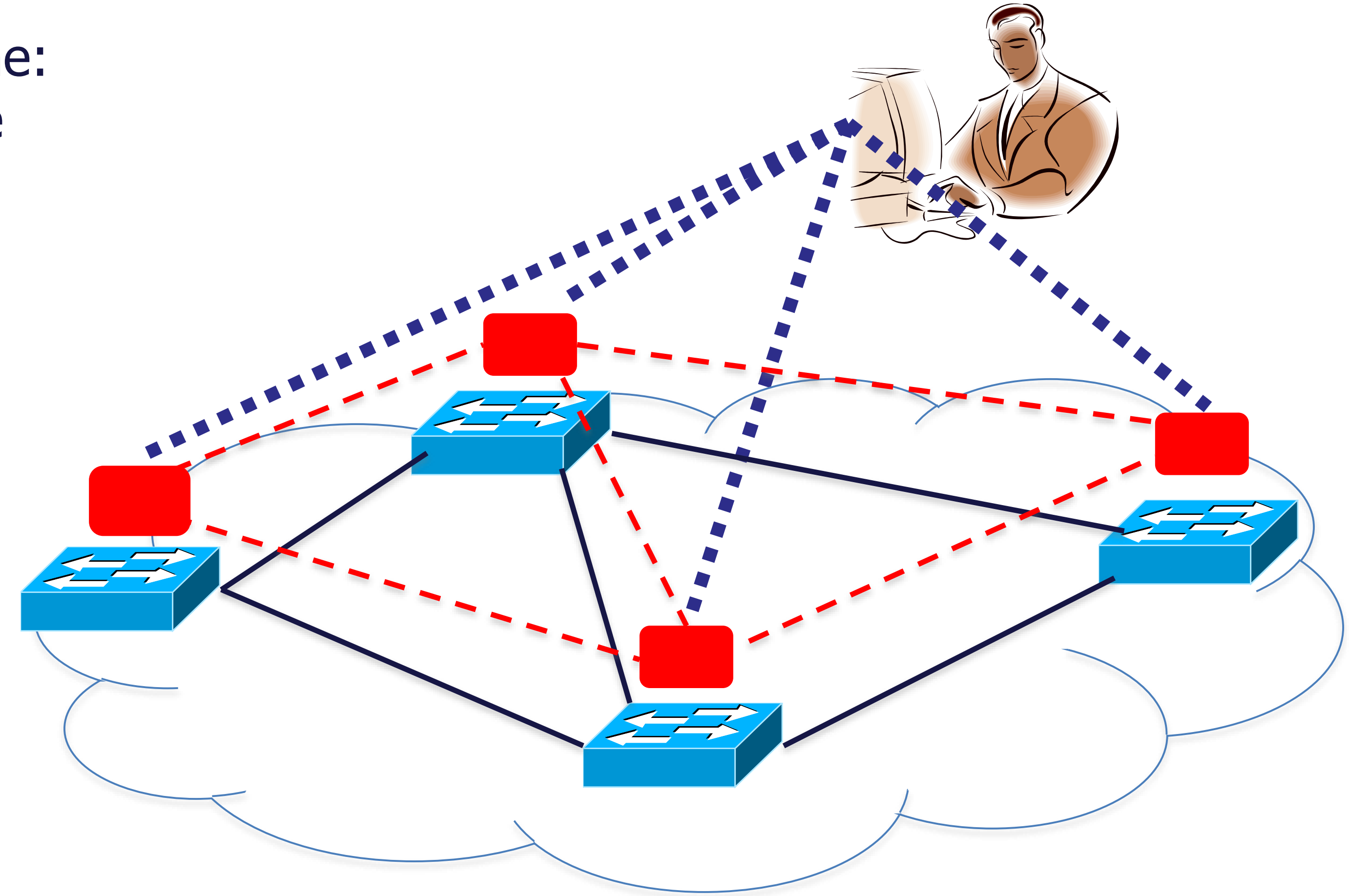
Control plane:
Distributed algorithms



Track topology changes, compute routes, install forwarding rules

Current Network's Problem Statements

Management plane:
Human time scale



Collect measurements and configure the equipment

Current Network's Problem Statements

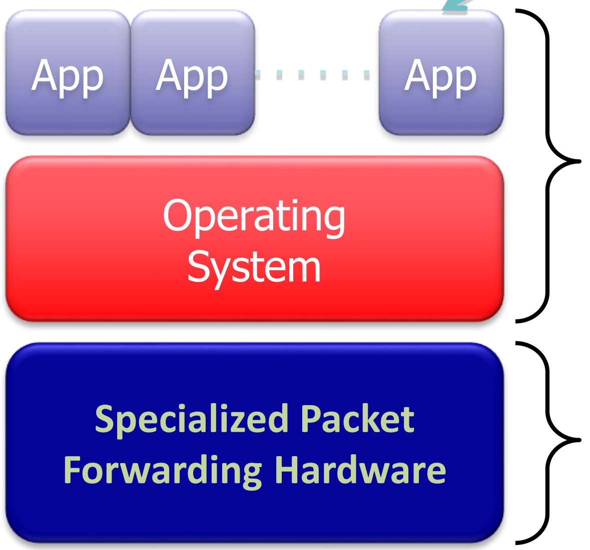
Time scales

	Data	Control	Management
Time scales	Packets	Events	Humans
Task	Forwarding/buffering /filtering/scheduling	Routing, circuit set-up	Analysis, configuration
Location	Hardware • Specialized hardware • Processes at line rate. • Every packet • Very fast	Router software • Uses CPU • Can only process a small number of packets • Very slow	Human or perl scripts

Current Network's Problem Statements

- **Conventional networking industry (2007)**
  - Many network functions baked into the infrastructure
    - *OSPF, BGP, multicast, differentiated services, Traffic Engineering, NAT, firewalls, MPLS, redundant layers, etc.*

Routing, management, mobility management, access control, VPNs, …

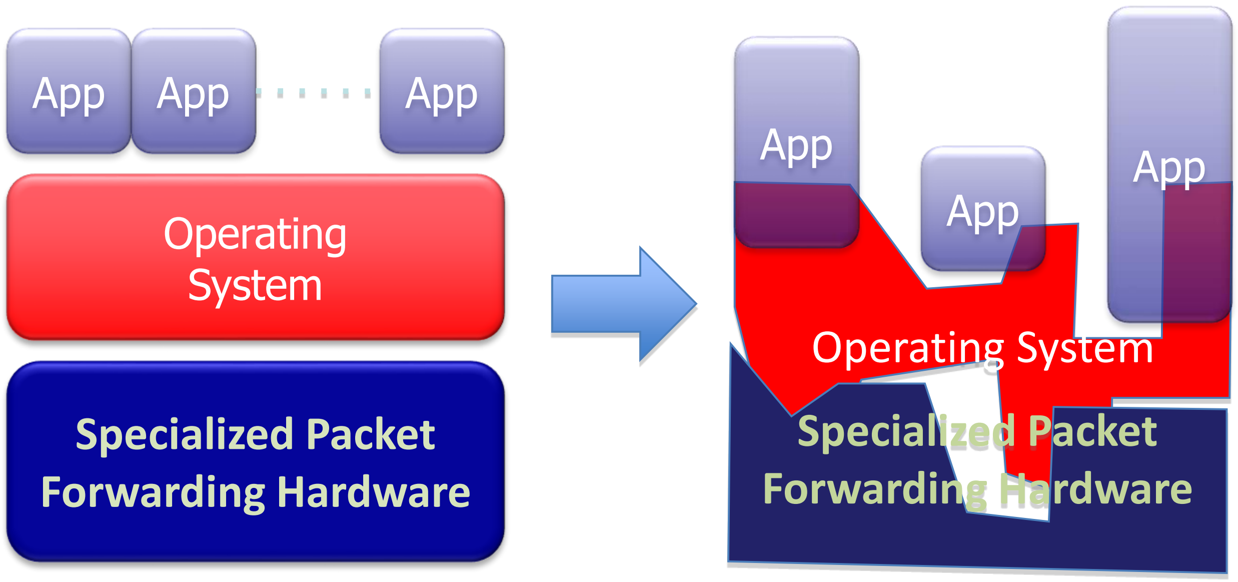


Million of lines of source code — Manage by 5400 RFCs

500M gates 10Gbytes RAM — Power Hungry

Current Network's Problem Statements

- ... and the reality (as in 2015)
  - Closed equipment
    - Software bundled with hardware
    - Vendor-specific interfaces
  - Over specified: slow protocol standardization
  - Hardly be innovated
    - Equipment vendors write the code
    - Long delays to introduce new features
  - Operating a network is expensive
    - More than half the cost of a network
    - Yet, operator error causes most outages
  - Buggy software in the equipment
    - Routers with 20+ million lines of codes
    - Cascading failures, vulnerabilities etc.

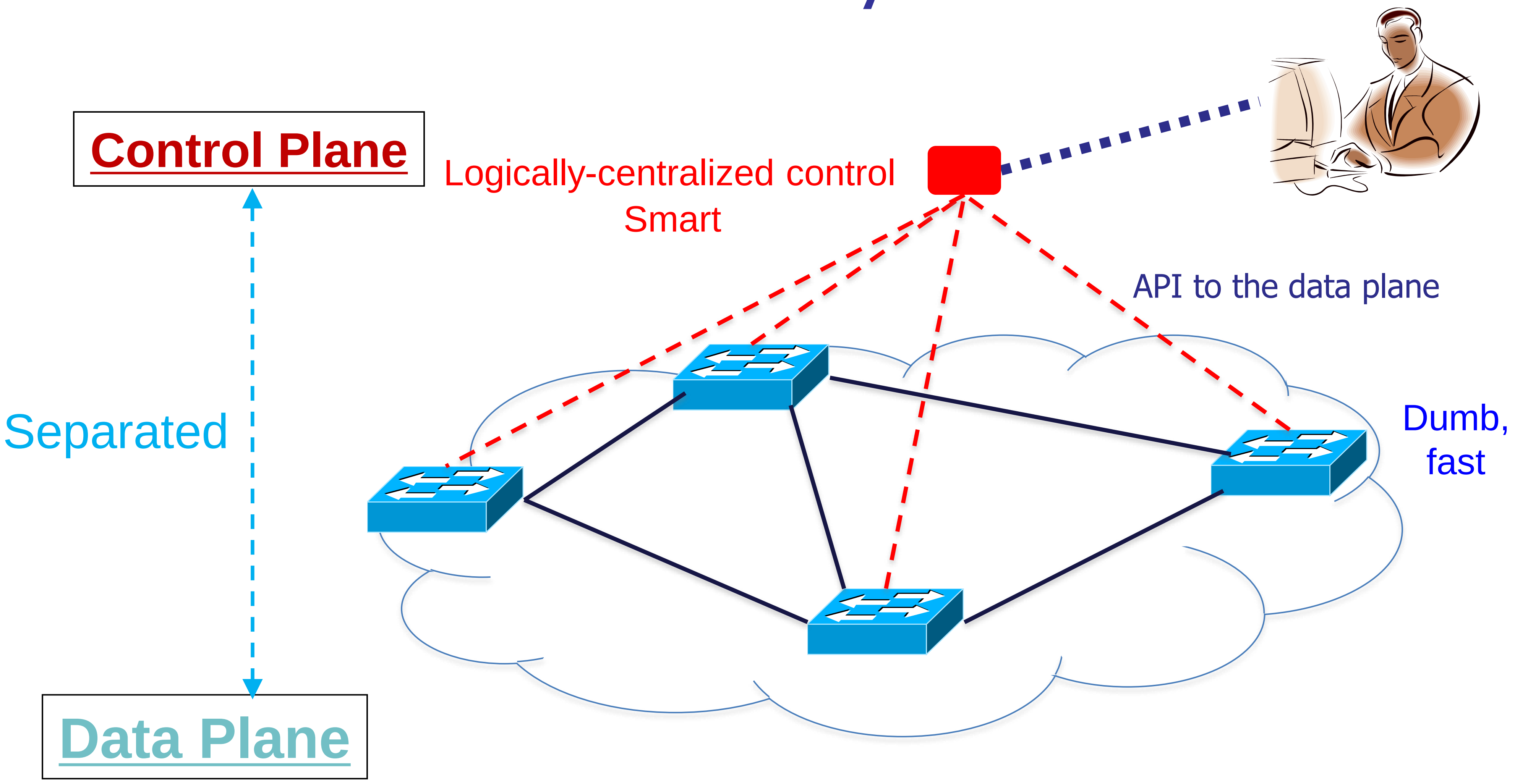


- Motivations
- SDN - Introduction
  - Why SDN?
  - What is SDN?
- Basic concepts

Why SDN?

- The network should be
  - Flexible
  - Agile
  - Programmable
- So that …
  - Networks are no longer closed, proprietary
    - Rapid deployment of new applications, services and infrastructure to meet new requirements
  - Networks should be an open and programmable component of the larger cloud infrastructure
    - Enabling innovation: creating new ICT business models
  - More control of infrastructure, allowing customization and optimization
  - Reducing costs
    - CAPEX
    - OPEX

Why SDN?


Control Plane
Logically-centralized control
Smart
API to the data plane
Separated
Dumb,
fast
Data Plane

What is SDN?

The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.

