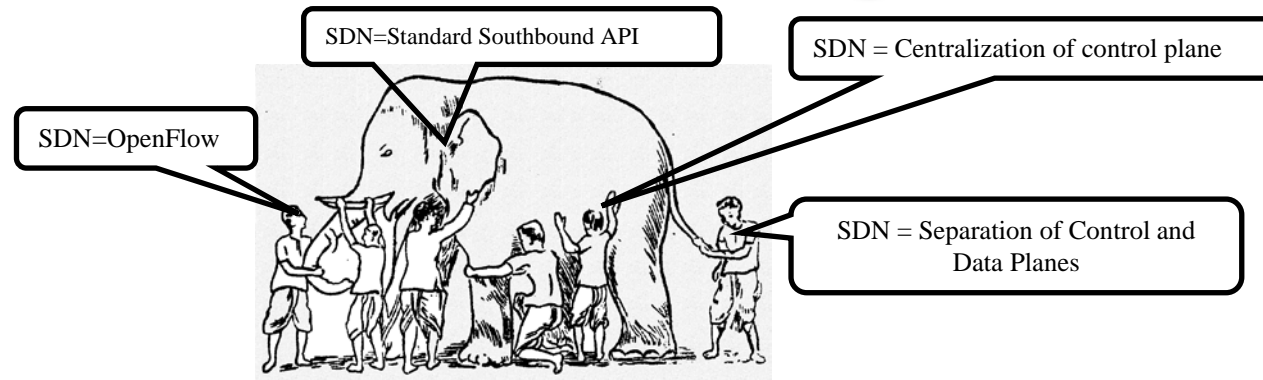


# Introduction to Software Defined Networking (SDN)



Raj Jain  
Washington University in Saint Louis  
Saint Louis, MO 63130  
Jain@cse.wustl.edu

These slides and audio/video recordings of this class  
lecture are at:

<http://www.cse.wustl.edu/~jain/cse570-23/>

## Student Questions



1. What is SDN?
2. SDN Controllers
3. Alternative APIs: XMPP, PCE, ForCES, ALTO
4. RESTful APIs and OSGi Framework

Note: This is the second module of three modules on OpenFlow, SDN, and NFV in this course.

## Student Questions

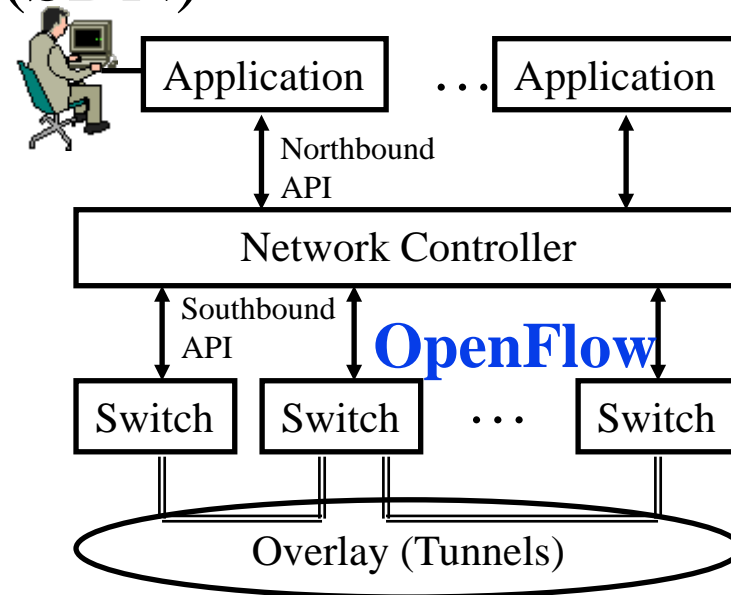
# Origins of SDN

- ❑ SDN originated from OpenFlow
- ❑ Centralized Controller
  - ⇒ Easy to program
  - ⇒ Change routing policies on the fly
  - ⇒ Software Defined Network (SDN)

❑ Initially, SDN=

- Separation of Control and Data Plane
- Centralization of Control
- OpenFlow to talk to the data plane

❑ Now, the definition has changed

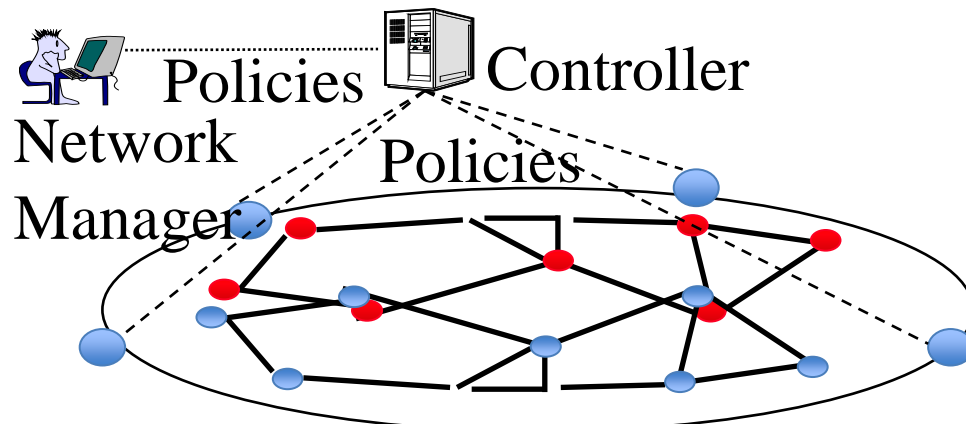


## Student Questions

- ❑ Please can you explain further the difference between control and centralized?  
*Controller = Boss who makes policies*  
*Distributed Controllers = Many bosses*  
*Centralized Controller = One boss*

# Three Features that Define SDN

1. **Abstract the Hardware:** No dependence on physical infrastructure. Software API.
2. **Programmable:** Shift away from static, manual operation to fully configurable and dynamic
3. **Centralized Control of Policies:**  
Policy delegation and management



## Student Questions

- ❑ Where is the SDN being deployed? Is it on server machines? Networking hardware like switches?

*Networking hardware, like routers and switches, load balancers, and firewalls, can be software-defined. SDN is not limited to networking.*

- ❑ Will this abstraction from the hardware result in slower performance?

*Maybe. But it allows people to develop software independent of the hardware, similar to windows applications.*

# What = Why We need SDN?

- 1. Virtualization:** Use network resources without worrying about where it is physically located, how much it is, how it is organized, etc. Abstraction  $\Rightarrow$  Virtualization.
- 2. Orchestration:** Should be able to control and manage thousands of devices with one command.
- 3. Programmable:** Should be able to change behavior on the fly.
- 4. Dynamic Scaling:** Should be able to change the size, quantity  
Virtualization  $\Rightarrow$  Scaling
- 5. Automation:** To lower OpEx and minimize manual involvement
  - Troubleshooting
  - Reduce downtime
  - Policy Enforcement
  - Provisioning/Re-provisioning/Segmentation of resources
  - Add new workloads, sites, devices, and resources

## Student Questions

- ❖ Is AWS one of the SDN solutions nowadays?  
*No. Amazon Web Services is a cloud computing service. It pre-dates SDN.*
- ❖ Virtualization is one of many concepts that are built into SDNs. Would a network only utilizing virtualization and no other defining concepts of SDNs be considered an SDN? Would virtual networks created before the SDN concept was defined be considered SDNs in today's terms?

*No. Virtualization does not define SDN. It is made possible by SDN. Virtual networks created before SDN cannot be considered SDN.*

# Why We need SDN? (Cont)

**6. Visibility:** Monitor resources, connectivity

**7. Performance:** Optimize network device utilization

- Traffic engineering/Bandwidth management
- Capacity optimization
- Load balancing
- High utilization
- Fast failure handling

**8. Multi-tenancy:** Tenants need complete control over their addresses, topology, routing, security

**9. Service Integration:** Load balancers, firewalls, and Intrusion Detection Systems (IDS) provisioned on demand and placed appropriately on the traffic path

## Student Questions

# Why We need SDN? (Cont)

## 10. Openness: Full choice of “How” mechanisms

⇒ Modular plug-ins

⇒ Abstraction:

➤ Abstract = Summary = Essence = General Idea

⇒ Hide the details.

➤ Also, the abstract is the opposite of concrete

⇒ Define tasks by APIs and **not by how** they should be done. E.g., send from A to B. Not OSPF.

## Student Questions

Ref: <http://www.networkworld.com/news/2013/110813-onug-sdn-275784.html>

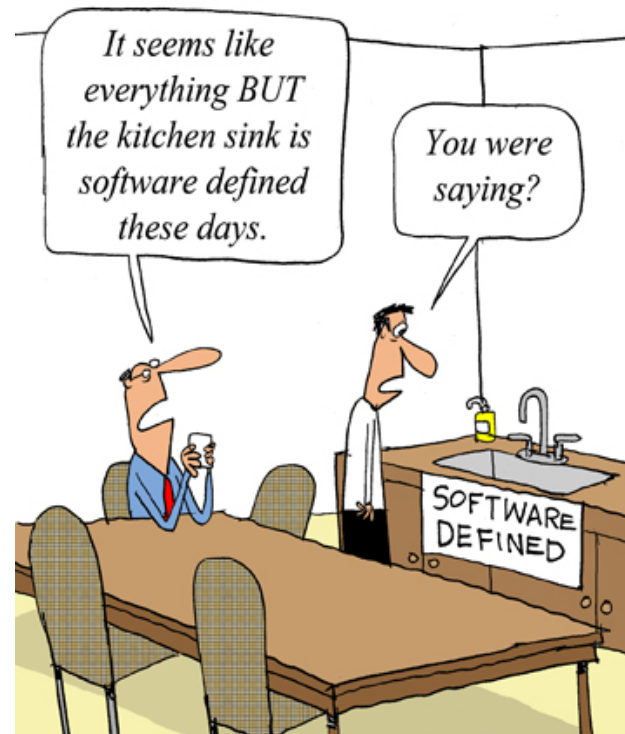
Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

