Introduction to Software Defined Networking (SDN)

SDN = Separation of Control and Data Planes

SDN = OpenFlow

SDN = Centralization of control plane

SDN = Standard Southbound API

SDN = Separation of Control and Data Planes

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/cse570-19/
Overview

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.
Origin 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 significantly.
ONOS

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

Ref: ONOS Architecture, [https://docs.google.com/presentation/d/1Y4S82YZyqsknKAzW4kKm-6llo04h_nnrRQyfrsCdt-I/edit?usp=sharing](https://docs.google.com/presentation/d/1Y4S82YZyqsknKAzW4kKm-6llo04h_nnrRQyfrsCdt-I/edit?usp=sharing)

Washington University in St. Louis

©2019 Raj Jain
OpenDaylight SDN Controller Platform (OSCP)

- Multi-company collaboration under Linux foundation
- Many projects including OpenDaylight Controller
- Supports multiple southbound protocols via plug-ins including OpenFlow
- Dynamically linked in to a Service Abstraction Layer (SAL) Abstraction ⇒ 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

https://wiki.opendaylight.org/view/Main_Page
RESTful APIs

- Software architecture style developed by W3C.
- Introduced by Roy Fielding in his PhD thesis.
- WWW uses this style. Very popular in other applications.
- Goals: Scalability, Generality, Independence, and allow 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. End point is not critical.
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.
  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, …
- 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
OSGi Framework

- Initially, Open Services Gateway initiative
- A set of specifications for dynamic application composition using reusable Java components called bundles
- Bundles publish their services with OSGi services registry and can find/use services of other bundles

Ref: [http://www.osgi.org/Technology/WhatIsOSGi](http://www.osgi.org/Technology/WhatIsOSGi)

Washington University in St. Louis

http://www.cse.wustl.edu/~jain/cse570-19/

©2019 Raj Jain
OSGi (Cont)

- Bundles can be installed, started, stopped, updated or uninstalled using a lifecycle API
- Modules defines how a bundle can import/export code
- Security layer handles security
- 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
  \[ \Rightarrow \] Bundles can be installed and uninstalled on the fly.
Centralized vs. Distributed

- Fast Response to changes
- Fast Consistency
- Less overhead $\Rightarrow$ Scalable
- Single Point of Failure

- Time to converge
- Slow consistency
- Not scalable
- Fault Tolerant
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 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 large infrastructure, need scalable solutions.
Separation vs. Centralization

Separation of Control Plane

Centralization of Policies

Micromanagement is not scalable
What SDN is Not?

- All of these are mechanisms.
- SDN is *not* about a mechanism.
- It is a framework ⇒ Many solutions

SDN = OpenFlow
SDN = Standard Southbound API
SDN = Centralization of control plane
SDN = Separation of Control and Data Planes
Trend 1: SDN to Disaggregation

SDN was invented in 2009. Then: SDN:
- Separation of control and data planes
- Centralization of Control
- Standard Protocol between the planes

203 Papers on OpenFlow on IEEEExplore in 2018!

Now: Software Defined = **Disaggregation** of HW/SW
- Commodity hardware
- Software on commodity HW
- Legacy protocols survive


©2019 Raj Jain
Disaggregation: Black Box to White Box

- Differentiation via software ⇒ White box networking
- **Black Box**: Proprietary HW with Proprietary SW
- **White Box**: Open Source Hardware and Software
- Software on a different hardware ⇒ hardware can change
  Different software on a hardware ⇒ Software can change

- **Bright Box**: Branded White box =
  Branded SW on open HW or Open SW on Branded HW


Washington University in St. Louis http://www.cse.wustl.edu/~jain/cse570-19/ ©2019 Raj Jain
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

Ref: Hardware Support and Certification, https://opennetlinux.org/hcl.html
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**


Washington University in St. Louis  [http://www.cse.wustl.edu/~jain/cse570-19/](http://www.cse.wustl.edu/~jain/cse570-19/)
Open Network Linux

- 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

Open Network Install Environment (ONIE)

- Part of 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
Washington University in St. Louis, http://www.cse.wustl.edu/~jain/cse570-19/
Mininet

- Widely used open source network emulation environment.
- Can simulate a number of end-hosts, switches, routers, links on a Linux
- Used for rapid prototyping of software define networks
- Built-in Open vSwitch, and a 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 code for several popular commercial switches are available.