- ONF Definition of SDN -

1. Directly programmable
2. Agile: *Abstracting control from forwarding*
3. Centrally managed
4. Programmatically configured
5. Open standards-based vendor neutral

OPEN NETWORKING FOUNDATION

Control and Data Separation

- One important feature of SDN is *Abstracting control from forwarding*
  - → Separate control and data
- Concept of control and data separation
  - What is control/data plane?
  - Why control/data separation?
  - Opportunities and challenges

Control and Data Separation

- Control plane
  - Logic for controlling forwarding behaviors of network devices
  - Example
    - Routing
    - Network middlebox configurations: firewall etc.
- Data plane
  - Forwarding data traffic according to logics given by control plane
  - Example
    - IP forwarding
    - Layer-2 switching

Control and Data Separation

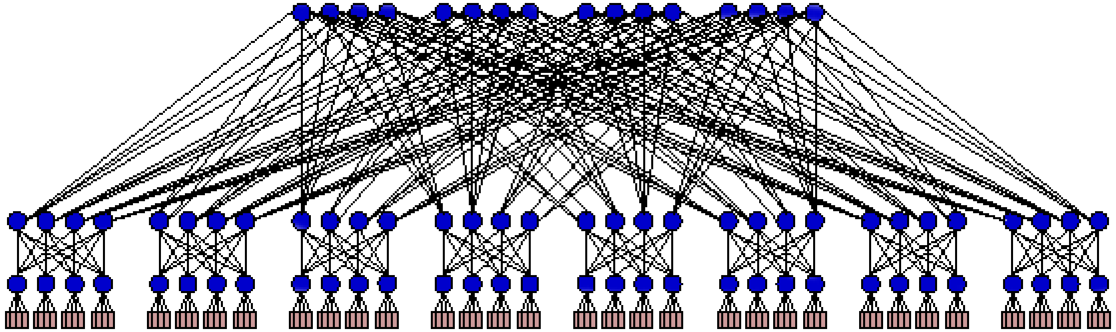
- Why separate control and data plane?
  - Independent evolution and development
    - Control logics can be more easily evolved, added independently of the underlying hardware
    - New solutions can be implemented without changing IP or end-hosts
  - Control from high-level software program
    - Control network behaviors using high-level programs
    - Debug/check behaviors more easily

Control and Data Separation

- Opportunities – where separation helps
  - Data centers: VM migration, layer-2 routing
  - Routing: more control over decision logics
  - Research networks: coexistence with production networks
  - Corporate networks: policy-based networking and security
- Examples
  - Data centers
  - Inter-domain routing

Control and Data Separation

- Data center networking (Yahoo!)
  - 20.000 servers/cluster ~ 400.000 VMs
    - Any-to-any, 1024 distinct inter-host links
    - Sub-second migration, guaranteed consistency
  - Biggest DCs
    - Google: 900.000 servers in 13 data centers
    - Amazon: 450.000 servers in 7 locations
    - Microsoft: 100.000 servers



Common Fattree DC network topology

Control and Data Separation

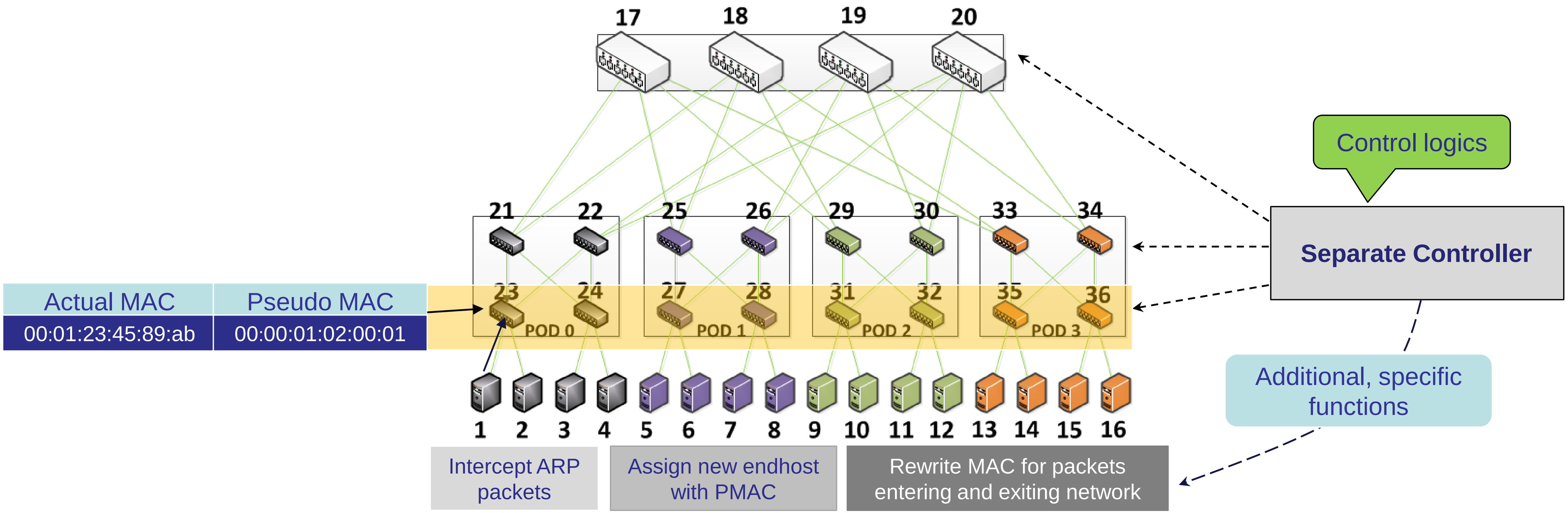
- Data center networking
  - Problem
    - ◇ Scalability
      - – Keeping 20k devices in sync with 400k entities?
      - – Routing in large network with complex topology
      - – Management
      - → How to route data in such complex topology?
    - ◇ Energy
      - – All machines in a DC should be available 24 hours per day and 7 days per week
      - – Average power consumption of a high performance server ~ 400W; average power consumption of a switch ~ 50W → total power consumption of a 20k-server DC: 8MW
      - → How to reduce energy-consumption to reduce OPEX and towards more environment-friendly?
    - ◇ Cost
      - – 20.000 servers → 2.500 switches in DC network
      - – $5k/vendor high-performance switch ~ $12,5M
      - – $1k/commodity switch ~ $2,5M
      - → How to save CAPEX by using commodity switches?
        - » *Savings in 10 data centers = $100M*

Control and Data Separation

- Data center networking
  - Requirements
    - More flexible control
    - Tailor network for services
    - Quickly improve and innovate
  - Solution – re-addressing hosts in DC towards more flexible routing
    - Layer-2:
      - Less configuration/administration
      - Bad scaling properties
        - MAC-based broadcasting
    - Layer-3:
      - Can use existing routing protocols but with high administration overhead due to complex network topology
    - → How to combine the advantages of these two schemes?

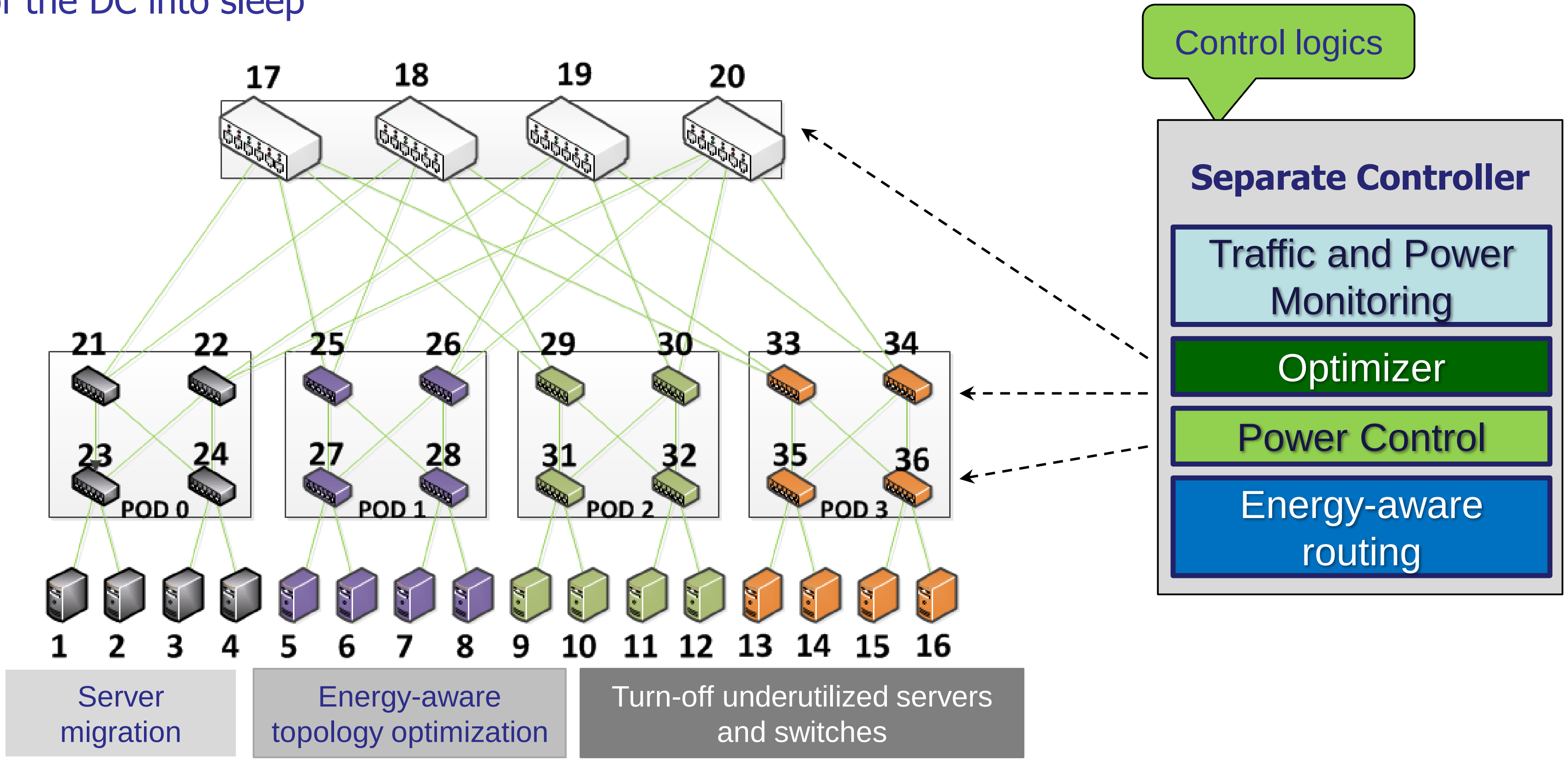
Control and Data Separation

- Data center networking
  - Re-addressing layer-2 MAC addresses for simple layer-2 host-to-host forwarding instead of layer-2 broadcasting/layer-3 routing [114]



Control and Data Separation

- Data center networking
  - Solution – energy-efficient data centers [115]
    - Aggregate and migrate VMs to a part of data center
    - Put the under-utilized part of the DC into sleep
    - → Reduce energy consumption



Control and Data Separation

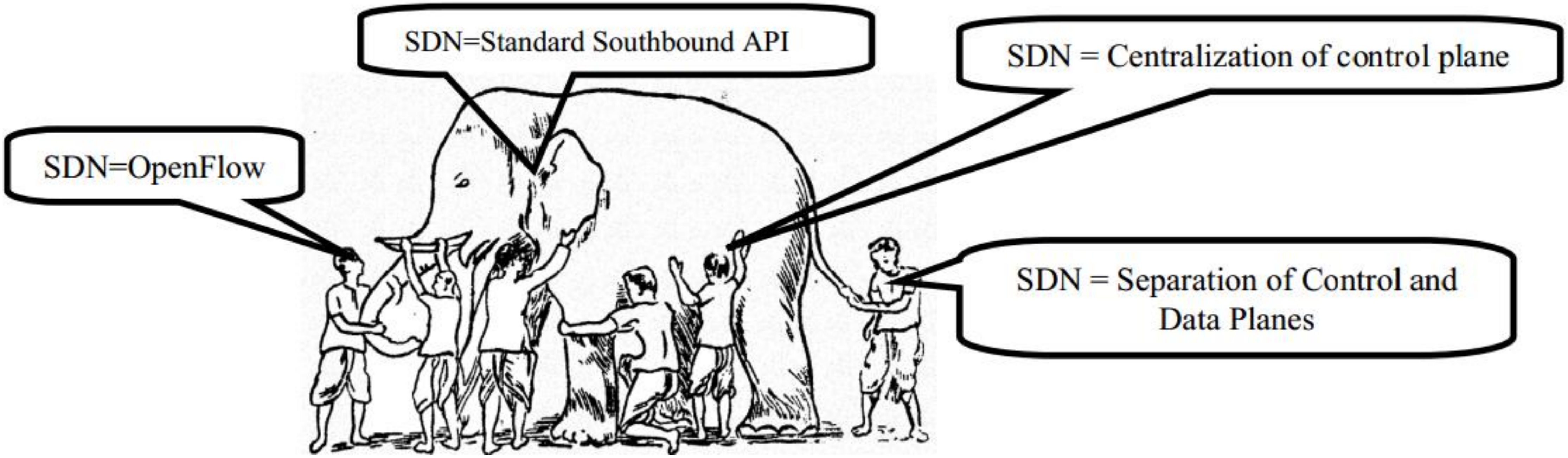
- Conclusions
  - A separate control plane from data forwarding is beneficial
    - New functions can easily be developed and deployed
    - More flexible
  - How the control plane is implemented in SDN?
  - What are the technical challenges?