# Software Defined Anything (SDx)

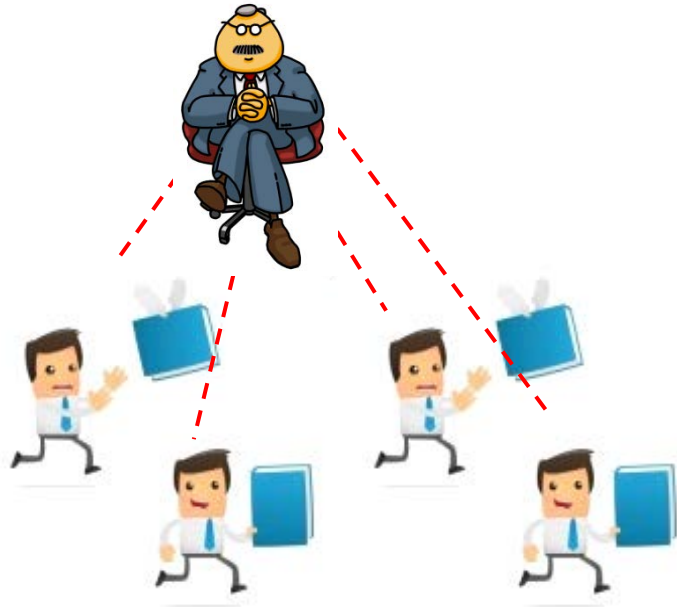
- ❑ Tsunami of software-defined things
  - Software Defined Networking (SDN)
  - Software Defined Datacenter (SDDC)
  - Software Defined Storage (SDS)
  - Software Defined Compute (SDC)
  - Software Defined Infrastructure (SDI)



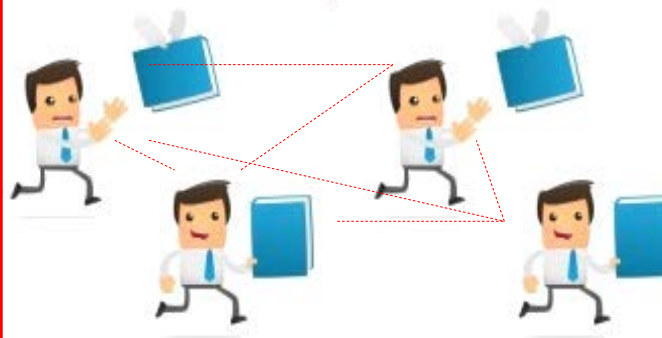
## Student Questions



# Centralized vs. Distributed Management



- ❑ Fast Response to changes
- ❑ Fast Consistency
- ❑ Less overhead ⇒ Scalable
- ❑ Single Point of Failure



- ❑ Time to converge
- ❑ Slow consistency
- ❑ Not scalable
- ❑ Fault Tolerant

## Student Questions

- ❑ Distributed services increase scalability. Perhaps the slide intended to mean that 'A centralized control plane, separated from the data plane, enhances the scalability of the entire system'?

*This slide is only about the management plane and not about the performance of the data plane. In SDN, the data plane is still distributed. The management plane is centralized.*

- ❑ Why is the decentralized not scalable? Is it due to the overhead?

*Decentralized management is not scalable. You can not have 300 Million Congress representatives to manage the USA.*

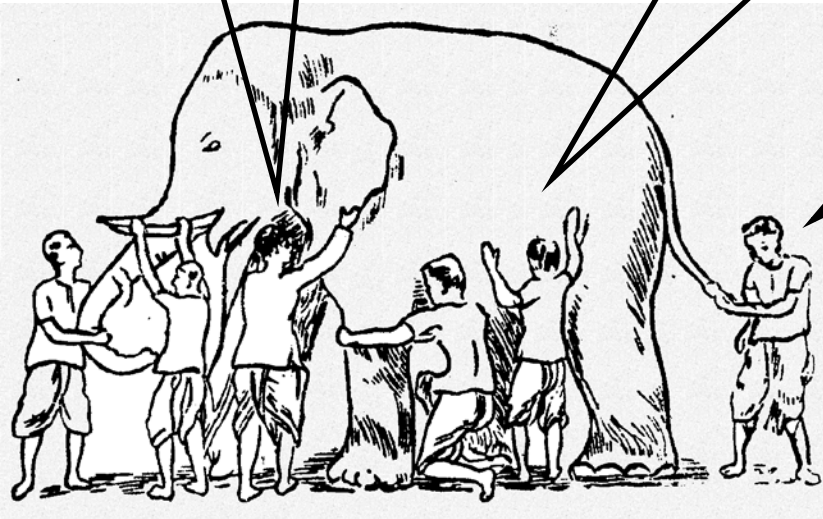
# What SDN is Not?

SDN = OpenFlow

SDN = Standard Southbound API

SDN = Centralization of control plane

SDN = Separation of Control and Data Planes



- ❑ All of these are mechanisms.
- ❑ SDN is *not* about a mechanism.
- ❑ It is a framework  $\Rightarrow$  Many solutions

## Student Questions

- ❑ Is SDN a conceptual thing? I'm still confused.

*SDN is a concept. Everyone has a different meaning of this concept. Like, "Beauty is in the eyes of the beholder." Beauty is a concept.*

# Four Confusions About SDN

## 1. Policies vs. Control:

Control = All bits and messages not sent by the user. In IP, control includes all header bits and all routing messages.

## 2. Separation of Control Plane:

Elements have only a data plane and have no brains

## 3. SDN vs. OpenFlow:

OpenFlow is the father of SDN but not SDN.

## 4. Need OpenFlow:

- OpenFlow is micro-management.
- It is not scalable.
- For extensive infrastructure, need scalable

## Student Questions

- Is the SDN scalable?

*Every centralized concept has its scalability limit. SDN, depending upon how you define and implement it, has its limit." Distributed has a much higher limit, but then there is an issue of consistency.*

- Can you say more about point 1? Is the idea that it is INCORRECT to say that the user does not send any control data?

*The user specifies control indirectly to the service layer. The service layer makes the control bits as needed.*

*Policies are rules.*

- To clarify once more, are these confusions all incorrect? Can you explain the three again?

*Yes. Point 3 states that OpenFlow is one of many ways to do SDN. I have seen papers breaking the security of SDN. They were breaking the security of OpenFlow.*

# Separation vs. Centralization

Separation of  
Control Plane



Centralization of  
Policies



**Student Questions**

**Micromanagement is not scalable**

# Current SDN Debate: What vs. How?

1. SDN is easy if control is centralized but not necessary.  
Distributed/hierarchical solutions may be required for fail-safe operation.
2. Complete removal of the control plane may be harmful.  
The exact division of the control plane between a centralized controller and distributed forwarders is yet to be worked out.

## Student Questions

- ❑ Why is everything moving to SDN? Is it the case that most solutions can work around these drawbacks?

*Yes, this debate has now subsided with multiple redundant controllers. This slide shows the two sides of a coin.*

- ❑ Distributed solutions: where would it be distributed?

*It is distributed among multiple managers.*

## Current SDN Debate: What vs. How? (Cont)

3. SDN is accessible with a standard southbound protocol like OpenFlow, but one protocol may not work/scale in all cases
  1. Diversity of protocols is a fact of life.
  2. There are no standard operating systems, processors, routers, or Ethernet switches.
4. If the industry finds an easier way to solve the same problems by another method, that method may win. E.g., ATM vs. MPLS.

### Student Questions

- Is there a security risk in using SDN? The controller of SDN is software installed on Windows or Linux, which makes the controller and the operating system face the same risk when attacked.

*Attacks on the controller can have a devastating effect since they can turn off the whole network. So, the controller is heavily guarded, monitored, and secured (locked).*

---

# Flavors of SDN

1. OpenDaylight: Multi-Protocol Southbound
2. Bare Metal Switches + Network Operating System
  - a. Switches from Dell, Edgecore, HP, Penguin, QCT, Agema, Supermicro
  - b. Open Network Install Environment (ONIE) on a set of programmable switch
  - c. Network operating systems: Alcatel-Lucent, Arista, Big Switch, Broadcom, Brocade, Cisco, Cumulus, Dell, Ericsson, Extreme, HP, Juniper, OCP, Pica8, Pluribus
3. Network Virtualization/Overlay: VMWare's NSX
4. ONF SDN: OpenFlow southbound

All provide Abstraction, Programmability, and Centralization

Ref: <http://onie.org/>, [http://www.opencompute.org/wiki/Networking/ONIE/HW\\_Status](http://www.opencompute.org/wiki/Networking/ONIE/HW_Status),  
[https://en.wikipedia.org/wiki/List\\_of\\_SDN\\_controller\\_software](https://en.wikipedia.org/wiki/List_of_SDN_controller_software)

Source: Alan J Weissberger

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

## Student Questions

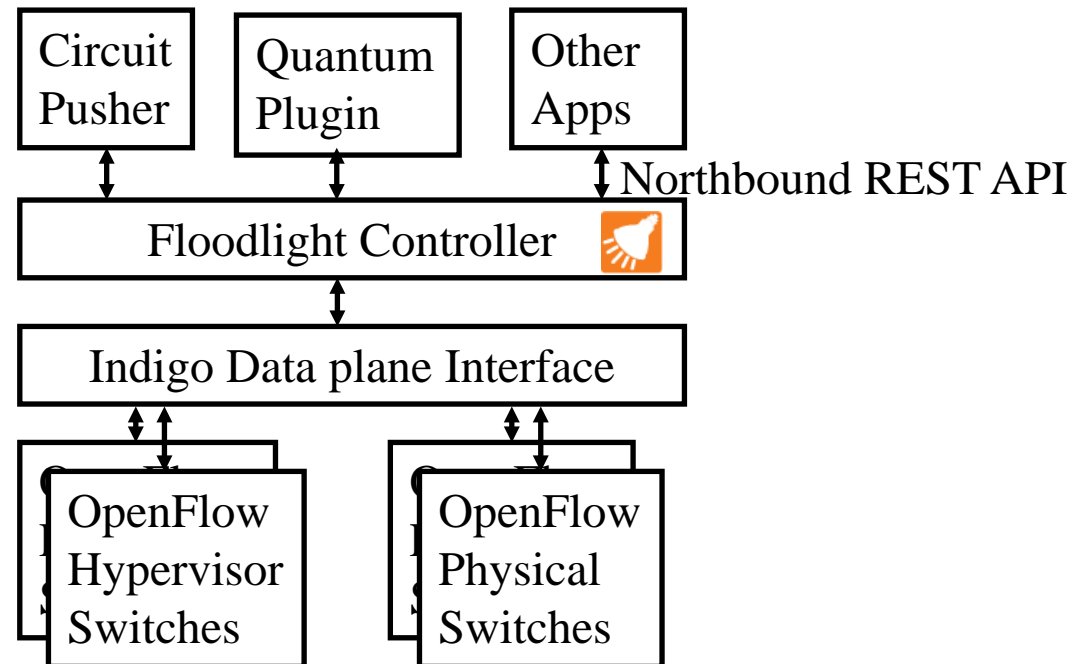
- ❑ So SDN, specifically OpenDaylight, is running on a single computer playing the controller role? Or is it a vast framework that lies in multiple layers of devices?