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,

3. OpenDaylight and ONOS are SDN Controllers. Differ on how much open.

4. Mininet for network simulation

5. REST=HTTP APIs
OSGI framework for modularity
Reading List

Wikipedia Links

References

References (Cont)


References (Cont)


References (Cont)

- http://www.osgi.org/Technology/WhatIsOSGi
- http://www.sdncentral.com/sdn-use-cases/
- https://wiki.opendaylight.org/view/Main_Page
<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTO</td>
<td>Application Layer Traffic Optimization</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>APIC</td>
<td>Application Policy Infrastructure Controller</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuit</td>
</tr>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>CDNI</td>
<td>Content Distribution Network Interconnection</td>
</tr>
<tr>
<td>CLI</td>
<td>Command Line Interface</td>
</tr>
<tr>
<td>COPS</td>
<td>Common Open Policy Service</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CRUD</td>
<td>Create, Read, Update, Delete</td>
</tr>
<tr>
<td>DLUX</td>
<td>OpenDaylight User Interface</td>
</tr>
<tr>
<td>DNI</td>
<td>Name of a company</td>
</tr>
<tr>
<td>FE</td>
<td>Forwarding Element</td>
</tr>
<tr>
<td>ForCES</td>
<td>Forwarding and Control Element Separation</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
</tbody>
</table>
Acronyms (Cont)

- HW Hardware
- IaaS Infrastructure as a Service
- IDS Intrusion Detection System
- IGP Interior Gateway Protocol
- IoT Internet of Things
- IP Internet Protocol
- IPv4 Internet Protocol version 4
- IPv6 Internet Protocol version 6
- ISO International Standards Organization
- L2 Layer 2
- LDP Label Distribution Protocol
- LS Link State
- MPLS Multi-protocol Label Switching
- NetIDE Network Interactive Development Environment
- NETCONF Network Configuration
### Acronyms (Cont)

- **NFV**: Network Function Virtualization
- **NOS**: Network Operating System
- **NVGRE**: Network Virtualization using Generic Routing Encapsulation
- **OF**: OpenFlow
- **OnePK**: Open Network Environment Platform Kit
- **ONF**: Open Networking Forum
- **ONiE**: Open Network Install Engine
- **ONL**: Open Net Linux
- **ONOS**: Open Networking Operating System
- **ONV**: OpenDaylight Network Virtualization
- **OS**: Operating System
- **OSCP**: OpenDaylight SDN Controller Platform
- **OSGi**: Open Services Gateway Initiative
- **OSPF**: Open Shortest Path First
- **OVS**: Open Virtual Switch
- **OVSD**: Open Virtual Switch Database
Acronyms (Cont)

- PCE  Path Computation Element
- PCEP  Path Computation Element Protocol
- PIM  Protocol Independent Multicast
- QoE  Quality of Experience
- QoS  Quality of Service
- REST  Representational State Transfer
- RIP  Routing Information Protocol
- SAL  Service Abstraction Layer
- SAI  Serial Audio Interface
- SDN  Software Defined Networking
- SNMP4SDN  SNMP for SDN
- SW  Software
- TIA  Telecom Industry Association
- TRILL  Transparent Interconnection of Lots of Links
- URI  Uniform Resource Identifier
Acronyms (Cont)

- **USB** Universal Serial Bus
- **VLAN** Virtual Local Area Network
- **VxLAN** Virtual Extensible Local Area Network
- **WAN** Wide Area Network
- **WWW** World Wide Web
- **XMPP** Extensible Messaging and Presence Protocol
# Style Guide

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

SDN Related Organizations and Projects

- Linux Foundation Open Source Networking, [https://www.linuxfoundation.org/projects/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)
- Internet Engineering Task Force (IETF): [www.ietf.org](http://www.ietf.org)
- OpenStack Quantum: [https://wiki.openstack.org/wiki/Quantum](https://wiki.openstack.org/wiki/Quantum)
- OpenDaylight: [www.opendaylight.org](http://www.opendaylight.org)
SDN Web Sites

- SDN Central, http://www.sdncentral.com
Related Modules

CSE567M: Computer Systems Analysis (Spring 2013),
https://www.youtube.com/playlist?list=PLjGG94etKypJEKjNAa1n_1X0bWWNyZcof

CSE473S: Introduction to Computer Networks (Fall 2011),
https://www.youtube.com/playlist?list=PLjGG94etKypJWOSPMh8Azcgy5e_10TiDw

Wireless and Mobile Networking (Spring 2016),
https://www.youtube.com/playlist?list=PLjGG94etKypKeb0nzyN9tSs_HCd5c4wXF

CSE571S: Network Security (Fall 2011),
https://www.youtube.com/playlist?list=PLjGG94etKypKvzfVtutHcPFJXumyyyg93u

Video Podcasts of Prof. Raj Jain's Lectures,
https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQM8s-8NUw