    → *... next section to come!*

- Motivations
- SDN - Introduction
  - Why SDN?
  - What is SDN?
- Basic concepts

Software-Defined Networking

- SDN Architecture
  - Control Plane
  - Data Plane
  - Centralized Management
- SDN Principles
- SDN Components
  - SDN Controllers
  - SDN Forwarding Devices



What is SDN?
(Software-Defined Networking)

SDN Basic Concepts

SDN as a network architecture with four pillars

1. The control and data planes are decoupled
   - ❑ Network intelligence and states are logically centralized
   - ❑ Data plane uses commodity servers and switches
2. Forwarding decisions are flow based, instead of destination based
3. Control logic is moved to an external entity, the so-called SDN controller or NOS (software platform)
4. The network is programmable through software applications running on top of the NOS that interacts with the underlying data plane devices

SDN in Real World – Google's Story

- The industries were skeptical whether SDN was possible.
- Google had big problems:
  - High financial cost managing their datacenters: Hardware and software upgrade, over provisioning (fault tolerant), manage large backup traffic, time to manage individual switch, and a lot of men power to manage the infrastructure.
  - Delay caused by rebuilding connections after link failure.
    - Slow to rebuild the routing tables after link failure.
    - Difficult to predict what the new network may perform.
- Google went a head and implemented SDN.
  - Built their hardware and wrote their own software for their internal datacenters.
  - Surprised the industries when Google announced SDN was possible in production

Reference: "*B4: Experience with a Globally-Deployed Software Defined WAN*", ACM Sigcomm 2013

A Short History of SDN

- 2004: Research on new management paradigms
  - RCP, 4D [Princeton, CMU,….]
  - SANE, Ethane [Stanford/Berkeley]
- 2006: Martin Casado and team (Stanford) proposed a clean-slate security architecture (SANE) which defines a centralized control of security. Ethane generalizes it to all access policies.
- 2008: Software-Defined Networking (SDN)
  - NOX Network Operating System (Nicira) [8]
  - OpenFlow switch interface from OpenFlow project (Stanford/Nicira) (ACM SIGCOMM 2008 [38])
- 2009:
  - Stanford publishes OpenFlow V1.0.0 specs
  - Martin Casado co-founds Nicira

Martin Casado

A Short History of SDN

- 2011: Open Networking Foundation (~69 members)
  - Board: Google, Yahoo, Verizon, DT, Microsoft, Facebook, NTT
  - Members: Cisco, Juniper, HP, Dell, Broadcom, IBM,…..
  - Oct 2011: First Open Networking Summit. Many Industries (Juniper, Cisco) announced to incorporate
- 2012: VMware buys Nicira for $1.26B
- 2013: Latest Open Networking Summit
  - 1600 attendees, Google: SDN used for their WAN
  - Commercialized, in production use (few places)

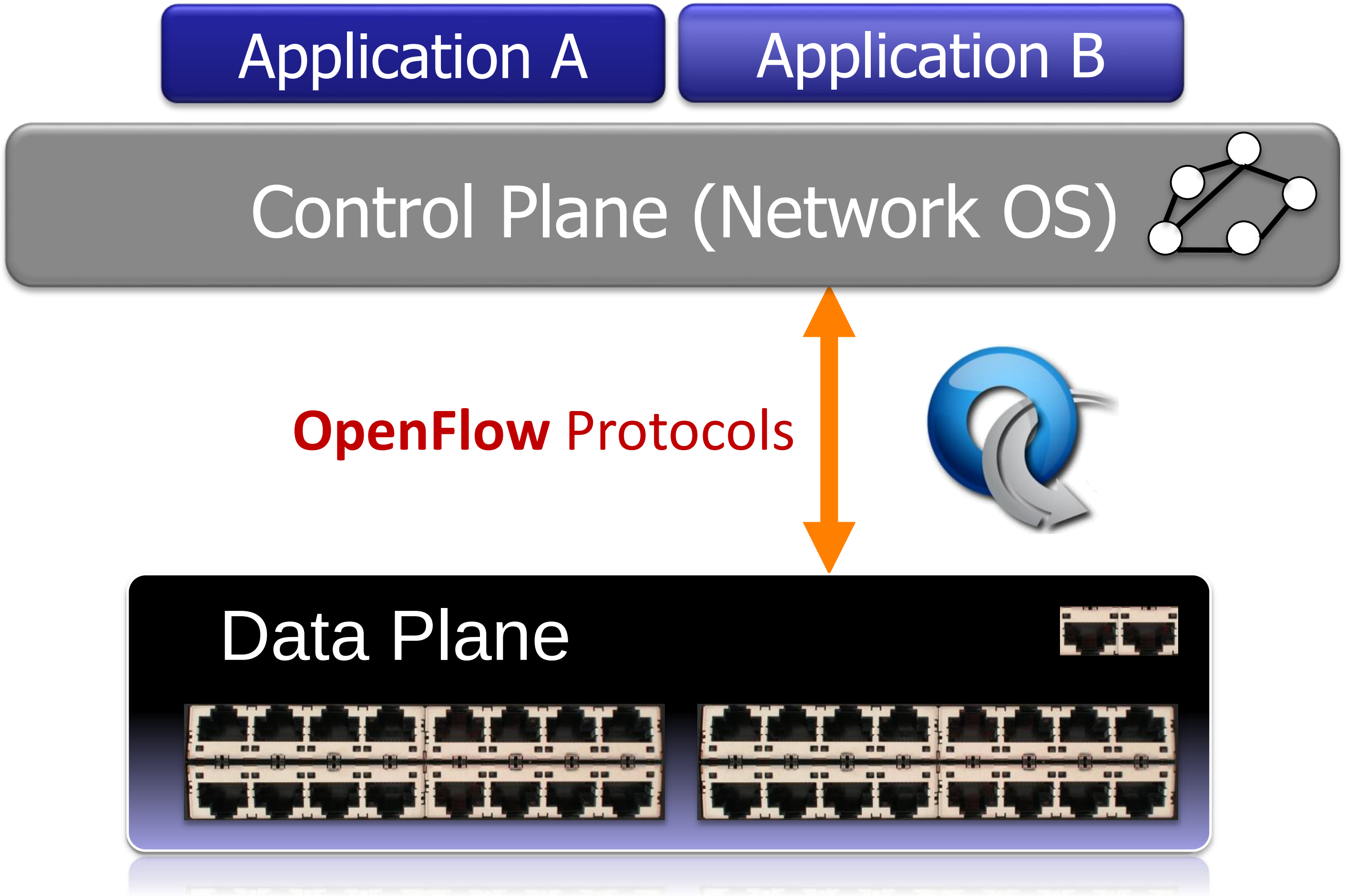
→ Lesson learned: Imagination is the key to *unlock* the power of possibilities

OPEN NETWORKING FOUNDATION

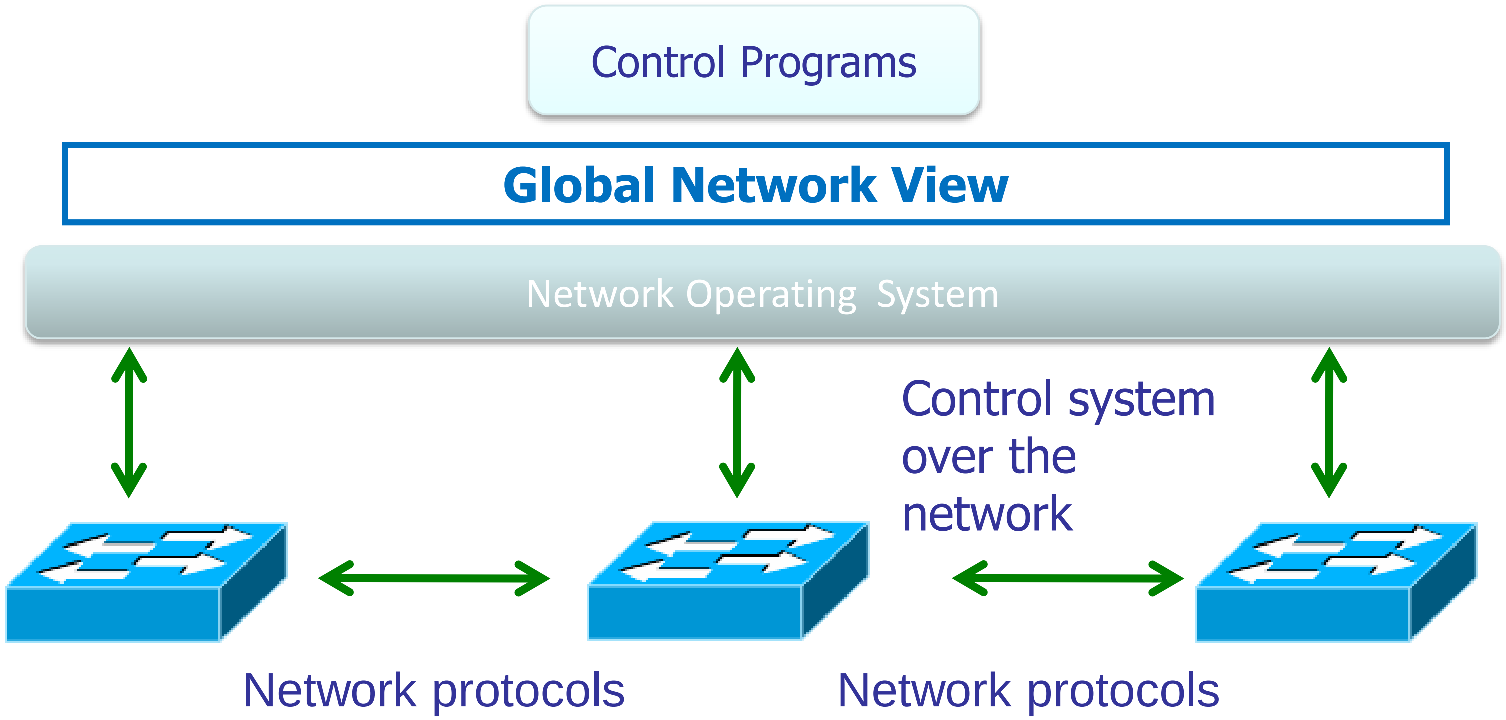
Open Flow and SDN

- SDN and OpenFlow are different
  - SDN is a concept of the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.
  - OpenFlow is communication interface between the control and data plane of an SDN architecture
    - Allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual.
    - Think of as a protocol used in switching devices and controllers interface.

OpenFlow

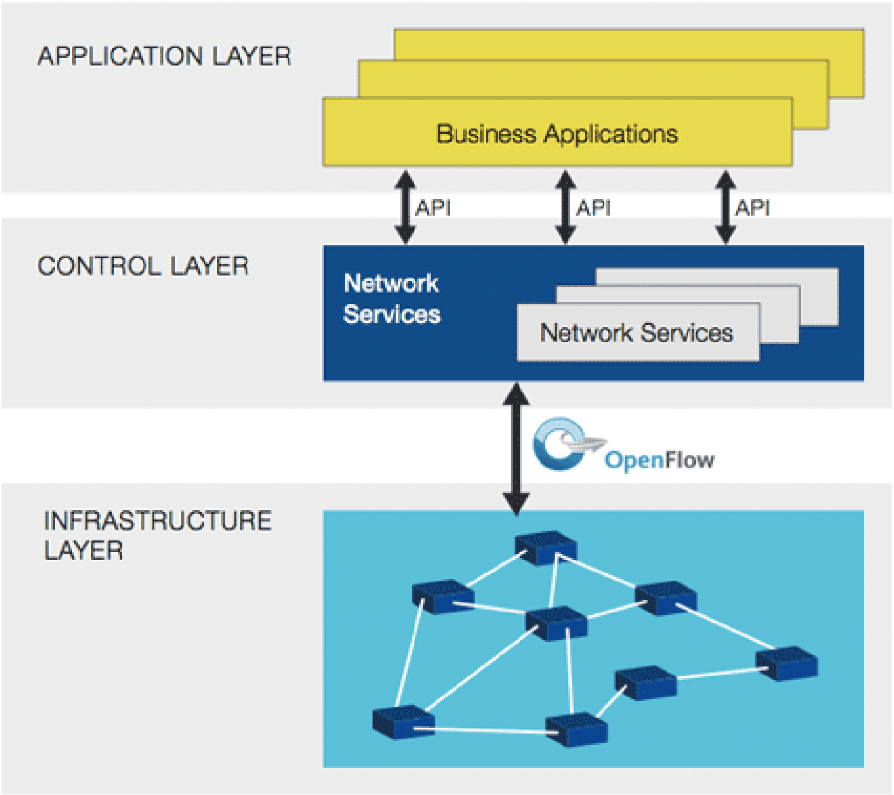

Application A
Application B
Control Plane (Network OS)
OpenFlow Protocols
Data Plane

SDN Introduction



SDN Architecture

Simplified view of an SDN architecture.