*SDN is a framework that affects multiple layers. The control for all those layers is centralized in a single computer.*



# Floodlight

- ❑ **Java-based** OpenFlow controller based on a Beacon runs within a JVM. Developers from Big Switch Networks
- ❑ Indigo: Software to make switch hardware OpenFlow compatible
- ❑ Floodlight is the core of the Big Switch Controller from Big Switch Networks



Ref: S. Azodolmolky, "Software Defined Networking with OpenFlow," Packt Publishing, October 2013, 152 pp., ISBN:978-1-84969-872-6 (Safari Book) Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

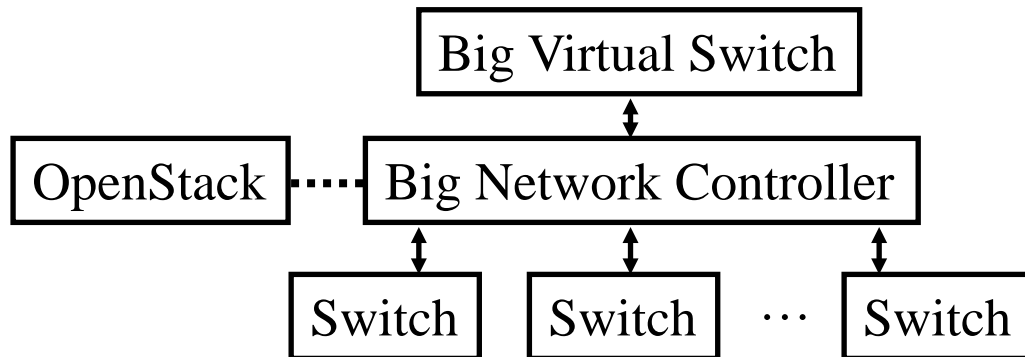
©2023 Raj Jain

## Student Questions



# Floodlight (Cont)

- ❑ Several real-world networking applications
  - **Neutron plug-in** for OpenStack cloud management system
  - **Static Flow Pusher**: Allows users to insert flows manually
  - **Circuit Pusher**: Creates permanent entries on all switches along the path
  - **Firewall**: Enforces access control list (ACL) rules on packets
  - **Big Virtual Switch**: Automates network provisioning for large-scale data centers. Includes provisioning, multi-tenant partitioning

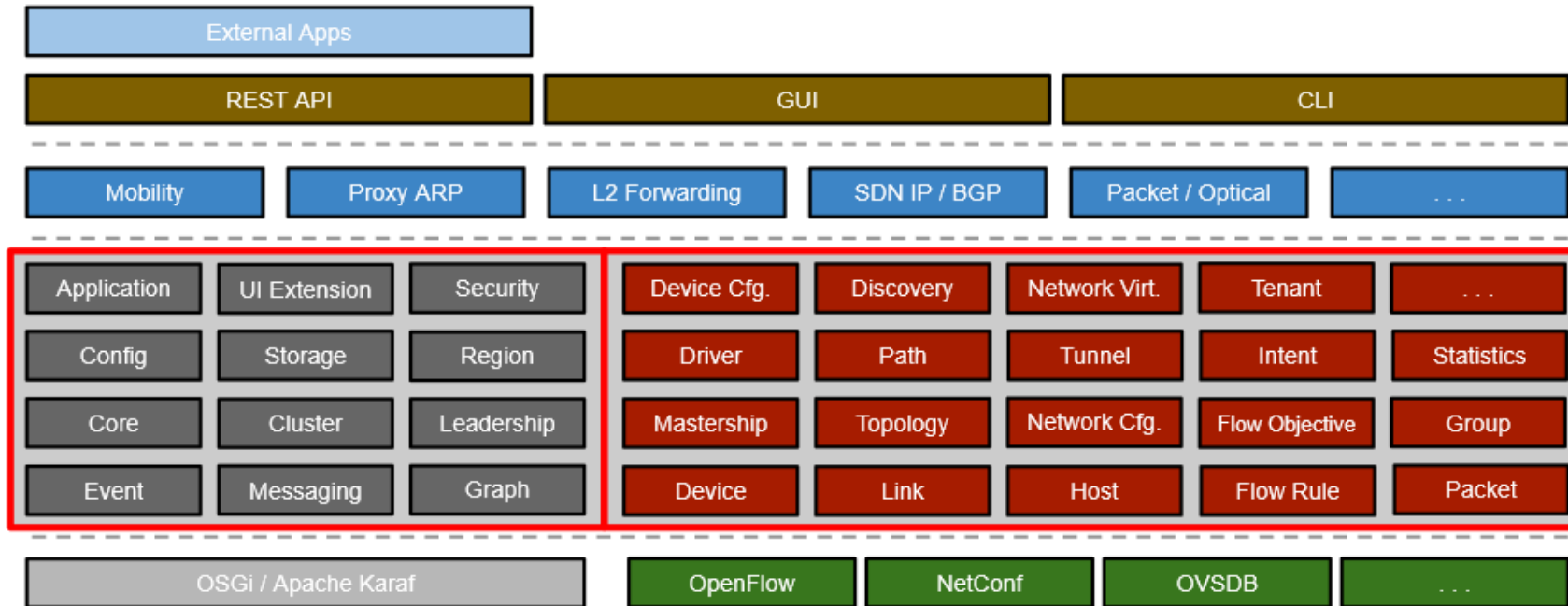


## Student Questions

Ref: <http://www.projectfloodlight.org/floodlight/>

# ONOS

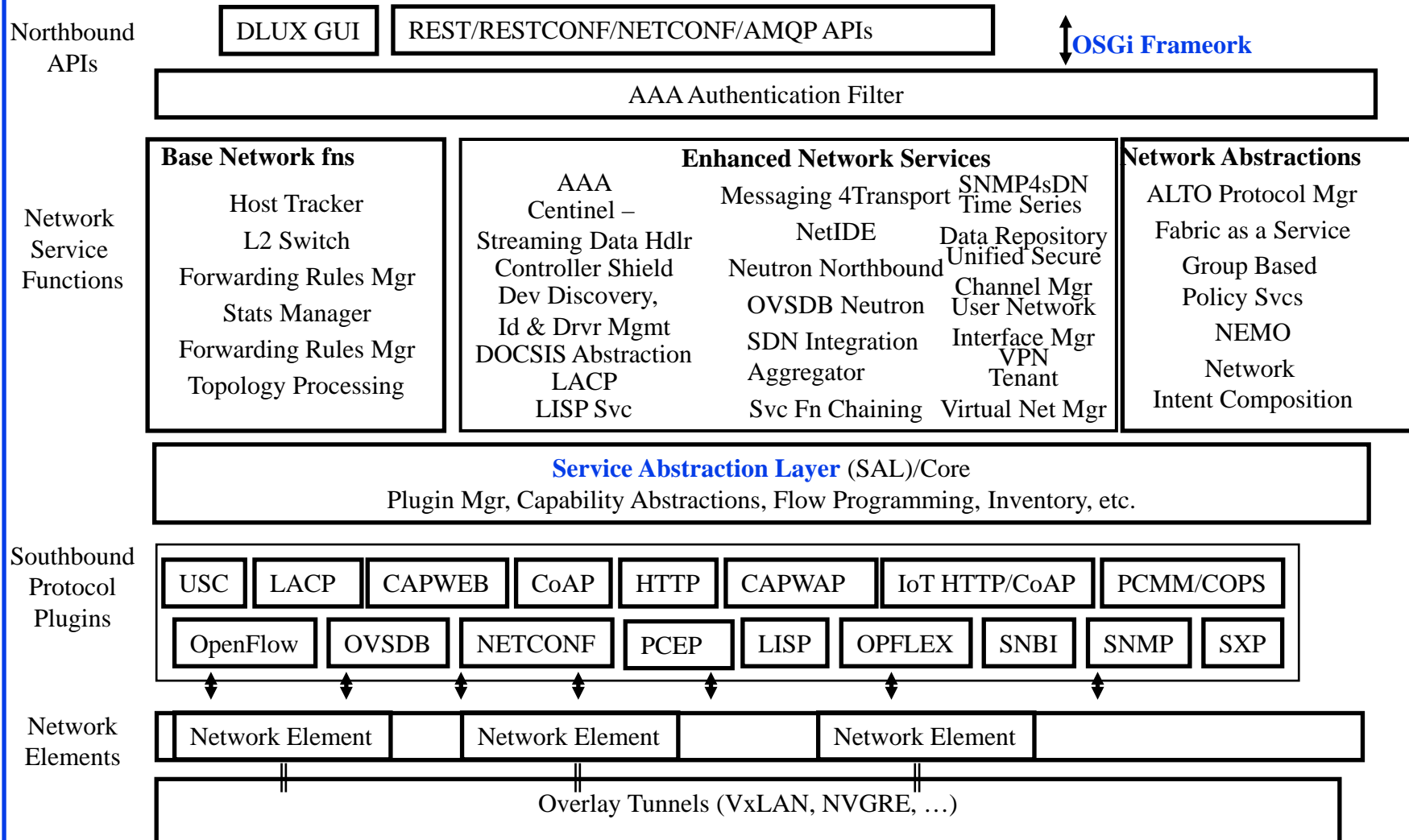
- ❑ Open Network Operating System:  
Distributed OpenFlow OS for a large WAN
- ❑ Initially OpenFlow-only. Now multi-protocol



## Student Questions

Ref: ONOS Architecture, [https://docs.google.com/presentation/d/1Y4S82YZyskqnKAzW4kKm-6IIO04h\\_nnrRQyfrsCdt-I/edit?usp=sharing](https://docs.google.com/presentation/d/1Y4S82YZyskqnKAzW4kKm-6IIO04h_nnrRQyfrsCdt-I/edit?usp=sharing)

# OpenDaylight: Multi-Protocol SDN



## Student Questions

- ❑ What is the role of AAA here?  
AAA = Authentication, Authorization, and Access Control  
*Authentication = You are who you say you are.*  
*Authorization = Use only what you are allowed.*  
*Access Control = You have access to what you are allowed and no access to what you are not.*
- ❑ Is it correct that southbound protocols are more associated with lower-level hardware concepts than northbound, which deal with application management?  
*Yes. This is how the 7-layer protocol stack is designed, with the application at the top and the physical layer at the bottom.*

# OpenDaylight SDN Controller Platform (OSCP)

- ❑ Multi-company collaboration under the Linux Foundation
- ❑ Many projects, including OpenDaylight Controller
- ❑ Supports multiple southbound protocols via plug-ins, including OpenFlow
- ❑ Dynamically linked into a Service Abstraction Layer (SAL) Abstraction  $\Rightarrow$  SAL figures out how to fulfill the service requested by higher layers irrespective of the southbound protocol
- ❑ Modular design using **OSGI framework**
- ❑ A rich set of North-bound APIs via **RESTful** services for loosely coupled applications and OSGI services for co-located applications using the same address space

Ref: C. Eckel, "OpenDaylight as a Platform for Network Programmability,"

<http://events17.linuxfoundation.org/sites/events/files/slides/OpenDaylight-Network-Programmability.pdf>

## Student Questions

- ❑ What is a RESTful service?