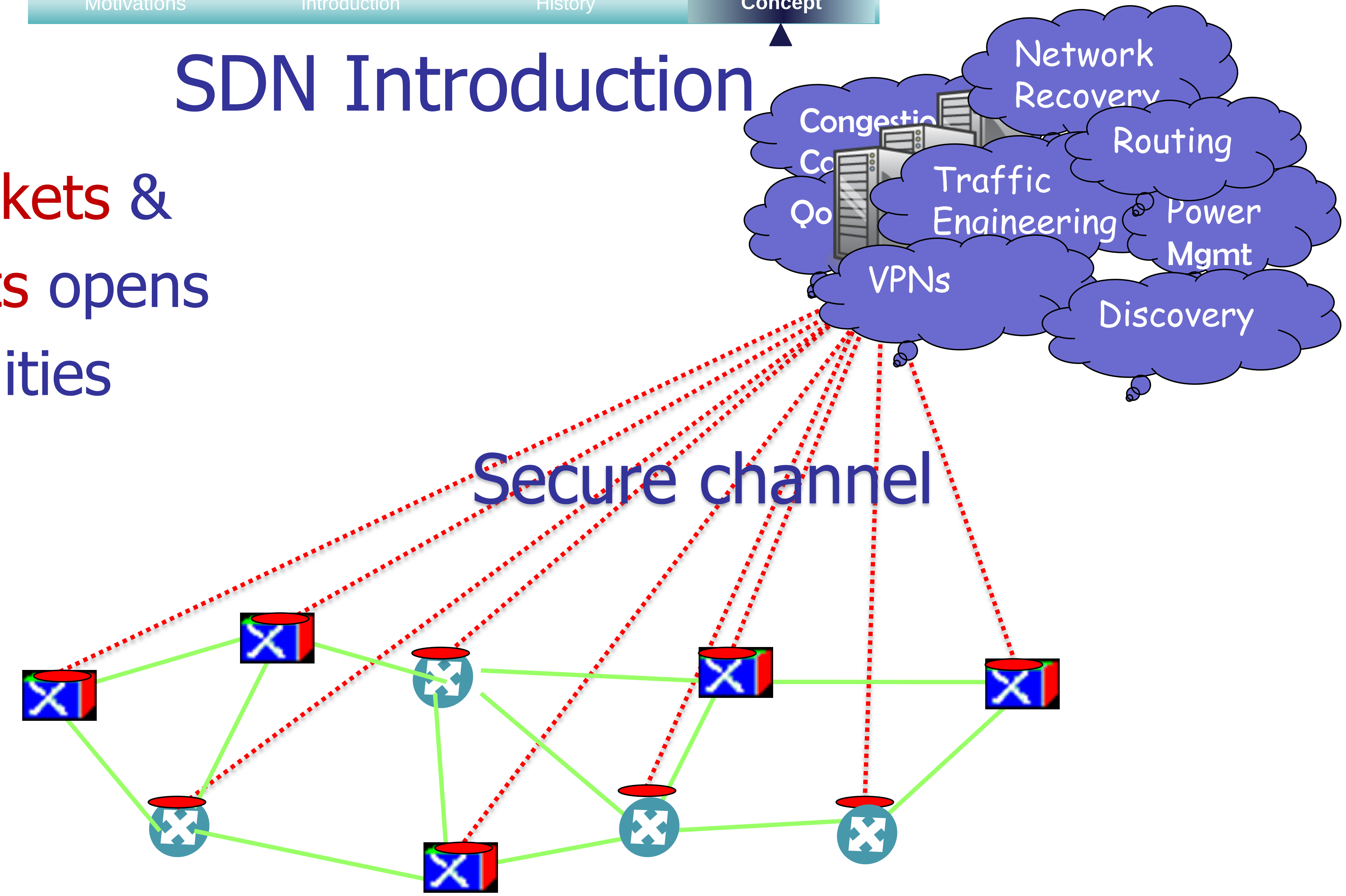


- Centralized management and control
- Control and data planes are decoupled

Ref: https://www.opennetworking.org/sdn-resources/sdn-definition

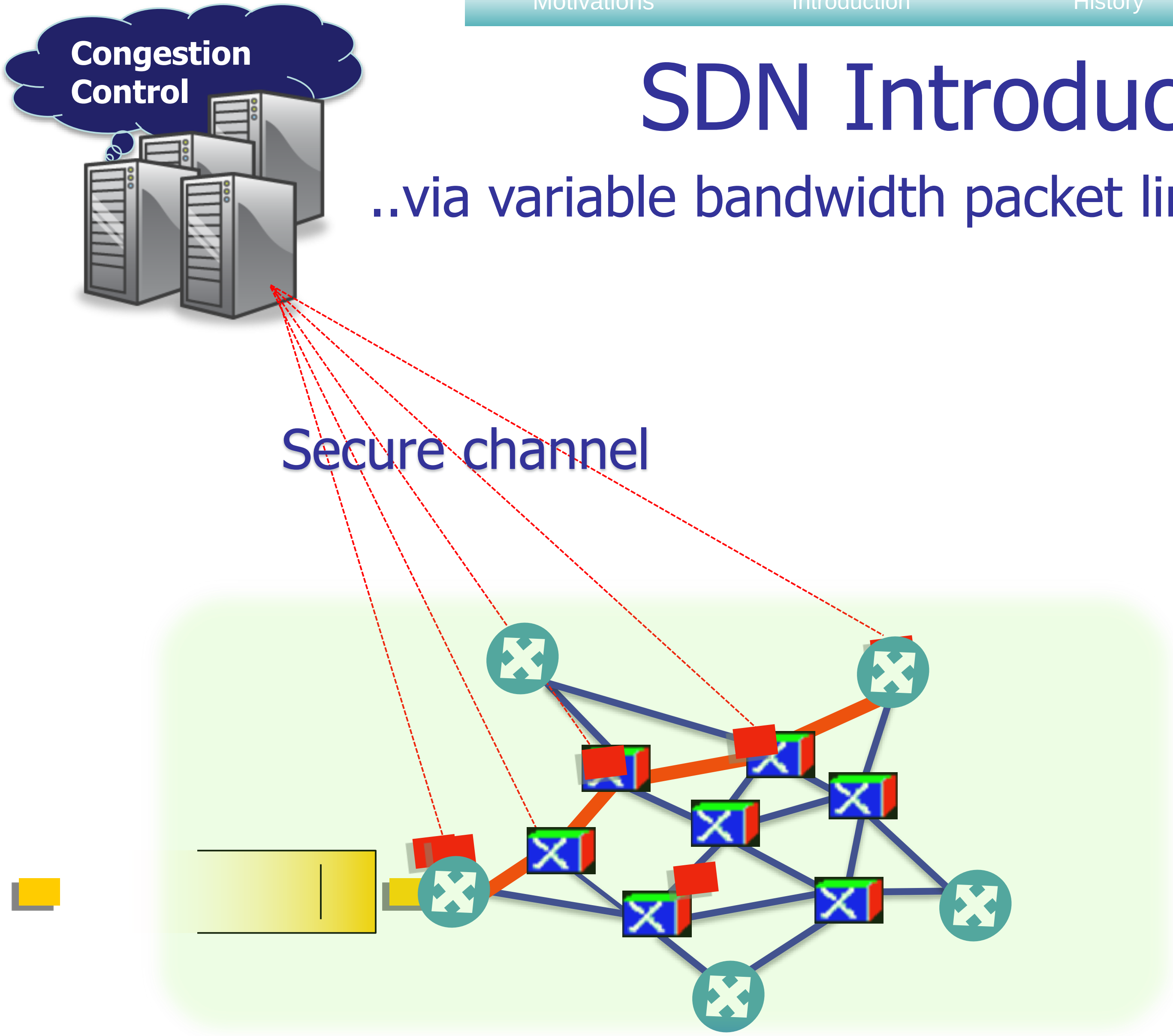
SDN Introduction

Converged packets & dynamic circuits opens up new capabilities



SDN Introduction

..via variable bandwidth packet links

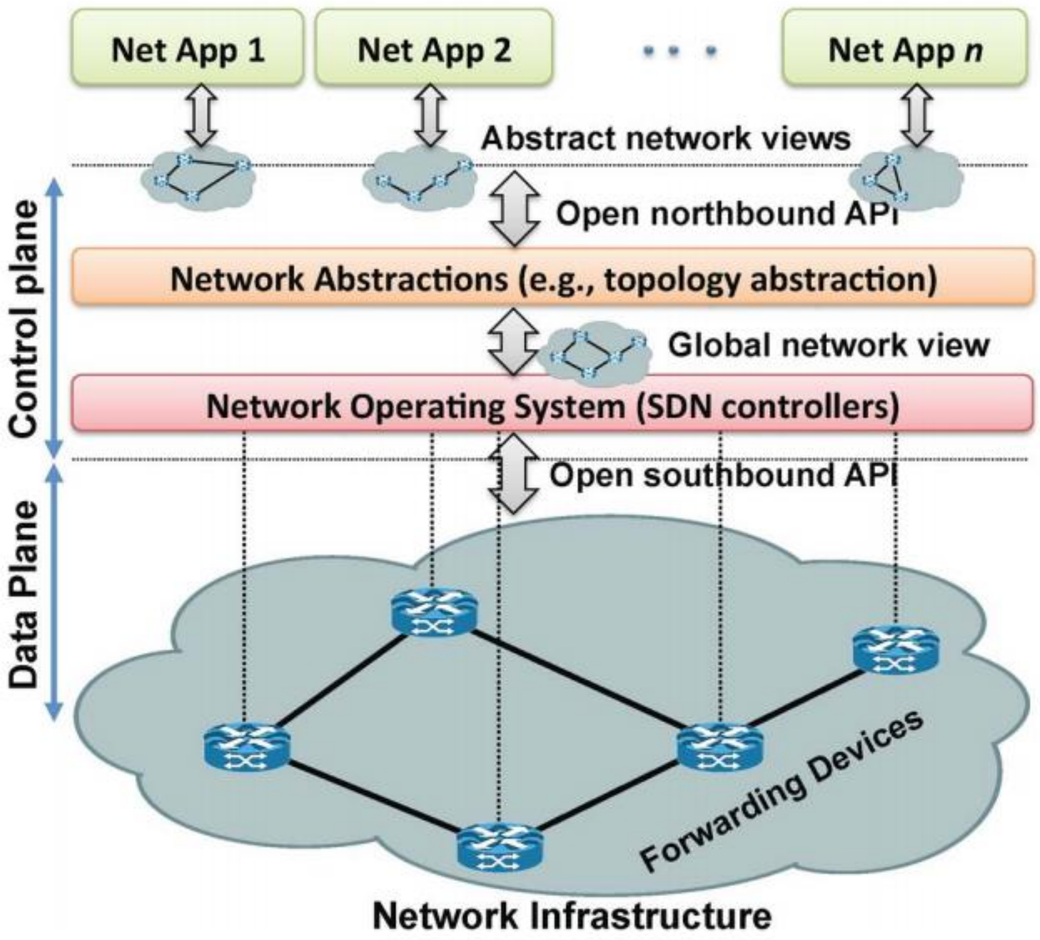


- Example:
  - Resource reservation for congestion control

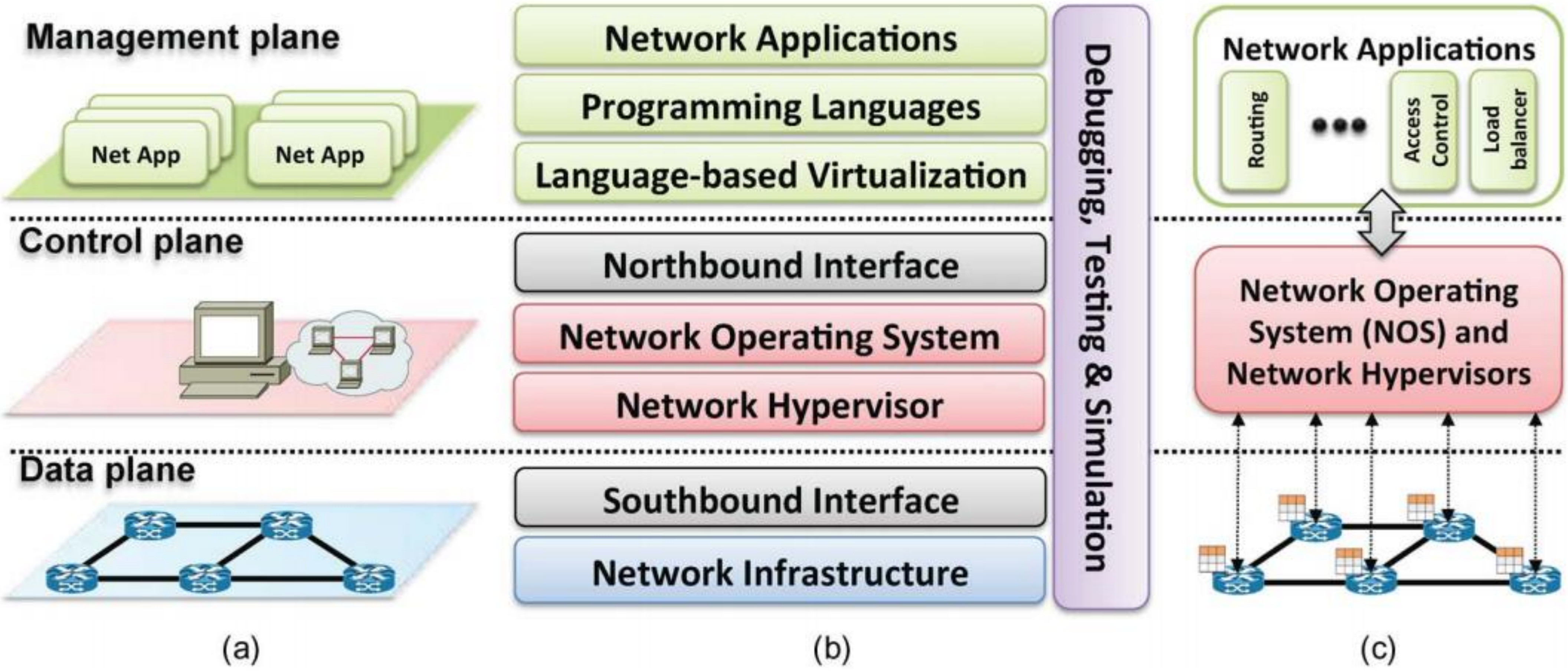
SDN Terminology [1]

1. Forwarding Devices (FD)
2. Data Plane (DP)
3. Southbound Interface (SI)
4. Control Plane (CP)
5. Northbound Interface (NI)
6. Management Plane (NP)

SDN architecture and its fundamental abstractions



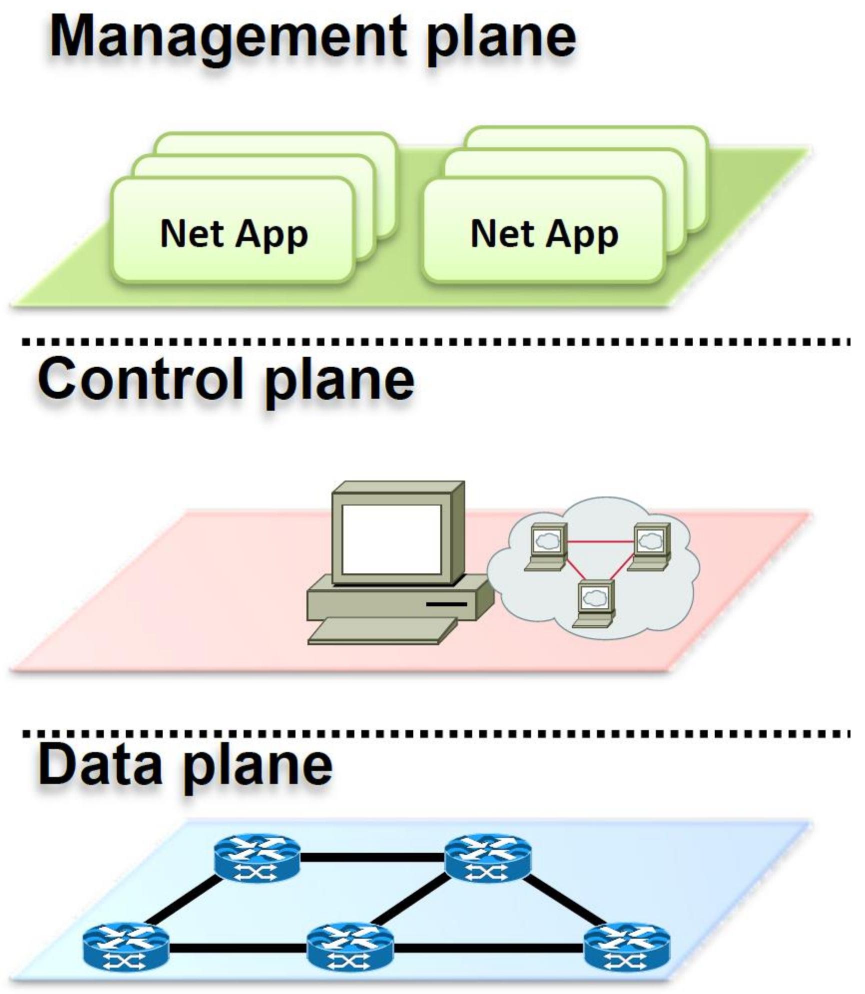
SDN Architecture – in Planes, Layers and System Design



The SDN layers are represented in Fig (b), as explained above. Fig. (a) and (c) depicts a plane-oriented view and a system design perspective, respectively [1]

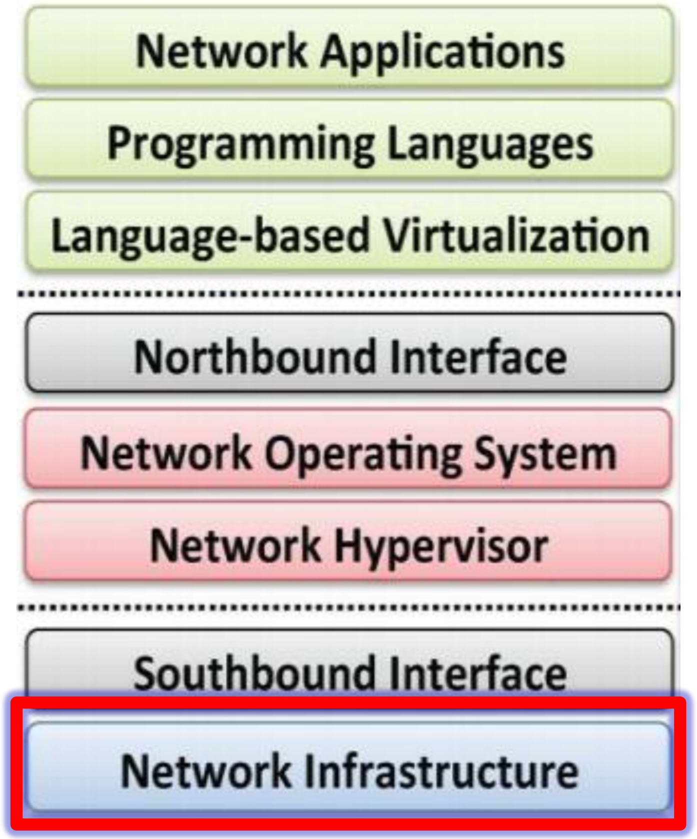
SDN Architecture – in Planes, Layers and System Design

- Data plane
  - Network devices and infrastructure
  - Data forwarding
- Control plane
  - "Network's brain", including control logics that control network devices
  - Control network devices by well-defined protocols on southbound interface
- Management plane
  - Set of applications based on functions offered by northbound interface
  - Management applications define the policies, which are ultimately translated to southbound-specific instructions that program the behavior of the forwarding devices
  - Example: routing, firewall, load balancing, congestion control, monitoring etc.



SDN Architecture – Network Infrastructure

- Set of equipments: switches, routers, middlebox appliances
- Network intelligence is removed to logically centralized Network Operating System
- Built with open, standard interfaces (e.g., OpenFlow)
- Flow tables/pipeline of flow tables are the essential part, in OpenFlow:
  - Matching rule
  - Actions to be executed on matching packets
  - Counters that keep statistics on matching packets



SDN Architecture – Network Infrastructure

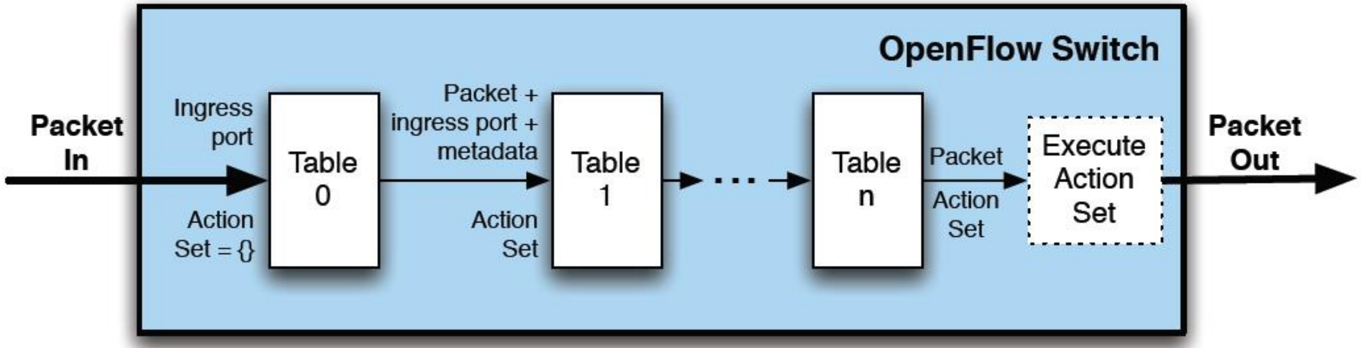
■ Flow table structure (OpenFlow)

Rule	Action	Counter

Packet + byte counters

- Forward packets to zero or more ports
- Encapsulate and forward to controller
- Send to normal processing pipline
- Modify fields
- *Any extentions you want to add!*

Switch Port	VLAN ID	VLAN pcp	MAC src	MAC dst	Eth type	IP src	IP dst	IP ToS	IP Prot	L4 s. port	L4 d. port



SDN Architecture – Network Infrastructure

- SDN/OpenFlow-enabled devices
  - Commercialized vs. open source solutions
  - Hardware vs. software
- Commercialized hardware products

BROCADE ciena hp JUNIPER NETWORKS

DASAN Networks NEC NETGEAR ADVA Optical Networking big switch networks

SDN Architecture – Network Infrastructure

- Open source soft OF routers
  - Open VSwitch
    - http://www.openvswitch.org/
    - based on Nicira's concept
    - a production quality open source software switch designed to be used as a virtual switch in virtualized server environments;
    - integrated into many cloud management systems including OpenStack, openQRM, OpenNebula, and oVirt
    - Open vSwitch as designed to be compatible with modern switching chipsets
  - Indigo:
    - Open source implementation that runs on Mac OS X, a part of the Project Floodlight
    - https://floodlight.atlassian.net/wiki/spaces/indigo2
  - Pantou:
    - Turns a commercial wireless router/access point to an OpenFlow enabled switch.
    - Supports generic Broadcom and some models of LinkSys and TP-Link access points with Broadcom and Atheros chipsets
    - http://archive.openflow.org/wk/index.php/OpenFlow_1.0_for_OpenWRT

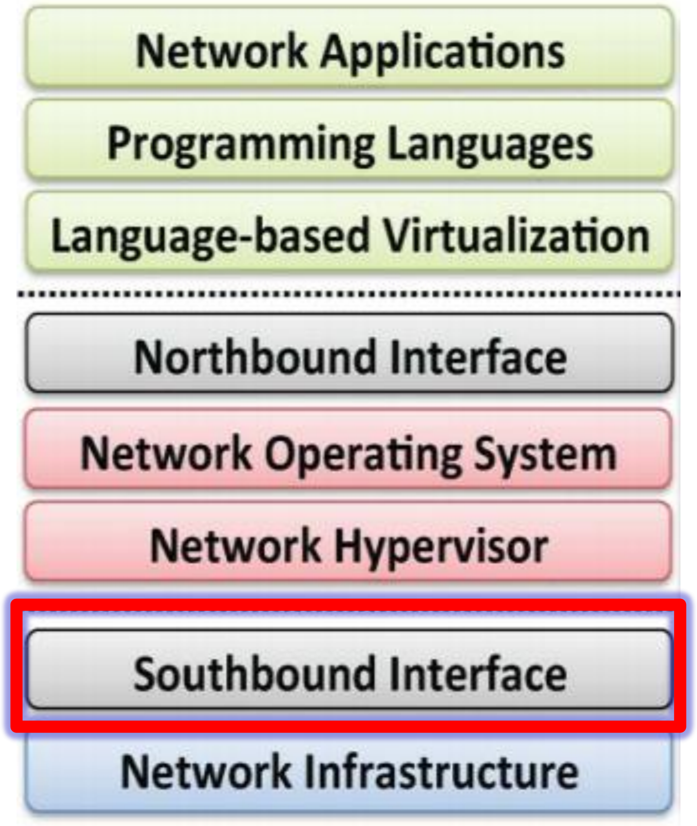
SDN Architecture – Network Infrastructure

- Open source hardware OF router
  - OpenFlow NetFPGA reference switch
    - http://NetFPGA.org
    - Offers OF switch implementations
    - FPGA chip from XiLink
    - 4 x 1Gbps, 10Gbps ports
    - Hardware router can be modified to add more functionalities (e.g. energy-efficiency, proxy, firewall etc.)

NetFPGA

SDN Architecture – Southbound Interface

- Southbound interfaces (or southbound APIs) are the connecting bridges between control and forwarding elements
  - OpenFlow is the most widely accepted and deployed open southbound standard.
    - Common specification to implement OpenFlow-enabled forwarding devices
    - Communication channel between data and control plane devices
  - OpenFlow provides three information for NOSs
    - Event-based messages are sent by forwarding devices to the controller when a link or port change is triggered
    - Flow statistics are generated by the forwarding devices and collected by controller
    - Packet-in message