*REpresentational*

*State*

*Transfer*

*A method of service that makes servers' jobs easy by not requiring them to maintain a state. This is how web servers work.*

- ❑ What was the OSGI framework here? Does it allow for the modular feature of OpenDaylight?

*OSGI Framework is a method to architect and implement software in a modular style.*

*OpenDaylight uses the OSGI framework.*

- ❑ Can you give a diagram of the OSGI framework?

*See the Slide 15.38*

# Examples Alternative APIs

- ❑ Southbound APIs: PCEP, BGP, ...
- ❑ Northbound APIs: ALTO, ...
- ❑ Overlay: VxLAN, TRILL, LISP, ...
- ❑ Configuration API: NETCONF, RESTCONF, ...
- ❑ Controller: PCE, ...

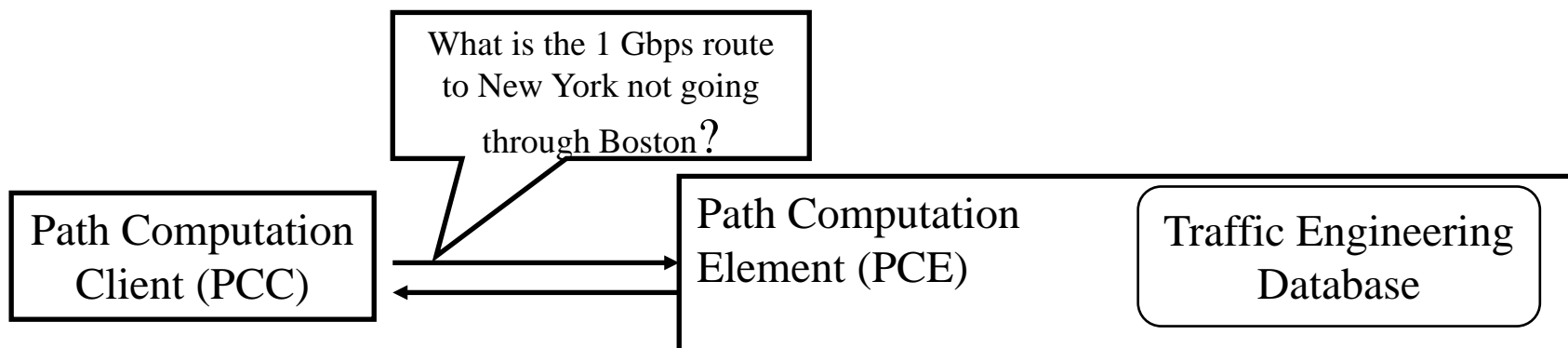
## Student Questions

- ❑ Is it still north = clients, south = servers?  
*North = up*  
*South = Down*  
*If you show clients on the top, clients are north.*  
*If you show clients on the bottom, clients are south. In this module so far, we have shown the user on the top.*
- 

Ref: T. Nadeau and K. Gray, "SDN," O'Reilly, 2013, 384 pp, ISBN:978-1-449-34230-2 (Safari Book)

# Path Computation Element (PCE)

- ❑ MPLS and GMPLS require originating routers  
To find paths that satisfy multiple constraints, including not using any backup routers and having a given bandwidth, etc.
- ❑ This may require more computing power or network knowledge than a router may have.
- ❑ The IETF PCE working group has developed a set of protocols that allow a Path computation client (PCC), i.e., router, to get the path from the path computation element (PCE)
- ❑ PCE may be centralized or distributed in many or every router.



## Student Questions

## PCE (Cont)

- ❑ PCE separates the route computation function from the forwarding function.
- ❑ Both functions may be resident in the same box or different boxes.
- ❑ 25+ RFCs documenting protocols for:
  - PCE-to-PCC communication
  - PCE-to-PCE communication (Multiple PCEs)
  - PCE discovery

Ref: <http://datatracker.ietf.org/wg/pce/>

Ref: [http://en.wikipedia.org/wiki/Path\\_computation\\_element](http://en.wikipedia.org/wiki/Path_computation_element)

Washington University in St. Louis

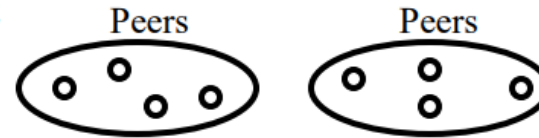
<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

## Student Questions

# Application Layer Traffic Optimization (ALTO)

- ❑ IETF working group to optimize P2P traffic  
⇒ Better to get files from nearby peers
- ❑ Guide peer selection
- ❑ ALTO Server: Knows distributed resources
- ❑ ALTO Client: Requests information from servers about the appropriate peers
- ❑ Ratio Criteria: Topological distance, traffic charges, ...
- ❑ ALTO Server could get information from providers or nodes about their characteristics, e.g., flat-rate or volume-based charging
- ❑ A client may get the list of potential peers and send it to the server, which can return an ordered list
- ❑ Also, need a protocol for ALTO server discovery



## Student Questions

- ❑ Is ALTO centralized or Distributed?

ALTO server is a central resource.

Ref: J. Seedorf and E. Berger, "ALTO Problem Statement," [http://datatracker.ietf.org/doc/rfc5693/?include\\_text=1](http://datatracker.ietf.org/doc/rfc5693/?include_text=1)

Ref: Y. Lee, et al., "ALTO Extensions for collecting Data Center Resource Information,"

[http://datatracker.ietf.org/doc/draft-lee-alto-ext-dc-resource/?include\\_text=1](http://datatracker.ietf.org/doc/draft-lee-alto-ext-dc-resource/?include_text=1)

Washington University in St. Louis

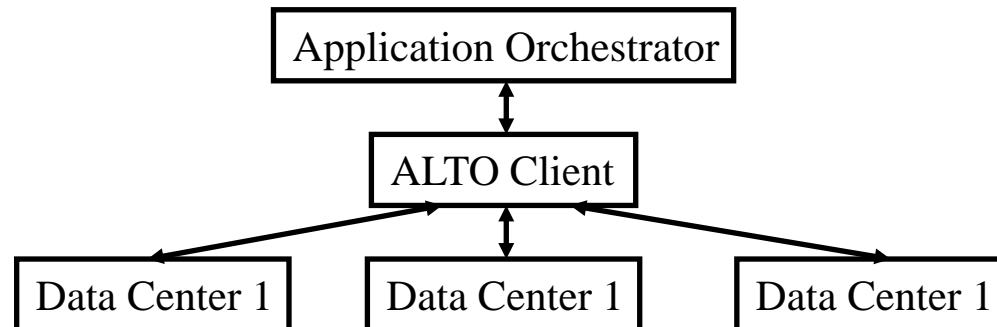
<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain



# ALTO Extension

- ❑ Now being extended to locate resources in data centers
- ❑ Need to be able to express
  - resource (memory, storage, CPU, network) availability
  - Cost of these resources
  - Constraints on resources, e.g., bandwidth
  - Constraints on structure, e.g., Power consumption
- ❑ ALTO client gets the info from various providers
- ❑ Issue of privacy of resource and cost info for the provider



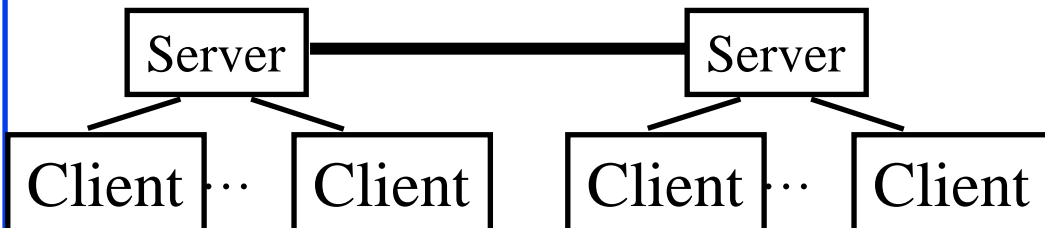
## Student Questions

- ❑ What do you mean by various providers?  
*Any resource (computing, storage, files, movies, etc.) may be available from many providers.*
- ❑ Is ALTO currently used?  
*It is widely used in content delivery networks, peer-to-peer networks, and cloud computing.*



# XMPP

- ❑ Extensible Messaging and Presence Protocol
- ❑ **Extensible** ⇒ Using XML
- ❑ Similar to SMTP email protocol but for near real-time communication
- ❑ Each client has an ID, e.g., [john@wustl.edu/mobile](mailto:john@wustl.edu/mobile) (John's mobile phone)
- ❑ Client sets up a connection with the server ⇒ Client is online
- ❑ **Presence**: The server maintains contact addresses and may let other contacts know that this client is now online
- ❑ **Messaging**: When a client sends a “chat” message to another client, it is forwarded to these other clients
- ❑ Messages are “*pushed*” (⇒ real-time) as opposed to “*polled*” as in SMTP/POP emails.



Ref: P. Saint-Andre, et al., “XMPP: The Definitive Guide,” O’Reilly, 2009, 320 pp., ISBN:9780596521264 (Safari Book)

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

## Student Questions

- ❑ Is ALTO similar to XMPP?  
*Not really. XMPP is like SMS between computers. ALTO is like Google search.*



**XMPP**

# XMPP (Cont)

- ❑ XMPP is the IETF standardization of Jabber protocol
- ❑ RFC 6121 defines XMPP using TCP connections.  
But HTTP is often used as transport to navigate firewalls
- ❑ All messages are XML-encoded  
⇒ Not efficient for binary file transfers  
⇒ Out-of-band binary channels are often used with XMPP.
- ❑ Several open-source implementations are available
- ❑ Variations of it are widely used in most instant messaging programs, including Google, Skype, Facebook, ..., many games
- ❑ Used in IoT and data centers for management. Network devices have XMPP clients that respond to XMPP messages containing CLI management requests ⇒ You can manage your network using any other XMPP client, e.g., your mobile phone
- ❑ XMPP can manage Arista switches, and Juniper uses XMPP as a southbound protocol for SDN

Ref: <http://en.wikipedia.org/wiki/XMPP>  
Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

## Student Questions

- ❑ What is the primary network structure of XMPP?