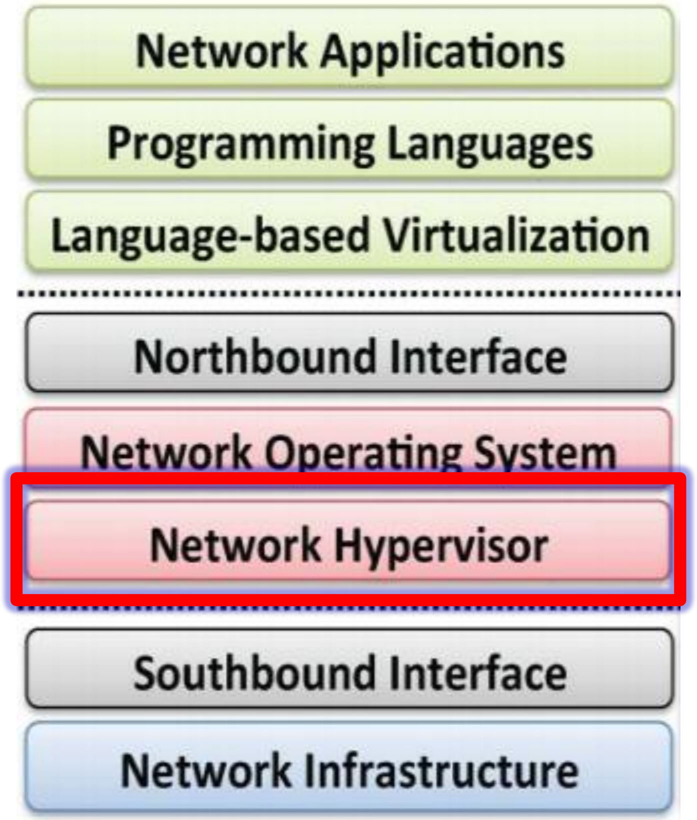


SDN Architecture – Southbound Interface

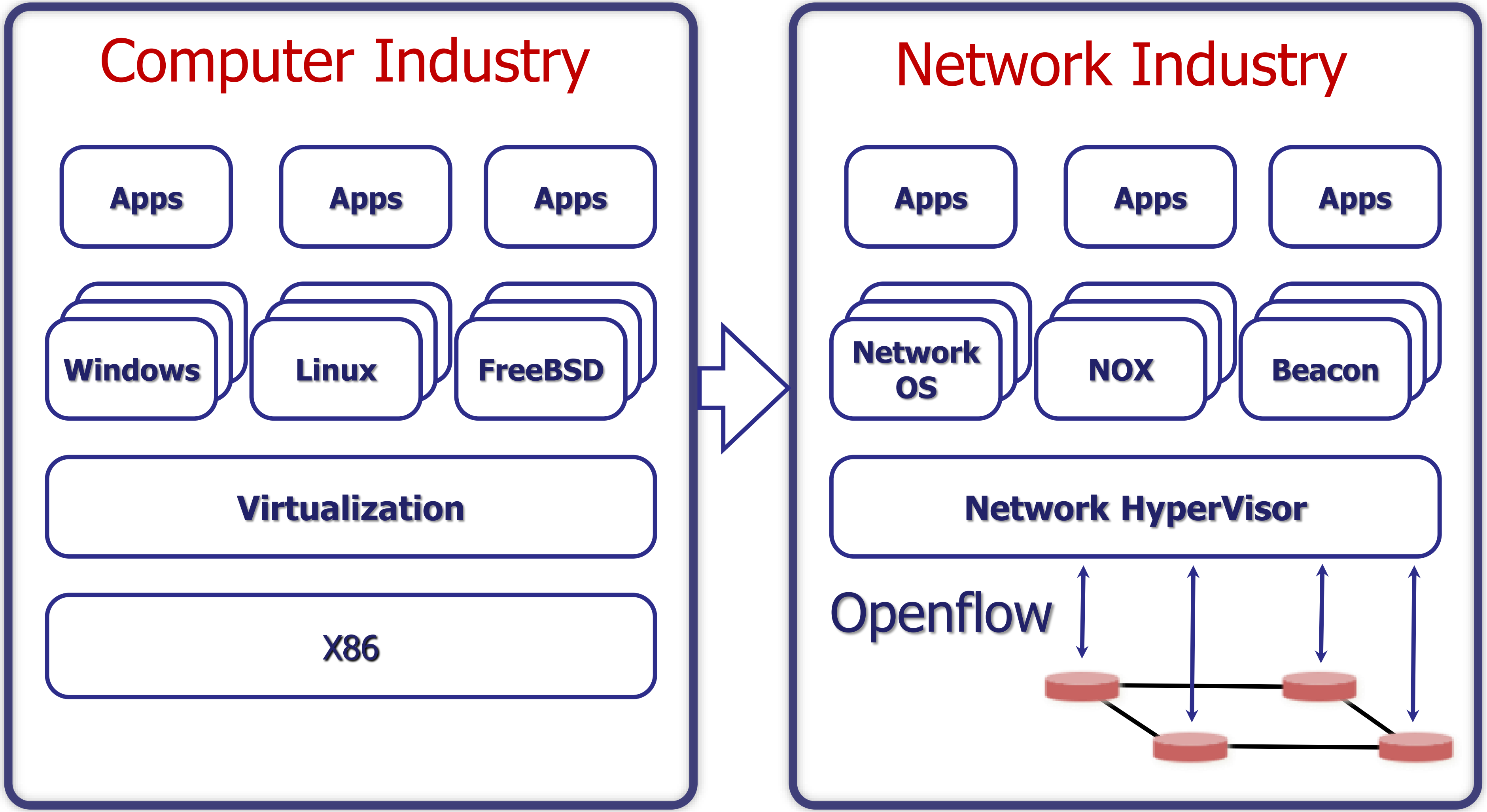
- OpenFlow is not the only available southbound interface for SDN. There are other API proposals such as:
  - ForCES [2] - a more flexible approach to traditional network management without changing the current architecture of the network, i.e., without the need of a logically centralized external controller
  - OVSDB [3] – provides advanced management capabilities for OpenVSwitches
  - OpFlex [4] distributes part of the complexity of managing the network back to the forwarding devices, with the aim of improving scalability
  - OpenState [5] and ROFL [6] propose extended finite machines (stateful programming abstractions) as an extension (superset) of the OpenFlow match/action abstraction

SDN Architecture – Network Hypervisors

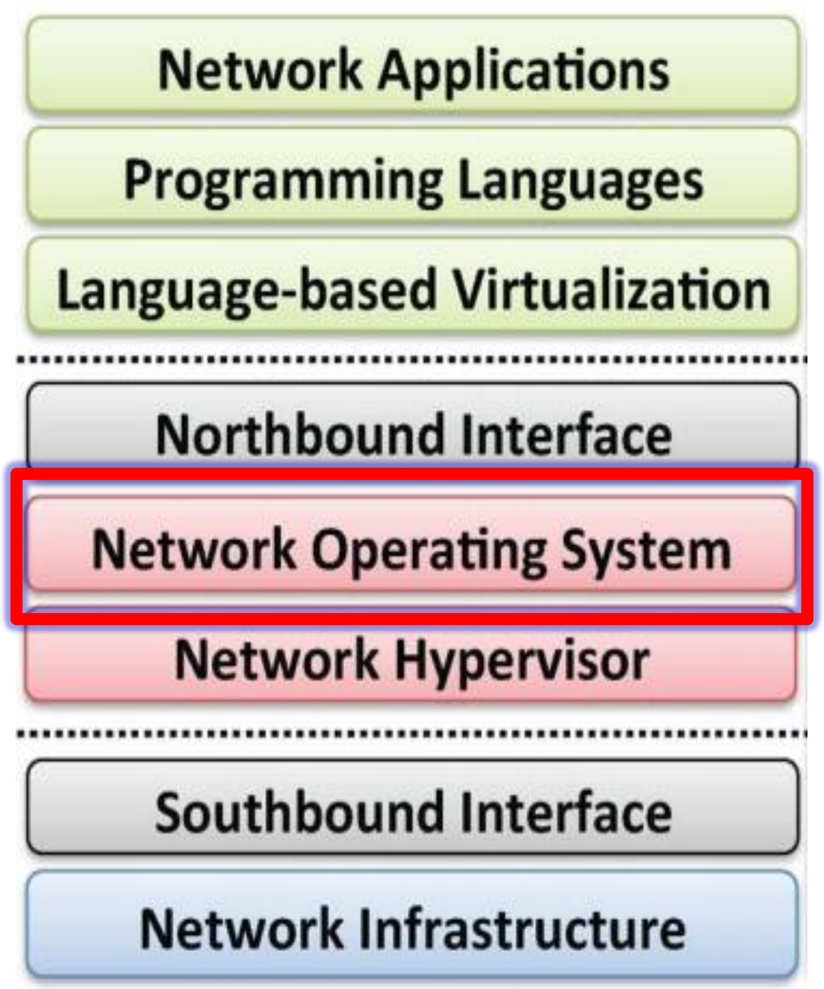
- Hiding the physical network infrastructure and providing an abstraction of physical network resources
  - Decouple virtual resources from physical resources (e.g., nodes, links)
- Creation and management of virtual networks
- Resource allocation and management
  - To create and manage resource allocated to each virtual network (e.g., physical nodes, physical allocated bandwidth)



SDN Architecture – Network Hypervisors



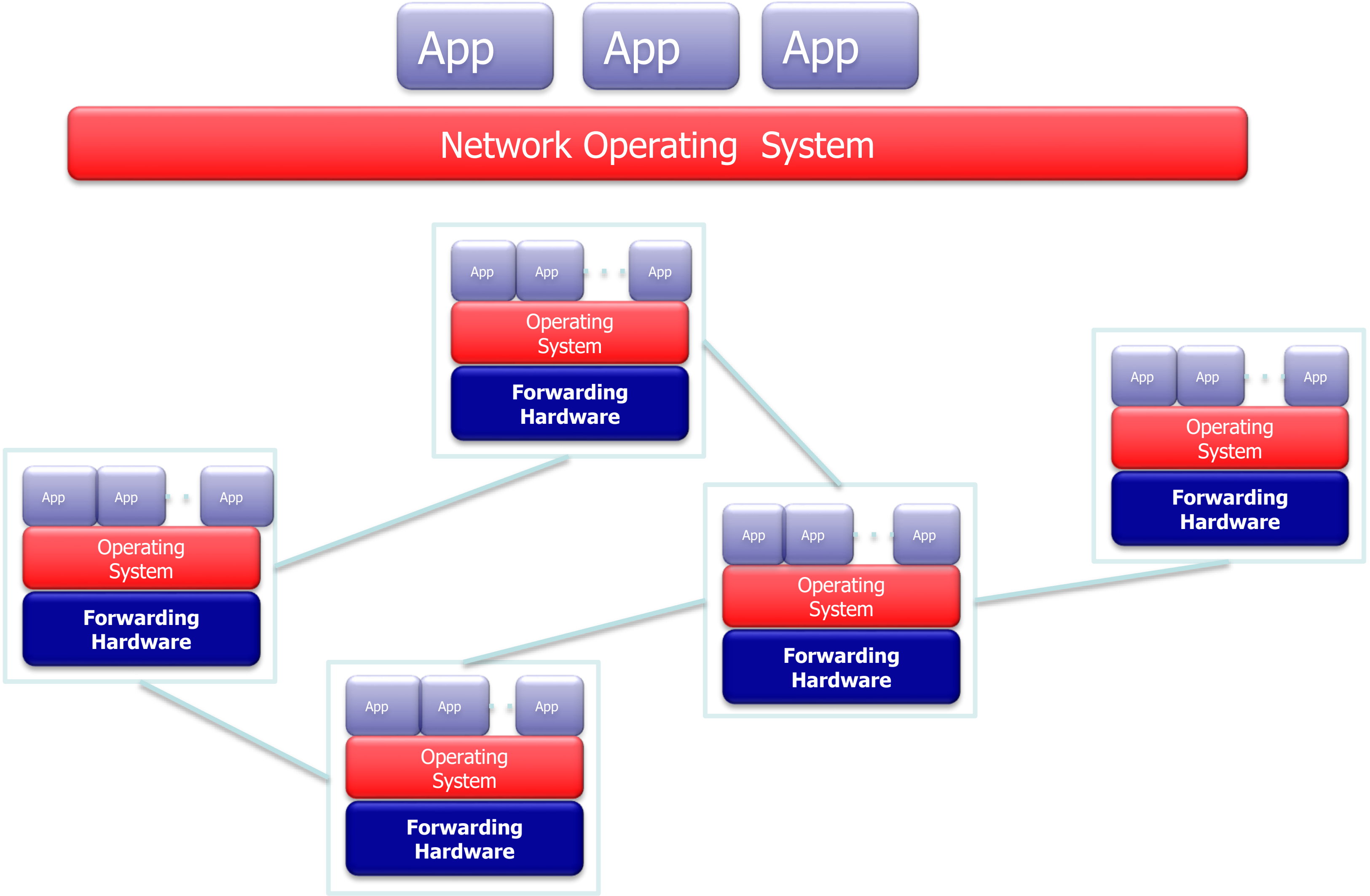
SDN Architecture - Network Operating System



- An SDN Network Operating System - (SDN controller) is a critical element
  - Key supporting piece for the control logic (applications) to generate the network configurations based on the policies defined by the network operator.
  - The most relevant architectures are centralized or distributed

SDN Architecture - Network Operating System

- Network Operating System (SDN Controller)
  - Global view: Controller collects information from all forwarding devices and make a global view
  - Centralized management: using the global view, a controller centralized and flexibly manages the network with its algorithm



SDN Architecture - Network Operating System

- A centralized controller is a single entity that manages all forwarding devices of the network.
  - NOX [8], POX [27], Maestro [9], Beacon [7], Floodlight [10] etc.
  - Easy to deploy
  - Single point of failure
- Distributed controllers: a centralized cluster of nodes or a physically distributed set of elements
  - Onix [11], HyperFlow [13], HP VAN SDN [14], ONOS [12], DISCO [15], yanc [16], PANE [17]
  - Redundancy
  - Comflexity

SDN Architecture - Network Operating System

- Core Controller Functions: essential network control functionalities such as:
  - Topology
  - Statistics and notifications
  - Device management, together with shortest path forwarding and security mechanisms
- Southbound: Most controllers support only OpenFlow as a southbound API. Still, a few of them offer a wider range of southbound APIs and/or protocol plug-ins (ODL, Onix,...)
  - Southbound API: OpenFlow, OVSDB, and ForCES
  - Protocol plug-ins to manage existing or new physical or virtual devices (e.g., SNMP, BGP, and NetConf)
  - OpenDaylight is going to allow several southboundAPIs and protocols to coexist in the control platform (to be discussed later…)
- Northbound: Current controllers offer a quite broad variety of northbound APIs, such as ad hoc APIs, RESTful APIs [35]

SDN Architecture - Network Operating System

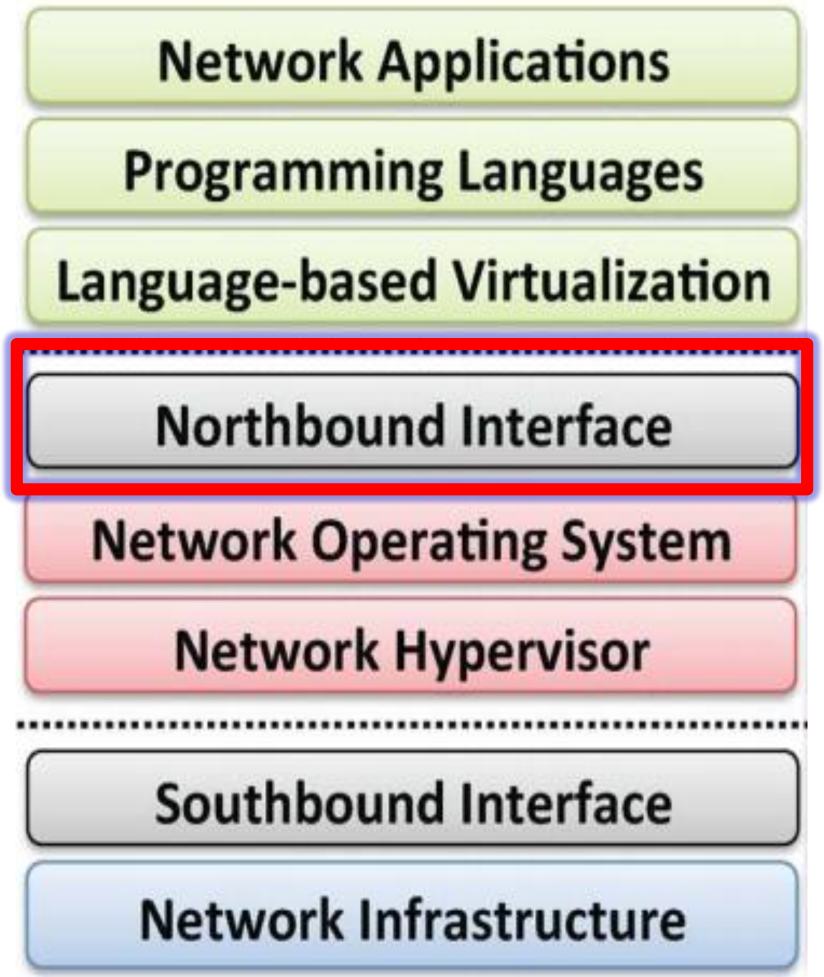
Controllers Classification

Name	Architecture	Northbound API	Consistency	Faults	License	Prog, language	Version
Beacon [7]	Centralized multi-threaded	ad-hoc API	No	No	GPLv2	Java	V1.0
DISCO [15]	Distributed	REST	---	Yes	---	Java	V1.1
ElastiCon [18]	Distributed	RESTful API	Yes	No	---	Java	V1.0
Fleet [19]	Distributed	ad-hoc	No	No	---	---	V1.0
Floodlight [10]	Centralized multi-threaded	RESTful API	No	No	Apache	Java	V1.1
HP VAN SDN [14]	Distributed	RESTful API	Weak	Yes	---	Java	V1.0
HyperFlow [13]	Distributed	---	Weak	Yes	---	C++	V1.0
Kamdoo [20]	Hierachically distributed	---	No	No	---	C, C++, Python	V1.0
Onix [11]	Distributed	NVP NBAPI	Weak, strong	Yes	Commercial	Python, C	V1.0
Maestro [9]	Centralized multi-threaded	Ad-hoc API	No	No	LGPLv2.1	Java	V1.0
Meridian [21]	Centralized multi-threaded	Extensible APIlayer	No	Noi	---	Java	V1.0
MobileFlow [22]	---	SDMN API	---	---	---	---	V1.2
Mul [23]	Centralized multi-threaded	Multi-level interface	No	No	GPLv2	C	V1.0
NOX [8]	Centralized	Ad-hoc API	No	No	GPLv3	C++	V1.0
NOX-MT [8]	Centralized multi-threaded	Ad-hoc API	No	No	GPLv3	C++	V1.0
NVP Controller [24]	Distributed	---	---	---	Commerical	---	---
OpenCOntrail [25]	---	REST API	No	No	Apache 2.0	Python, C++, Java	V1.0
OpenDaylight [26]	Distributed	REST, RESTCONF	Weak	No	EPL v1.0	Java	V1.{0,3}
ONOS [12]	Distributed	RESTful API	Weak, strong	Yes	---	Java	V1.0
PANE [17]	Distributed	PANE API	yes	---	---	---	---
POX [27]	Centralized	Ad-hoc API	No	No	GPLv3	Python	V1.0
ProgrammableFlow [28]	Centralized	---	---	---	---	C	V1.3
Rosemary [29]	Centralized	Ad-hoc	---	---	---	---	V1.0
Ryu NOS [30]	Centralized multi-threaded	Ad-hoc API	No	No	Apache 2.0	Python	V1.{0,2,3}
SMaRtLight [31]	Distributed	RESTful API	Yes	Yes	---	Java	V1.0
SNAC [32]	Centralized	Ad-hoc API	No	No	GPL	C++	V1.0
Trema [33]	Centralized multi-threaded	Ad-hoc API	No	No	GPLv2	C, Ruby	V1.0
Unified Controller [34]	---	REST API	---	---	Commerical	---	V1.0
Yanc [16]	Distributed	File system	---	---	---	---	---

- The control platform is one of the critical points
- Main issues: interoperability - Standardized APIs for multi-controller and multidomain deployments are necessary

SDN Architecture - Northbound Interfaces

- Northbound Interface → still open issues, a bit too early to define a northbound interface Standard
- Existing controllers such as Floodlight, Trema, NOX, Onix, and OpenDaylight propose and define their own northbound APIs [36], [37]. However, each of them has its own specific definitions
- Northbound APIs depend on types of the Application (security, routing, etc.)



Discussions

- What are the open issues?
- Is SDN/OpenFlow suitable for every future networking scenario? Can it replace today's networks?
- What are the key selling points of SDN?

Challenges

- **Scalability**
  - Bottleneck in SDN controller
    - ◇ Single controller is insufficient to manage a large SDN network → distributed, multiple controller scenarios
    - →How many controllers are needed to support large scale network?
    - →How multiple controllers are coordinated, load-balanced?
- **Security**
  - The SDN controller is vulnerable to attack
    - ◇ DDoS attack to the controller can be a serious threat for the whole network
    - ◇ Hacking to the controller or allowing unauthorized programs running on the controller is dangerous
    - → How the controllers are protected?

Why SDN solutions are attractive, despite of some open issues?

- Based on open interfaces and standards for control and management planes
  - Allowing integrating third-party applications and services
  - Fast service deployment
  - Highly customizable
  - Allowing contributions from open source communities → leverage innovation
- Low cost
  - Simple forwarding hardware devices
  - Reduced operational expenditure by automated network management and control
  - Many solutions are based on open source
- Suitable for some new networking paradigms
  - Fully virtualized cloud environments and data center
    - A new business model for network operators: virtualized network and services
    - Third-party service providers
  - Future 5G networking
  - IoT

Conclusions

- Key ideas of SDN
  - Decoupling control and data plane
  - Dynamic programmability in forwarding packets
  - Global view network by logical centralization in control plane
  - Applications can be implemented on top of the control plane
  - SDN is a concept to manage network that leverages OpenFlow protocols

Xin trân trọng cám ơn

References

[1] Software-Defined Networking: A Comprehensive Survey, Proceedings of the IEEE (Volume: 103, Issue: 1, Jan. 2015)

[2] A. Doria et al., "Forwarding and control element separation (ForCES) protocol specification," Internet Engineering Task Force, Mar. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5810.txt

[3] B. Pfaff and B. Davie, "The Open vSwitch database management protocol," Internet Engineering Task Force, RFC 7047 (Informational), Dec. 2013. [Online]. Available: http://www.ietf.org/rfc/ rfc7047.txt

[4] M. Smith et al., "OpFlex control protocol," Internet Engineering Task Force," Internet Draft, Apr. 2014. [Online]. Available: http:// tools.ietf.org/html/draft-smith-opflex-00

[5] G. Bianchi, M. Bonola, A. Capone, and C. Cascone, "OpenState: Programming platform-independent stateful OpenFlow applications inside the switch," SIGCOMM Comput. Commun. Rev., vol. 44, no. 2, pp. 44–51, Apr. 2014

[6] M. Sune, V. Alvarez, T. Jungel, U. Toseef, and K. Pentikousis, "An OpenFlow implementation for network processors," inProc. 3rd Eur. Workshop Softw. Defined Netw., 2014, 2 pp

[7] D. Erickson, "The Beacon OpenFlow controller," in Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw., 2013, pp. 13–18.

[8] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On controller performance in software-defined networks," in Proc. 2nd USENIX Conf. Hot Topics Manage. Internet Cloud Enterprise Netw. Services, 2012, p. 10.

[9] Z. Cai, A. L. Cox, and T. S. E. Ng, "Maestro: A system for scalable OpenFlow control," Rice Univ., Houston, TX, USA, Tech. Rep., 2011

[10] Project Floodlight, "Floodlight," 2012. [Online]. Available: http://floodlight.openflowhub.org/

[11] T. Koponen et al., "Onix: A distributed control platform for large-scale production networks," in Proc. 9th USENIX Conf. Oper.Syst. Design Implement., 2010, pp. 1–6

[12] U. Krishnaswamy et al., "ONOS: An open source distributed SDN OS," 2013.

[13] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in Proc. Internet Netw.Manage. Conf. Res. Enterprise Netw., 2010, p. 3.

References

[14] HP, "SDN controller architecture," Tech. Rep., Sep. 2013.

[15] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers," Aug. 2013. [Online]. Available: http://arxiv.org/abs/1308.6138

[16] M. Monaco, O. Michel, and E. Keller, "Applying operating system principles to SDN controller design," in Proc. 12th ACM Workshop Hot Topics Netw., College Park, MD, USA, Nov. 2013, DOI: 10.1145/ 2535771.2535789

[17] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi, "Participatory networking: An API for application control of SDNs," in Proc. ACM SIGCOMM Conf., 2013, pp. 327–338

[18] A. A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Elasticon: An elastic distributed SDN controller," in Proc. 10th ACM/IEEE Symp. Architect. Netw. Commun. Syst., 2014, pp. 17–28

[19] S. Matsumoto, S. Hitz, and A. Perrig, "Fleet: Defending SDNs from malicious administrators," in Proc. 3rd Workshop Hot Topics Softw. Defined Netw., 2014, pp. 103–108.

[20] S. Hassas Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in Proc.1st Workshop Hot Topics Softw. Defined Netw.,2012, pp. 19–24

[21] M. Banikazemi, D. Olshefski, A. Shaikh, J. Tracey, and G. Wang, "Meridian: An SDN platform for cloud network services," IEEE Commun. Mag., vol. 51, no. 2,pp. 120–127, Feb. 2013

[22] K. Pentikousis, Y. Wang, and W. Hu, "MobileFlow: Toward software-defined mobile networks," IEEE Commun. Mag.,vol. 51, no. 7, pp. 44–53, Jul. 2013

[23] D. Saikia, "MuL OpenFlow Controller," 2013. [Online]. Available: http://sourceforge.net/projects/mul/

[24] T. Koponen et al., "Network virtualization in multi-tenant datacenters," in Proc. 11th USENIX Symp. Netw. Syst. Design Implement., Apr. 2014, pp. 203–216

[25] Juniper Networks, "Opencontrail," 2013. [Online]. Available: http://opencontrail.org/

[26] OpenDaylight, A Linux Foundation Collaborative Project, 2013. [Online].Available: http://www.opendaylight.org

[27] M. McCauley, "POX," 2012. [Online]. Available: http://www.noxrepo.org/

References

[28] H. Shimonishi and S. Ishii, "Virtualized network infrastructure using OpenFlow," in Proc. IEEE/IFIP Netw. Oper. Manage. Symp. Workshops, 2010, pp. 74–79

[29] S. Shin et al., "Rosemary: A robust, secure, high-performance network operating system," in Proc. 21st ACM Conf. Comput. Commun. Security, Scottsdale, AZ, USA, Nov. 2014, pp. 78–89

[30] Nippon Telegraph and Telephone Corporation, "RYU network operating system," 2012

[31] F. Botelho, A. Bessani, F. M. V. Ramos, and P. Ferreira, "On the design of practical fault-tolerant SDN controllers," in Proc. 3rd Eur. Workshop Softw. Defined Netw., 2014, 6 pp

[32] G. Appenzeller, "SNAC," 2011. [Online]. Available: http://www. openflowhub.org/display/Snac

[33] Y. Takamiya and N. Karanatsios, "Trema OpenFlow controller framework," 2012. [Online]. Available: https://github.com/trema/trema

[34] S. Racherla et al., Implementing IBM Software Defined Network for Virtual Environments. Durham, NC, USA: IBM RedBooks, May 2014

[35] L. Richardson and S. Ruby, RESTful Web Services. Sebastopol, CA, USA: O'Reilly Media, 2008

[36] B. Salisbury, "The northbound APIVA big little problem," 2012.

[37] R. Chua, "OpenFlow northbound API: A new Olympic sport," 2012. [Online]. Available: http://www.sdncentral.com/sdn-blog/openflow-northbound-api-olympics/ 2012/07/

[38] Nick McKeown, Tom Anderson, H. Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner, "OpenFlow: enabling innovation in campus networks", ACM SIGCOMM 2008

References

[39] J. Reich, C. Monsanto, N. Foster, J. Rexford, and D. Walker, “Modular SDN Programming with Pyretic,” USENIX; login, vol. 38, no. 5, October 2013

[40] S. Gutz, A. Story, C. Schlesinger, and N. Foster, “Splendid isolation: A slice abstraction for software-defined networks,” in Proceedings of the First Workshop on Hot Topics in Software Defined Networks, ser. HotSDN ’12. New York, NY, USA: ACM, 2012, pp. 79–84.

[41] D. Turull, M. Hidell, and P. Sjodin, “Evaluating OpenFlow in libnetvirt,” in The 8th Swedish National Computer Networking Workshop 2012 (SNCNW 2012), Oct 2012

[42] M. Scharf, V. Gurbani, T. Voith, M. Stein, W. Roome, G. Soprovich, and V. Hilt, “Dynamic VPN optimization by ALTO guidance,” in Software Defined Networks (EWSDN), 2013 Second European Workshop on, Oct 2013, pp. 13–18

[43] N. Handigol, S. Seetharaman, M. Flajslik, A. Gember, N. McKeown, G. Parulkar, A.

Akella, N. Feamster, R. Clark, A. Krishnamurthy, V. Brajkovic, and T. A. and, “Aster*x: Load-Balancing Web Traffic over Wide-Area Networks,” 2009

[44] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “ElasticTree: saving energy in data center networks,” in Proceedings of the 7th USENIX conference on Networked systems design and implementation, ser. NSDI’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 17–17

[45] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, and Y.-Q. Song, “FlowQoS: QoS for the rest of us,” in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’14. New York, NY, USA: ACM, 2014, pp. 207–208

[46] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: dynamic flow scheduling for data center networks,” in Proceedings of the 7th USENIX conference on Networked systems design and implementation, ser. NSDI’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 19–19

[47] C. Macapuna, C. Rothenberg, and M. Magalhaes, “In-packet bloom filter based data center networking with distributed OpenFlow controllers,” in GLOBECOM Workshops (GC Wkshps), 2010 IEEE, 2010, pp. 584–588

References

[48] T. Benson, A. Anand, A. Akella, and M. Zhang, “MicroTE: Fine grained traffic engineering for data centers,” in Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies, ser. CoNEXT ’11. New York, NY, USA: ACM, 2011, pp. 8:1–8:12

[49] H. Jamjoom, D. Williams, and U. Sharma, “Don’t call them middleboxes, call them middlepipes,” in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’14. New York, NY, USA: ACM, 2014, pp. 19–24

[50] H. Egilmez, S. Dane, K. Bagci, and A. Tekalp, “OpenQoS: An Open-Flow controller design for multimedia delivery with end-to-end quality of service over software-defined networks,” in Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, 2012, pp. 1–8

[51] A. Sgambelluri, A. Giorgetti, F. Cugini, F. Paolucci, and P. Castoldi, “OpenFlow-based segment protection in Ethernet networks,” Optical Communications and Networking, IEEE/OSA Journal of, vol. 5, no. 9, pp. 1066–1075, Sept 2013

[52] M. Bari, S. Chowdhury, R. Ahmed, and R. Boutaba, “PolicyCop: An autonomic QoS policy enforcement framework for software defined networks,” in Future Networks and Services (SDN4FNS), 2013 IEEE SDN for, Nov 2013, pp. 1–7

[53] K. Nagaraj and S. Katti, “ProCel: Smart traffic handling for a scalable software epc,” in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN ’14. New York, NY, USA: ACM, 2014, pp. 43–48

[54] P. Xiong and H. Hacig¨um¨us, “Pronto: A software-defined networking based system for performance management of analytical queries on distributed data stores,” PVLDB, vol. 7, no. 13, pp. 1661–1664, 2014.