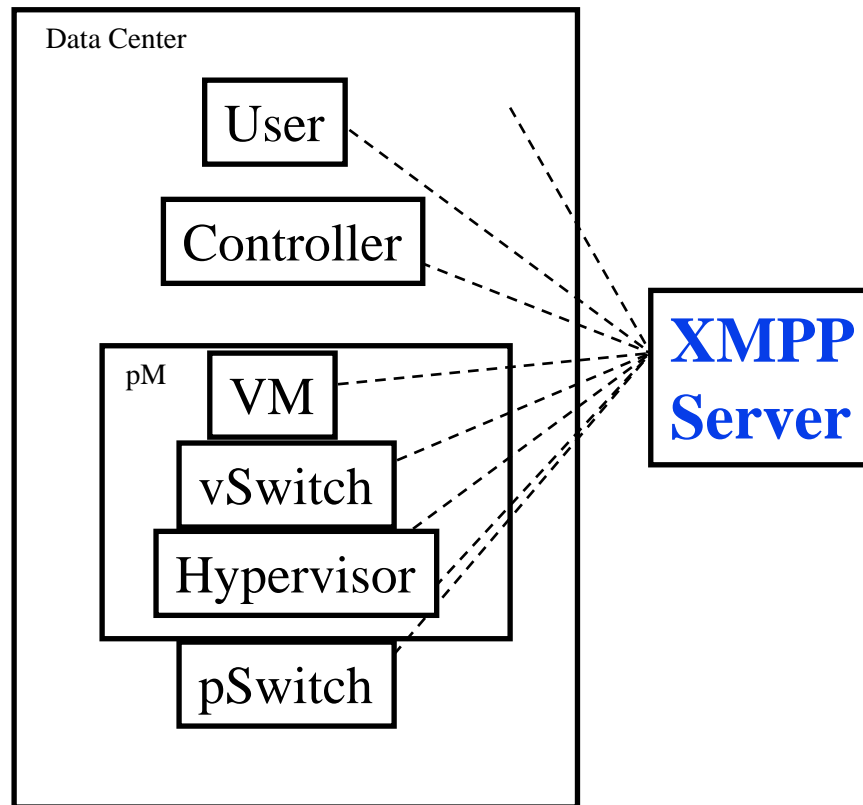
*See slide 15.28.*

- ❑ What advantages did XMPP have that made it attractive for game developers to use?

*Message passing.*

# XMPP in Data Centers

- Everything is an XMPP entity.  
It has its own contact list and authorizations.



## Student Questions

- Which Layer is XMPP on?  
*XMPP server is a controller. It is not a layer. It is a plane.*

# OpenDaylight Tools

1. **Application:** Provides Virtual Network Segments (VNS) for each tenant
  1. OpenDaylight Network Virtualization (ONV):
  2. OpenDaylight Virtual Tenant Network (VTN)
2. **Services:**
  1. Unified Secure Channel Manager
3. **Northbound APIs:**
  1. **REST:** Representational State Transfer (like HTTP)
  2. **RESTCONF:** RESTful Configuration
  3. **NETCONF:** Network Configuration
  4. **Dlux:** Northbound API using AngularJS, an extension of HTML by Google for dynamic views
  5. **AMQP:** Advanced Message Queuing Protocol

## Student Questions

- Can you refresh our memory on Northbound vs. Southbound in this context?

*See Q&A on slide 15.21*

---

# OpenDaylight Tools (Cont)

## 4. Southbound APIs:

1. OpenFlow Plug-in + Protocol Library (V1.0, V1.1,...)
2. Locator ID Separation Protocol (LISP) Mapping Service
3. SNMP4SDN
4. BGP Link State Path Control Element Protocol

## 5. Overlay:

1. Open Distributed Overlay Virtual Ethernet (DOVE):  
Like VxLAN but does not use IP Multicast

## 6. Configuration:

1. OpenDaylight YANG Tools: NETCONF
2. Open vSwitch Database (OVSDB) Integration

## Student Questions

# Open Network Linux (ONL)

- ❑ Linux distribution for “*open hardware*” bare metal switches
- ❑ Part of Open Compute Project
- ❑ Supports multiple *switch fabric APIs*:
  - **OF-DPA**: OpenFlow Data Plane Abstraction (API) for Broadcom chips
  - **OpenNSL**: Open Network Switch Layer for Broadcom switches
  - **SAI**: Switch Abstraction Interface (vendor-independent API to control forwarding elements)
- ❑ Compatible with many open-source *forwarding agents* or routing protocol suites

Ref: <http://opennetlinux.org/> , <https://github.com/Broadcom-Switch/OpenNSL>, <https://github.com/Broadcom-Switch/of-dpa>, <https://github.com/opencomputeproject/SAI>

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

## Student Questions

- ❑ ONL supports Broadcom SDK, which means it should use Broadcom ASICs. So, our ONL should use a bare metal switch that uses Broadcom ASICs. Broadcom also gives us OF-DPA and Open-NSL, and by using these two, we can interact with the SDK and ASIC. So, we cannot use other vendors' ASICs, and we have to use Broadcom products. So, what is the benefit of "openness"? Is the SDK or ASIC open? If not, we have to buy a complete package from Broadcom.

*Broadcom contributed its SDK, OF\_DPA, and OpenNSL to the Open Compute Project. They released their copyright to it. This allows any vendor to develop its hardware. The users do not have to worry about the hardware because they all provide the same interface. Many vendors have already implemented such switches, as indicated in Slide 15.32.*

- ❑ From the perspective of a company, doesn't disaggregation reduce monopoly power or profitability?

*Yes. This is good for the consumers.*

# Open Network Linux (ONL)

- ❑ Linux distribution for “*open hardware*” bare metal switches
- ❑ Part of Open Compute Project
- ❑ Supports multiple *switch fabric APIs*:
  - **OF-DPA**: OpenFlow Data Plane Abstraction (API) for Broadcom chips
  - **OpenNSL**: Open Network Switch Layer for Broadcom switches
  - **SAI**: Switch Abstraction Interface (vendor-independent API to control forwarding elements)
- ❑ Compatible with many open-source *forwarding agents* or routing protocol suites

Ref: <http://opennetlinux.org/>, <https://github.com/Broadcom-Switch/OpenNSL>, <https://github.com/Broadcom-Switch/of-dpa>, <https://github.com/opencomputeproject/SAI>

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

## Student Questions

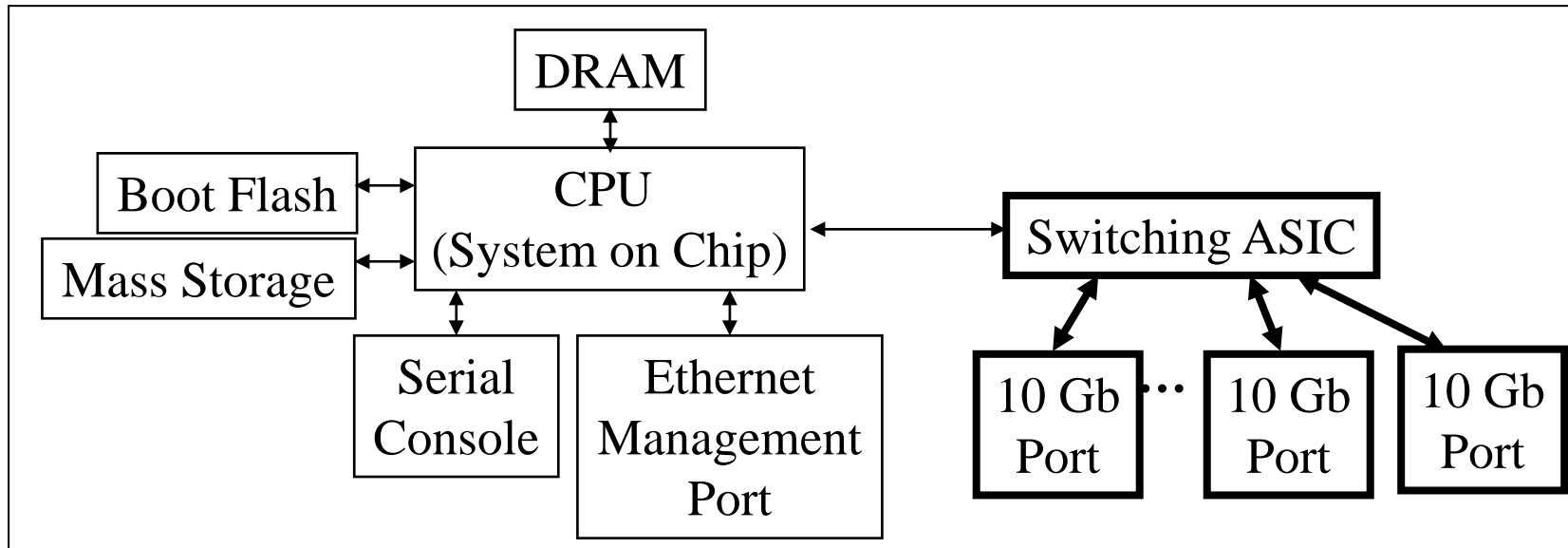
- ❑ Is ONL still widely used, and are Linux engineers needed? Why are the switches called bare metal when they have software?

*No software (other than bootloader and ONIE) comes when you buy them. Yes, Linux engineers are in demand.*



# Bare Metal Switches

- ❑ Hardware that can be used to load different network operating systems
- ❑ Open Network Linux is supported by hardware from: | Accton/Edge-Core, Quanta, Dell, Mellanox, Netberg, Inventec, Celestica, HPE, DNI, Ingrasys, and Alpha Networks



## Student Questions

- ❑ So regular switches could only support unique NOS?  
*Any switch that supports ONIE can be used to load different NOSs. See Slide 15.34*

# Open Source Forwarding Agents

- ❑ **Quagga**: A popular open-source routing software suite including OSPF, RIP, BGP, ...
- ❑ **FRRouting**: a fork of Quagga. Linux routing protocol suite including BGP, IS-IS, LDP, OSPF, PIM, and RIP (Free Range Routing?)
- ❑ **BIRD**: Internet Routing Daemon developed as a school project at Charles University, Prague. Supports IPv4, IPv6, BGP, RIP, OSPF, ...
- ❑ Facebook Open Switching System (**FBOSS**): S/w stack for controlling and managing network switches with several user-space applications
- ❑ Azure Software for Open Networking in the Cloud (**SONiC**)
- ❑ Google **gNOS**

Ref: <https://www.nongnu.org/quagga/>, <https://www.opensourcerouting.org/>, <http://bird.network.cz/>,  
<https://github.com/facebook/fboss>, <http://azure.github.io/SONiC/>

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

## Student Questions

- ❑ Can you explain the difference between Quagga and FRRouting more?

*These are two different software, similar to Linux and MAC.*

- ❑ I'm confused by the value Quagga provides. Aren't these protocols already available as open-source? Is it just that you can install one software package instead of multiple separately?

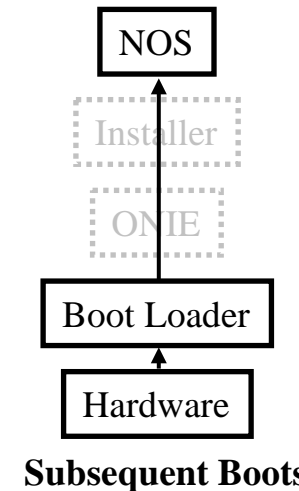
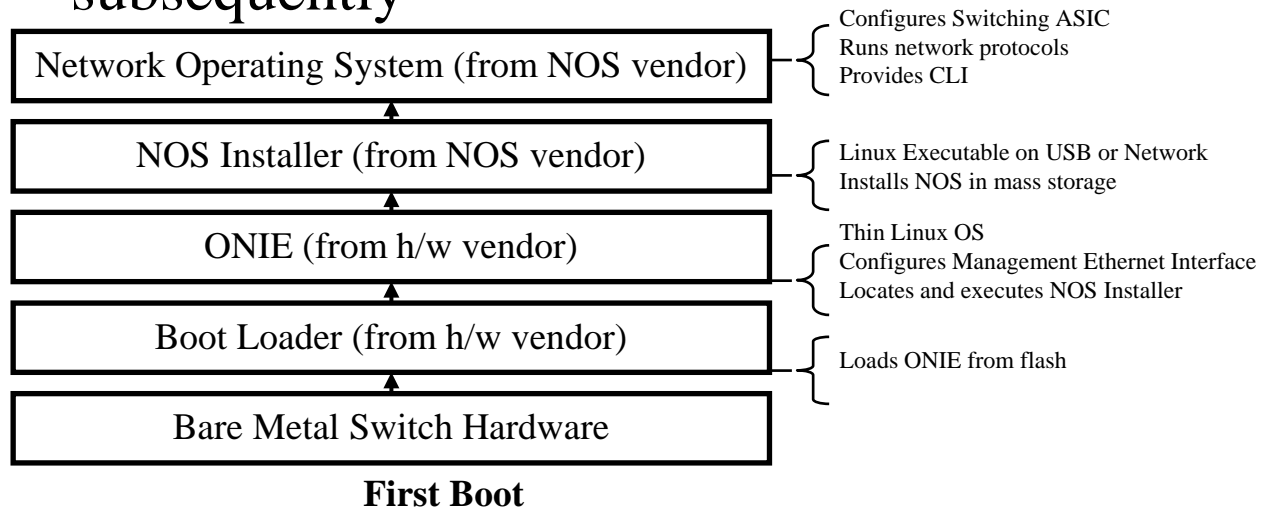
*Quagga is a complete interoperable suite with some missing pieces filled in.*

- ❑ ONL does not include TCP/IP suite as the default choice?

*ONL is a network operating system. You need to load protocols.*

# Open Network Install Environment (ONIE)

- ❑ Part of the **Open Compute Project (OCP)** open-source initiative
- ❑ Allows many different “**Network Operating Systems (NOS)**” on bare metal network switches
- ❑ Like a firmware that locates the NOS boot image and loads it
- ❑ ONIE sets the environment on the first boot and is not required subsequently



Ref: ONIE Overview, <https://opencomputeproject.github.io/onie/overview/index.html#onie-overview>

## Student Questions

- ❑ Can you please explain the figure on the left?
- Sure.*
- ❑ To double check, would we use ONIE to load Open Network Linux (a NOS) onto a bare metal switch?

*Yes, only on the first boot.*

- ❑ Could you explain more about what a forwarding agent does?

*The forwarding agent looks up a table to decide the action on a received packet.*

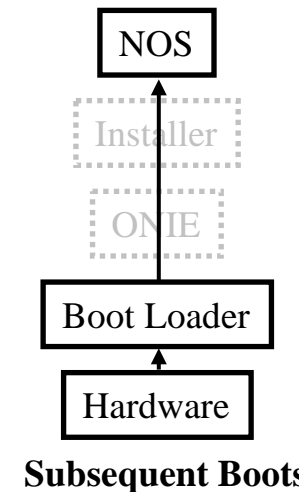
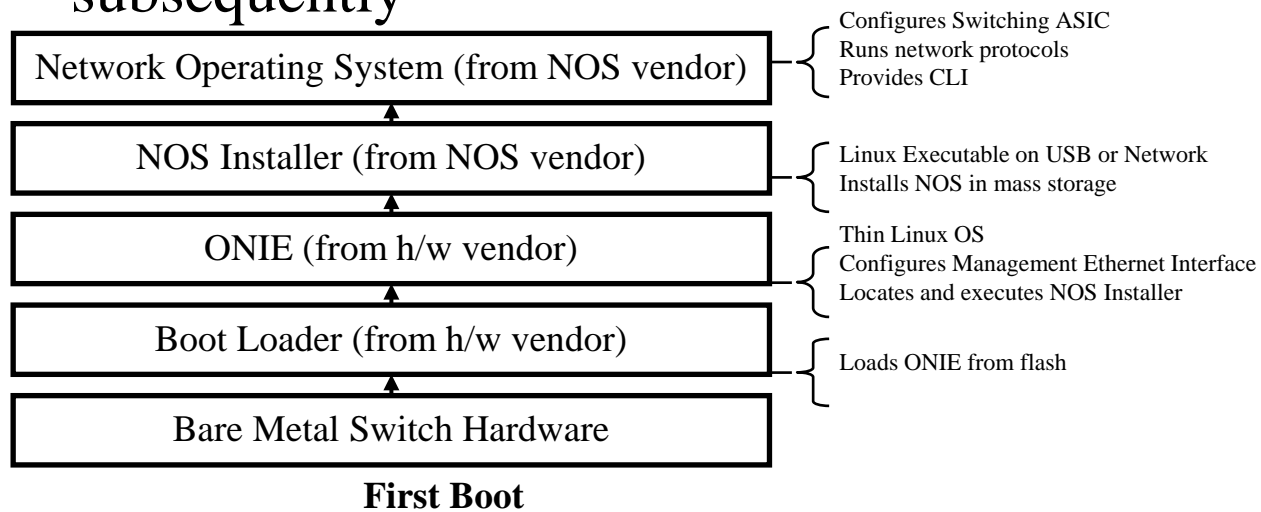
- ❑ Why do we need ONIE? Can't we install Open Network Linux (ONL) on Bare Metal Switches without ONIE?

*Without ONIE, the hardware and the software should come from the same vendor.*

- ❑ What does the OCP do?  
*It is a standards organization like IETF. Facebook initiated it.*

# Open Network Install Environment (ONIE)

- ❑ Part of the **Open Compute Project (OCP)** open-source initiative
- ❑ Allows many different “**Network Operating Systems (NOS)**” on bare metal network switches
- ❑ Like a firmware that locates the NOS boot image and loads it
- ❑ ONIE sets the environment on the first boot and is not required subsequently



Ref: ONIE Overview, <https://opencomputeproject.github.io/onie/overview/index.html#onie-overview>

## Student Questions

- ❑ Does the ONIE allow for faster subsequent boots?

*Yes. Subsequently, the software is all loaded automatically, like your MAC or Windows Laptop.*

- ❑ Are there any OSES specifically designed to support bare metal switches that would serve as many multiple virtual switches?

*Yes, similar to those listed earlier in Slide 15.33.*

- ❑ You said ONIE is only used once, but what if the environment changes? e.g., A new NOS is added to the support list?

*If the environment changes, you start from the forking point where the differences begin.*

# Mininet

- ❑ Widely used open-source network emulation environment.
- ❑ Can simulate several end-hosts, switches, routers, and links on a Linux
- ❑ Used for rapid prototyping of software-defined networks
- ❑ Built-in Open vSwitch and an OpenFlow capable switch
- ❑ Command line launcher and Python API for creating networks of varying sizes, e.g., *mn -topo tree,depth=2,fanout=3*
- ❑ Useful diagnostic commands like iperf, ping, and other commands in a host, e.g., *mininet> h11 ifconfig -a*
- ❑ Mininet codes for several popular commercial switches are available.

Ref: <https://github.com/mininet/mininet> , <http://www.brianlinkletter.com/set-up-mininet/>

## Student Questions

- ❑ Does Mininet run on physical switches (make pSwitches into vSwitches and emulate their connection) or run on computers (like a software application)?

*It runs on computers, PCs/Macs. Windows/Linux.*

- ❑ Will we do the Mininet lab you refer to in the video? If not, can you post the instructions for that lab somewhere in case we are interested?

*This year, we are not going to do it. See the course web page in 2018.*

<https://www.cse.wustl.edu/~jain/cse570-18/index.html>

- ❑ The slide mentioned a singular build in Open vSwitch and an OpenFlow capable switch. Is the idea of Mininet that you can simulate a whole network but get detailed statistics on one switch?

*You can get statistics on each switch in the simulated network.*

# RESTful APIs

- ❑ Software architecture style developed by W3C.
- ❑ Introduced by Roy Fielding in his Ph.D. thesis.
- ❑ WWW uses this style. Very popular in other applications.
- ❑ Goals: Scalability, Generality, Independence, and allowing intermediate components
- ❑ Client-Server Model: Clients and servers can be developed independently.
- ❑ Server is stateless
- ❑ Responses can be cached for the specified time
- ❑ Intermediate Servers (Proxies) can respond. The endpoint is not critical.