[55] P. Xiong, H. Hacigumus, and J. F. Naughton, “A software-defined networking based approach for performance management of analytical queries on distributed data stores,” in Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD ’14. New York, NY, USA: ACM, 2014, pp. 955–966

[56] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, “Plug-n-Serve: Load-balancing web traffic using OpenFlow,” 2009

[57] M. V. Neves, C. A. F. D. Rose, K. Katrinis and H. Franke, "Pythia: Faster Big Data in Motion through Predictive Software-Defined Network Optimization at Runtime," *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, Phoenix, AZ, 2014, pp. 82-90

References

[58] K. Jeong, J. Kim, and Y.-T. Kim, "QoS-aware network operating system for software defined networking with generalized OpenFlows," in Network Operations and Management Symposium (NOMS), 2012 IEEE, April 2012, pp. 1167 –1174

[59] S. Sharma, D. Staessens, D. Colle, D. Palma, J. G. R. Figueiredo, D. Morris, M. Pickavet, and P. Demeester, "Implementing quality of service for the software defined networking enabled future internet," in Third European Workshop on Software Defined Networks, 2014, pp.

[60] W. Kim, P. Sharma, J. Lee, S. Banerjee, J. Tourrilhes, S.-J. Lee, and P. Yalagandula, "Automated and scalable QoS control for network convergence," in Proceedings of the 2010 internet network management conference on Research on enterprise networking, ser. INM/WREN'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–1

[61] A. Ishimori, F. Farias, E. Cerqueira, and A. Abelem, "Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking," in Software Defined Networks (EWSDN), 2013 Second European Workshop on, Oct 2013, pp. 81–86

[62] D. Palma, J. Goncalves, B. Sousa, L. Cordeiro, P. Simoes, S. Sharma, and D. Staessens, "The QueuePusher: Enabling queue management in OpenFlow," in Third European Workshop on Software Defined Networks, 2014, pp

[63] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "SIMPLE-fying middlebox policy enforcement using SDN," in Proceedings of the Conference on Applications, technologies, architectures, and protocols for computer communications, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013

[64] P. Skoldstrom and B. C. Sanchez, "Virtual aggregation using SDN," in 2013 Second European Workshop on Software Defined Networks, 2013, pp. –

[65] J. Schulz-Zander, N. Sarrar, and S. Schmid, "AeroFlux: A nearsighted controller architecture for software-defined wireless networks," in Presented as part of the Open Networking Summit 2014 (ONS 2014). Santa Clara, CA: USENIX, 2014.

[66] J. Schulz-Zander, N. Sarrar, and S. Schmid, "Towards a scalable and near-sighted control plane architecture for WiFi SDNs," in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '14, 2014

[67] H. Ali-Ahmad, C. Cicconetti, A. de la Oliva, M. Draxler, R. Gupta, V. Mancuso, L. Roullet, and V. Sciancalepore, "CROWD: An SDN approach for densenets," in Software Defined Networks (EWSDN), 2013 Second European Workshop on, Oct 2013, pp. 25–31

References

[68] J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC: towards software defined WLANs," SIGMOBILE Mob. Comput. Commun. Rev., vol. 16, no. 4, pp. 42–45, Feb. 2013

[69] A. Dawson, M. K. Marina, and F. J. Garcia, "On the benefits of RAN virtualisation in C-RAN based mobile networks," in Third European Workshop on Software Defined Networks, 2014, pp. –

[70] Y. Yamasaki, Y. Miyamoto, J. Yamato, H. Goto, and H. Sone, "Flexible access management system for campus VLAN based on OpenFlow," in Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on, 2011, pp. 347–351

[71] K. Pentikousis, Y. Wang, and W. Hu, "MobileFlow: Toward software defined mobile networks," Communications Magazine, IEEE, vol. 51, no. 7, pp. 44–53, 2013

[72] J. Schulz-Zander, L. Suresh, N. Sarrar, A. Feldmann, T. Huehn, and R. Merz, "Programmatic orchestration of WiFi networks," in 2014 USENIX Annual Technical Conference (USENIX ATC 14), 2014

[73] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "OpenRAN: a software-defined ran architecture via virtualization," in Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 549–550

[74] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: empowering research in mobile networks," SIGCOMM Comput. Commun. Rev., vol. 40, no. 1, pp. 125–126, Jan. 2010

[75] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in Proceedings of the second workshop on Hot topics in software defined networks, ser. HotSDN '13. New York, NY, USA: ACM, 2013

[76] H. Kim and N. Feamster, "Improving network management with software defined networking," Communications Magazine, IEEE, vol. 51, no. 2, pp. 114–119, 2013

[77] Y. Yu, C. Qian, and X. Li, "Distributed and collaborative traffic monitoring in software defined networks," in Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '14. New York, NY, USA: ACM, 2014, pp. 85–90

[78] S. Shirali-Shahreza and Y. Ganjali, "FleXam: flexible sampling extension for monitoring and security applications in OpenFlow," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 167–168

References

[79] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, “FlowSense: monitoring network utilization with zero measurement cost,” in Proceedings of the 14th international conference on Passive and Active Measurement, ser. PAM’13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 31–41

[80] L. Jose, M. Yu, and J. Rexford, “Online measurement of large traffic aggregates on commodity switches,” in Proceedings of the 11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services, ser. Hot-ICE’11. Berkeley, CA, USA: USENIX Association, 2011, pp. 13–13

[81] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, “OpenNetMon: Network monitoring in openflow software-defined networks,” in 2014 IEEE Network Operations and Management Symposium, NOMS 2014, Krakow, Poland, May 5-9, 2014, 2014, pp. 1–8.

[82] J. Suh, T. Kwon, C. Dixon, W. Felter, and J. Carter, “OpenSample: A low-latency, sampling-based measurement platform for commodity SDN,” in Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on, June 2014, pp. 228–237

[83] M. Yu, L. Jose, and R. Miao, “Software defined traffic measurement with OpenSketch,” in Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation, ser. nsdi’13. Berkeley, CA, USA: USENIX Association, 2013, pp. 29–42

[84] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, “OpenTM: traffic matrix estimator for OpenFlow networks,” in Proceedings of the 11th international conference on Passive and active measurement, ser. PAM’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 201–210

[85] C. Argyropoulos, D. Kalogeras, G. Androulidakis, and V. Maglaris, “PaFloMon – a slice aware passive flow monitoring framework for OpenFlow enabled experimental facilities,” in Software Defined Networking (EWSDN), 2012 European Workshop on, 2012, pp. 97–102.

[86] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, “PayLess: A Low Cost Netowrk Monitoring Framework for Software Defined Networks,” in 14th IEEE/IFIP Network Operations and Management Symposium (NOMS 2014), 2014

[87] G. Wang, T. E. Ng, and A. Shaikh, “Programming your network at run-time for big data applications,” in HotSDN. ACM, 2012.

[88] T. Benson, A. Akella, A. Shaikh, and S. Sahu, “Cloudnaas: a cloud networking platform for enterprise applications,” in Proceedings of the 2nd ACM Symposium on Cloud Computing, ser. SOCC ’11. New York, NY, USA: ACM, 2011, pp. 8:1–8:13

References

[89] A. Das, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and C. Yu, "Transparent and flexible network management for big data processing in the cloud," in Proceedings of the 5th USENIX conference on Hot Topics in Cloud Ccomputing, ser. HotCloud'13. Berkeley, CA, USA: USENIX Association, 2013

[90] A. Arefin, V. K. Singh, G. Jiang, Y. Zhang, and C. Lumezanu, "Diagnosing data center behavior flow by flow," in IEEE 33rd International Conference on Distributed Computing Systems. Philadelphia, USA: IEEE, July 2013.

[91] E. Keller, S. Ghorbani, M. Caesar, and J. Rexford, "Live migration of an entire network (and its hosts)," in Proceedings of the 11th ACM Workshop on Hot Topics in Networks, ser. HotNets-XI. New York, NY, USA: ACM, 2012, pp. 109–114

[92] R. Raghavendra, J. Lobo, and K.-W. Lee, "Dynamic graph query primitives for SDN-based cloudnetwork management," in Proceedings of the first workshop on Hot topics in software defined networks, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 97–102.

[93] M. Ghobadi, "TCP Adaptation Framework in Data Centers," Ph.D. dissertation, Graduate Department of Computer Science of University of Toronto, 2013

[94] R. Hand, M. Ton, and E. Keller, "Active Security," in Twelfth ACM Workshop on Hot Topics in Networks (HotNets-XII), College Park, MD, November 2013.

[95] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-De?ned Networks," in Proceedings of the 2013 ACM conference on Computer and communications security, ser. CCS '13. New York, NY, USA: ACM, 2013.

[96] S. Shin and G. Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)," in Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP), ser. ICNP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1–6.

[97] E. Tantar, M. Palattella, T. Avanesov, M. Kantor, and T. Engel, "Cognition: A tool for reinforcing security in software defined networks," in EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V, ser. Advances in Intelligent Systems and Computing, A.-A. Tantar, E. Tantar, J.-Q. Sun, W. Zhang, Q. Ding, O. Schtze, M. Emmerich, P. Legrand, P. Del Moral, and C. A. Coello Coello, Eds. Springer International Publishing, 2014, vol. 288, pp. 61–78.

References

[98] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using NOX/OpenFlow,” in Local Computer Networks (LCN), 2010 IEEE 35th Conference on, Oct 2010, pp. 408 –415.

[99] G. Stabler, A. Rosen, S. Goasguen, and K.-C. Wang, “Elastic ip and security groups implementation using OpenFlow,” in Proceedings of the 6th international workshop on Virtualization Technologies in Distributed Computing Date, ser. VTDC ’12. New York, NY, USA: ACM, 2012, pp. 53–60

[100] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, “Ethane: taking control of the enterprise,” in Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, ser. SIGCOMM ’07. New York, NY, USA: ACM, 2007, pp. 1–12

[101] J. Matias, J. Garay, A. Mendiola, N. Toledo, and E. Jacob, “FlowNAC: Flow-based network access control,” in Third European Workshop on Software Defined Networks, 2014, pp. –

[102] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, “A security enforcement kernel for OpenFlow networks,” in Proceedings of the First Workshop on Hot Topics in Software Defined Networks, ser. HotSDN ’12. New York, NY, USA: ACM, 2012, pp. 121–126

[103] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, “FRESCO: Modular composable security services for software-defined networks,” in Internet Society NDSS., Feb 2013

[104] K. Wang, Y. Qi, B. Yang, Y. Xue, and J. Li, “LiveSec: Towards Effective Security Management in Large-Scale Production Networks,” in Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, june 2012, pp. 451 –460.

[105] A. Sapio, M.Baldi, Y.Liao, G.Ranjan, F. Risso, A. Tongaonkar, R.Torres, and A. Nucci, “MAPPER: A mobile application personal policy enforcement router for enterprise networks,” in Third European Workshop on Software Defined Networks, 2014, pp. –.

[106] Y. Wang, Y. Zhang, V. Singh, C. Lumezanu, and G. Jiang, “NetFuse: Short-Circuiting Traffic Surges in the Cloud,” in IEEE International Conference on Communications, 2013.

[107] J. H. Jafarian, E. Al-Shaer, and Q. Duan, “OpenFlow random host mutation: transparent moving target defense using software defined networking,” in Proceedings of the first workshop on Hot topics in software defined networks, ser. HotSDN ’12. New York, NY, USA: ACM, 2012, pp. 127–132

References

[108] J. R. Ballard, I. Rae, and A. Akella, "Extensible and scalable network monitoring using OpenSAFE," in Proceedings of the 2010 internet network management conference on Research on enterprise networking, ser. INM/WREN'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 8–8.

[109] A. Zaalouk, R. Khondoker, R. Marx, and K. Bayarou, "OrchSec: An orchestrator-based architecture for enhancing network-security using network monitoring and SDN control functions," in Network Operations and Management Symposium (NOMS), 2014 IEEE, May 2014, pp. 1–9

[110] D. Kotani, K. Suzuki, and H. Shimonishi, "A design and implementation of OpenFlow controller handling ip multicast with fast tree switching," in Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on, 2012, pp. 60–67

[111] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker, "SANE: a protection architecture for enterprise networks," in Proceedings of the 15th conference on USENIX Security Symposium - Volume 15, ser. USENIX-SS'06, Berkeley, CA, USA, 2006

[112] K. Giotis, G. Androulidakis, and V. Maglaris, "Leveraging SDN for efficient anomaly detection and mitigation on legacy networks," in Third European Workshop on Software Defined Networks, 2014, pp. –

[113] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," in Network Protocols (ICNP), 2011 19th IEEE International Conference on, 2011, pp. 7–12

[114] R Niranjan Mysore, A Pamboris, N Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, Amin Vahdat, "PortLand: a scalable fault-tolerant layer 2 data center network fabric", ACM SIGCOMM 2008

[115] Tran Manh Nam, Nguyen Van Huynh, Le Quang Dai, Nguyen Huu Thanh, "*An Energy-Aware Embedding Algorithm for Virtual Data Centers*", accepted to the 28th International Teletraffic Congress (ITC 28), September 12th – 16th 2016, Wuerzburg, Germany

[116] Matthew Caesar, Donald Caldwell, Nick Feamster, Jennifer Rexford, Aman Shaikh, and Jacobus van der Merwe. 2005. 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), Vol. 2. USENIX Association, Berkeley, CA, USA, 15-28