## Student Questions

# REST (Cont)

- ❑ Create, Read, Update, Delete (CRUD) Operations
- ❑ Uniform Interface: GET (Read), POST (Insert), PUT (write), DELETE
- ❑ Resources identified by global identifiers, e.g., URI in Web.
- ❑ Get `http://<fqdn-or-ip-address>/rest/v1/model/<data-type>/<optional-id>?<optional-query-params>`  
E.g., GET `http://odcp.org/rest/v1/model/controller-node`
- ❑ Data Types: Controller node, Firewall rule, Topology configuration, Switch, Port, link, flow entry, VLAN, ...
- ❑ Data types can include commercial entities, such as Big Virtual Switch from Big Switch Networks, vCenter from VMware, etc....
- ❑ If optional-id and query parameters are omitted, the returned text includes all of the items of the given data type.

Ref: [http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)

## Student Questions

- ❑ Does RESTful API allow Conditional GET requests?

*Yes, it is allowed as long as the condition does not require maintaining a state. For example, HTTP allows get if-not-modified-since 1-nov-2021.*

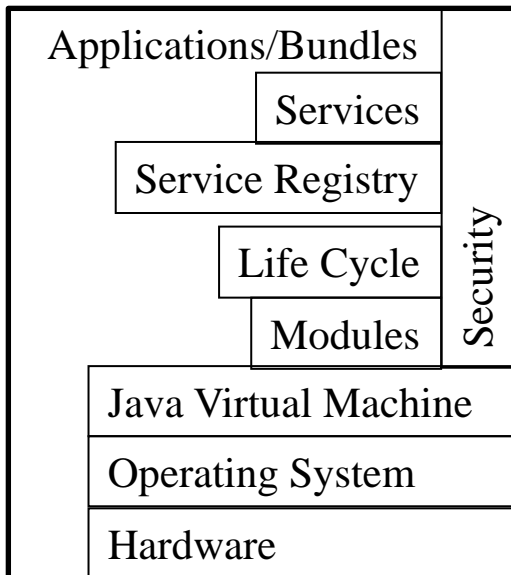
- ❑ What is the impact of APIs like GraphQL on REST? Are people transitioning away from REST?

*These are different. REST is a property. GraphQL is language. REST is not a standard. GraphQL is a standard initiated by Facebook.*

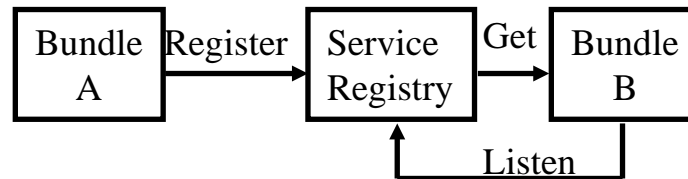
- ❑ What is the flow entry?  
*A row of OpenFlow table.*

# OSGi Framework

- ❑ Initially, **O**pen **S**ervices **G**ateway **I**nitiative
- ❑ A set of specifications for dynamic application composition using reusable Java components called bundles
- ❑ Bundles publish their services with the OSGi services registry and can find/use services of other bundles



Ref: <https://en.wikipedia.org/wiki/OSGi>



## Student Questions

- ❑ What kind of communication is required to find services?

*OSGi specifies an API.*

- ❑ Doesn't the OSGi framework complicate the system to some degree?

*No. It is just a few commands at the start of a service. The results can be cached and refreshed only if needed.*

- ❑ Do RESTful APIs use minimal computing and system resources?

*Not necessarily. They minimize state, which is helpful to keep scalability and reliability in case of failures.*

- ❑ Is OSGi a service or a framework?

*Framework.*

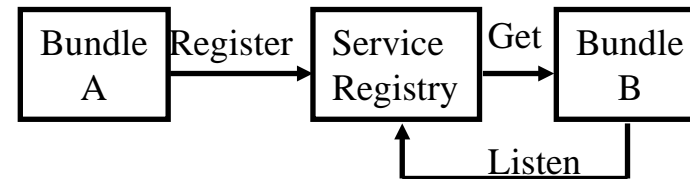
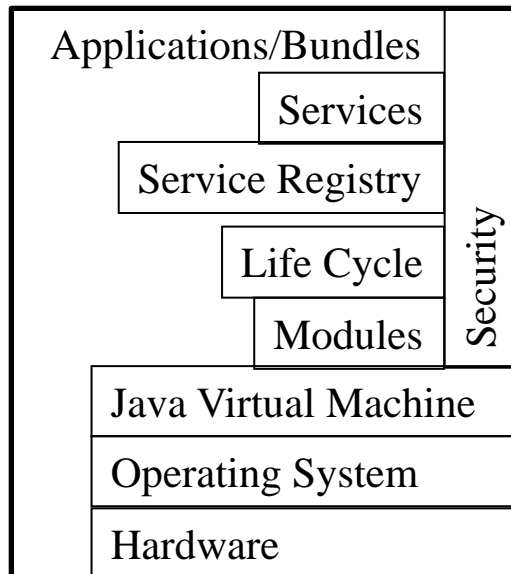
- ❑ Is the client like a Bundle in the OSGi framework?

*Yes, both clients and servers can offer some services and use other services so they can be bundles.*



# OSGi Framework

- ❑ Initially, **O**pen **S**ervices **G**ateway **I**nitiative
- ❑ A set of specifications for dynamic application composition using reusable Java components called bundles
- ❑ Bundles publish their services with the OSGi services registry and can find/use services of other bundles



Ref: <https://en.wikipedia.org/wiki/OSGi>

## Student Questions

- ❑ What is the benefit or purpose of using OSGi?  
*Allows reusing modules and module libraries.*

# OSGi (Cont)

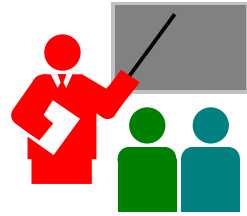
- ❑ Bundles can be installed, started, stopped, updated, or uninstalled using a lifecycle API
  - ❑ Modules define how a bundle can import/export code
  - ❑ The security layer handles security
  - ❑ The execution environment defines what methods and classes are available in a specific platform
  - ❑ A bundle can get a service, or it can listen for a service to appear or disappear.
  - ❑ Each service has properties that allow others to select among multiple bundles offering the same service
  - ❑ Services are dynamic. A bundle can decide to withdraw its service. Other bundles should stop using it
- ⇒ Bundles can be installed and uninstalled on the fly.

## Student Questions

- ❑ In what applications is the OSGi framework deployed (for example, within a network)?

*See Slide 15.19.*

# Summary



1. SDN = Abstraction + Programmability + Centralization  
SDN = Disaggregation of h/w and s/w  
= Bare metal switches + ONIE + ONL
2. OpenFlow originated SDN, but now many different southbound and northbound APIs, intermediate services, and tools are being discussed and implemented by the industry, e.g., XMPP, PCE, ALTO
3. OpenDaylight and ONOS are SDN Controllers.  
Differ on how much is open.
4. Mininet for network simulation
5. REST=HTTP APIs  
OSGI framework for modularity

## Student Questions

- Having a mind map for the SDN module would be very helpful!

*Sure. Will try.*

---

# Reading List

- ❑ Thomas D. Nadeau, Ken Gray, "SDN: Software Defined Networks," O'Reilly Media, Inc., August 2013, 384 pp., ISBN:978-1-4493-4230-2 (Safari Book).
- ❑ Jim Doherty, "SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization," Addison-Wesley Professional, March 2, 2016, 320 pp., ISBN:978-0-13-430739-8 (Safari Book).
- ❑ Reza Toghraee, "Learning OpenDaylight," Packt Publishing, May 2017, 336 pp., ISBN:978-1-78217-452-3 (Safari Book).
- ❑ Antonio Sanchez Monge; Krzysztof Grzegorz Szarkowicz, "MPLS in the SDN Era," O'Reilly Media, Inc., December 2015, 920 pp., ISBN:978-1-4919-0545-6 (Safari Book).

## Student Questions

# Wikipedia Links

- ❑ [http://en.wikipedia.org/wiki/Software-defined\\_networking](http://en.wikipedia.org/wiki/Software-defined_networking)
- ❑ [http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)
- ❑ <http://en.wikipedia.org/wiki/OSGI>
- ❑ <http://en.wikipedia.org/wiki/XMPP>
- ❑ [http://en.wikipedia.org/wiki/Path\\_computation\\_element](http://en.wikipedia.org/wiki/Path_computation_element)

## Student Questions

# References

- ❑ J. Seedorf and E. Berger, “ALTO Problem Statement,” [http://datatracker.ietf.org/doc/rfc5693/?include\\_text=1](http://datatracker.ietf.org/doc/rfc5693/?include_text=1)
- ❑ Y. Lee, et al., “ALTO Extensions for collecting Data Center Resource Information,” [http://datatracker.ietf.org/doc/draft-lee-alto-ext-dc-resource/?include\\_text=1](http://datatracker.ietf.org/doc/draft-lee-alto-ext-dc-resource/?include_text=1)
- ❑ <http://datatracker.ietf.org/wg/pce/>
- ❑ <http://events17.linuxfoundation.org/sites/events/files/slides/OpenDaylight-Network-Programmability.pdf>
- ❑ OpenDaylight Components and Tools, <https://wiki.opendaylight.org>

## Student Questions

## References (Cont)

- ❑ Oswald Coker, Siamak Azodolmolky, "Software-Defined Networking with OpenFlow - Second Edition," Packt Publishing, October 2017, 246 pp., ISBN:978-1-78398-429-9 (Safari Book).
- ❑ William Stallings, "Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud," Addison-Wesley Professional, October 2015, 544 pp., ISBN:0-13-417539-5 (Safari Book).
- ❑ Russ White, Jeff Tantsura, "Navigating Network Complexity: Next-generation Routing with SDN, Service Virtualization, and Service Chaining," Addison-Wesley Professional, November 2015, 320 pp., ISBN:0-13-398792-2 (Safari Book).
- ❑ Abhishek Ratan, "Practical Network Automation," Packt Publishing, November 2017, 266 pp., ISBN:978-1-78829-913-8 (Safari Book).
- ❑ Scott S. Lowe, Matt Oswalt, Jason Edelman, "Network Programmability and Automation," O'Reilly Media, Inc., February 2018, 581 pp., ISBN:978-1-4919-3125-7 (Safari Book).
- ❑ Guy Pujolle, "Software Networks," John Wiley & Sons, August 2015, 260 pp., ISBN:978-1-119-00796-8 (Safari Book).

## Student Questions

## References (Cont)

- ❑ Sriram Subramanian, Sreenivas Voruganti, "Software-Defined Networking (SDN) with OpenStack," Packt Publishing, October 2016, 216 pp., ISBN:978-1-78646-599-3 (Safari Book).
- ❑ Paul Goransson, Chuck Black, Timothy Culver, "Software Defined Networks, 2nd Edition," Morgan Kaufmann, October 2016, 436 pp., ISBN:978-0-12-804555-8 (Safari Book).
- ❑ Patricia A. Morreale, James M. Anderson, "Software Defined Networking," CRC Press, September 2015, 186 pp., ISBN:1-4822-3864-0 (Safari Book).
- ❑ V. Josyula, M. Orr, and G. Page, "Cloud Computing: Automating the Virtualized Data Center," Cisco Press, 2012, 392 pp., ISBN: 1587204347 (Safari Book).
- ❑ P. Saint-Andre, et al., "XMPP: The Definitive Guide," O'Reilly, 2009, 320 pp., ISBN:9780596521264 (Safari Book)

## Student Questions



# Acronyms

- ❑ ACI            Application Policy Infrastructure
- ❑ ACL            Access Control List
- ❑ AEX            Application Information Exposure
- ❑ ALG            Application Level Gateway
- ❑ ALTO           Application-Layer Traffic Optimization
- ❑ AMQP           Advanced Message Queueing Protocol
- ❑ ANDSF           Access Network Discovery and Selection Function
- ❑ API            Application Programming Interface
- ❑ APIC            Application Policy Infrastructure Controller
- ❑ ARP            Address Resolution Protocol
- ❑ ATIS            Association for Telecom Industry Solutions
- ❑ ATM            Asynchronous Transfer Mode
- ❑ AVNP            Active Virtual Network Management Protocol
- ❑ BGP            Border Gateway Protocol
- ❑ BNC            Big Switch Network Controller
- ❑ BSD            Berkeley Software Distribution

## Student Questions

# Acronyms (Cont)

- ❑ BUM Broadcast, Unknown, and Multicast
- ❑ CDN Content Distribution Network
- ❑ CDNI Content Distribution Network Interconnection
- ❑ CE Control Element
- ❑ CLI Command Line Interface
- ❑ CMS Content Management System
- ❑ CPU Central Processing Unit
- ❑ CRUD Create, Read, Update, Delete
- ❑ CSP Cloud Service Provider
- ❑ DHCP Dynamic Host Control Protocol
- ❑ DNS Domain Name System
- ❑ DOCSIS Data over Cable Service Interface Specification
- ❑ DOVE Distributed Overlay Virtual Ethernet
- ❑ DVS Distributed Virtual Switch
- ❑ EID Endpoint Identifier
- ❑ ETSI European Telecommunications Standards Institute

## Student Questions

# Acronyms (Cont)

- ❑ FCAPS      Faults, configuration, accounting, performance , and security
- ❑ FE          Forwarding Element
- ❑ FE          Forwarding Element
- ❑ ForCES     Forwarding and Control Element Separation
- ❑ GMPLS     Generalized Multi-Protocol Label Switching
- ❑ GUI        Graphical User Interface
- ❑ HTML      Hypertext Markup Language
- ❑ HTTP      Hypertext Transfer Protocol
- ❑ I2AEX      Infrastructure to Application Information Exposure
- ❑ IaaS        Infrastructure as a Service
- ❑ ID          Identifier
- ❑ IDS        Intrusion Detection System
- ❑ IEEE      Institution of Electrical and Electronic Engineers
- ❑ IETF      Internet Engineering Task Force
- ❑ IGP        Interior Gateway Protocol
- ❑ IoT        Internet of Things

## Student Questions

# Acronyms (Cont)

- ❑ IP Internet Protocol
- ❑ IPv4 Internet Protocol version 4
- ❑ IPv6 Internet Protocol version 6
- ❑ IRTF Internet Research Taskforce
- ❑ IS-IS Intermediate System to Intermediate System
- ❑ ISO International Standards Organization
- ❑ L2 Layer 2
- ❑ LACP Link Aggregation Control Protocol
- ❑ LAN Local Area Network
- ❑ LISP Locator-ID Separation Protocol
- ❑ LS Link State
- ❑ MAC Media Access Control
- ❑ MPLS Multi-protocol Label Switching
- ❑ NAT Network Address Translation
- ❑ NetIDE Network Interactive Development Environment
- ❑ NEMO File Manager for Linux Distribution

## Student Questions

# Acronyms (Cont)

- ❑ NFV Network Function Virtualization
- ❑ NTP Network Time Protocol
- ❑ NVGRE Network Virtualization using Generic Routing Encapsulation
- ❑ NVO3 Network Virtualization over L3
- ❑ NVP Network Virtualization Platform
- ❑ OF OpenFlow
- ❑ OnePK Open Network Environment Platform Kit
- ❑ ONF Open Networking Forum
- ❑ ONV OpenDaylight Network Virtualization
- ❑ OpEx Operational Expenses
- ❑ OS Operating System
- ❑ OSCP OpenDaylight SDN Controller Platform
- ❑ OSGi Open Services Gateway Initiative
- ❑ OSPF Open Shortest Path First
- ❑ OVS Open Virtual Switch
- ❑ OVSDB Open Virtual Switch Database

## Student Questions

# Acronyms (Cont)

- ❑ PCC Path Computation Client
- ❑ PCE Path Computation Element
- ❑ PCEP Path Computation Element Protocol
- ❑ POP Post Office Protocol
- ❑ PWE3 Pseudowire Emulation Edge to Edge
- ❑ QoS Quality of Service
- ❑ REST Representational State Transfer
- ❑ RFC Request for Comments
- ❑ RLOC Routing Locator
- ❑ RLOC Routing Locator
- ❑ RS Routing System
- ❑ SAL Service Abstraction Layer
- ❑ SDN Software Defined Networking
- ❑ SMTP Simple Mail Transfer Protocol
- ❑ SNMP Simple Network Management Protocol

## Student Questions

# Acronyms (Cont)

- ❑ SNMP4SDN    SNMP for SDN
- ❑ SSH        Secure Socket Host
- ❑ STT        Stateless TCP-like Transport
- ❑ TCP        Transmission Control Protocol
- ❑ TE         Traffic Engineering
- ❑ TIA        Telecom Industry Association
- ❑ TRILL     Transparent Interconnection of Lots of Links
- ❑ URI        Uniform Resource Identifier
- ❑ vBridge    Virtual Bridge
- ❑ VIRL      Virtual Internet Routing Lab
- ❑ VLAN      Virtual Local Area Network
- ❑ VM        Virtual Machine
- ❑ VNS        Virtual Network Segement
- ❑ VPN        Virtual Private Network
- ❑ vTep      Virtual Tunnel End Point
- ❑ VTN        Virtual Tenant Network

## Student Questions

# Acronyms (Cont)

- ❑ VxLAN Virtual Extensible Local Area Network
- ❑ WAN Wide Area Network
- ❑ XML Extensible Markup Language
- ❑ XMPP Extensible Messaging and Presence Protocol

## Student Questions



# Style Guide

Correct	Incorrect
ACL	Acl or acl
API	api
ARP	Arp or arp
IPsec	IPSEC or ipsec
IPv4 or IPv6	Ipv4, Ipv6, ipv4, ipv6, IPV4, or IPV6
Karaf	karaf
Linux	LINUX or linux
NETCONF	Netconf or netconf
Neutron	neutron
OSGi	osgi or OSGI
Open vSwitch	OpenvSwitch, OpenVSwitch, or Open V Switch.
OpenDaylight	Opendaylight, Open Daylight, or OpenDayLight.
OpenFlow	Openflow, Open Flow, or openflow.
OpenStack	Open Stack or Openstack
QoS	Qos, QOS, or qos
RESTCONF	Restconf or restconf
RPC	Rpc or rpc
URL	Url or url

## Student Questions

Source: <http://docs.opendaylight.org/en/stable-nitrogen/documentation.html>

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

# SDN Related Organizations and Projects

- ❑ Linux Foundation Open Source Networking, <https://www.linuxfoundation.org/projects/networking/>
- ❑ Open Networking Foundation (ONF): [www.opennetworking.org](http://www.opennetworking.org)
- ❑ Telecom Industry Association (TIA): [www.tiaonline.org](http://www.tiaonline.org)
- ❑ European Telecommunications Standards Institute (ETSI): [www.etsi.org/](http://www.etsi.org/)
- ❑ Association for Telecom Industry Solutions (ATIS): [www.atis.org/topsc/sdn.asp](http://www.atis.org/topsc/sdn.asp)
- ❑ Internet Engineering Task Force (IETF): [www.ietf.org](http://www.ietf.org)
- ❑ OpenStack Quantum: <https://wiki.openstack.org/wiki/Quantum>
- ❑ OpenDaylight: [www.opendaylight.org](http://www.opendaylight.org)

## Student Questions

# SDN Web Sites

- HotSDN 2012, <http://yuba.stanford.edu/~casado/of-sw.html>  
(Papers downloadable)

## Student Questions

**Scan This to Download These Slides**



Raj Jain

<http://rajjain.com>

[http://www.cse.wustl.edu/~jain/cse570-23/m\\_15sdn.htm](http://www.cse.wustl.edu/~jain/cse570-23/m_15sdn.htm)

Washington University in St. Louis

<http://www.cse.wustl.edu/~jain/cse570-23/>

©2023 Raj Jain

**Student Questions**

# Related Modules



CSE567M: Computer Systems Analysis (Spring 2013),

[https://www.youtube.com/playlist?list=PLjGG94etKypJEKjNAa1n\\_1X0bWWNyZcof](https://www.youtube.com/playlist?list=PLjGG94etKypJEKjNAa1n_1X0bWWNyZcof)

CSE473S: Introduction to Computer Networks (Fall 2011),

[https://www.youtube.com/playlist?list=PLjGG94etKypJWOSPMh8Azcg5e\\_10TiDw](https://www.youtube.com/playlist?list=PLjGG94etKypJWOSPMh8Azcg5e_10TiDw)



Wireless and Mobile Networking (Spring 2016),

[https://www.youtube.com/playlist?list=PLjGG94etKypKeb0nzyN9tSs\\_HCd5c4wXF](https://www.youtube.com/playlist?list=PLjGG94etKypKeb0nzyN9tSs_HCd5c4wXF)

CSE571S: Network Security (Fall 2011),

<https://www.youtube.com/playlist?list=PLjGG94etKypKvzfVtutHcPFJXumyyg93u>



Video Podcasts of Prof. Raj Jain's Lectures,

<https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw>

## Student Questions