Introduction to Software Defined Networking (SDN)

SDN = Separation of Control and Data Planes
SDN = Centralization of control plane
SDN = Standard Southbound API
SDN = OpenFlow

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/cse570-13/
1. What is SDN?
2. Alternative APIs: XMPP, PCE, ForCES, ALTO
3. RESTful APIs and OSGi Framework
4. OpenDaylight SDN Controller Platform and Tools

Note: This is the third module of four modules on OpenFlow, OpenFlow Controllers, SDN and NFV in this course.
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 significantly.
What is SDN?

- All of these are mechanisms.
- SDN is *not* a mechanism.
- It is a framework to solve a set of problems ⇒ Many solutions

SDN = Separation of Control and Data Planes
SDN = OpenFlow
SDN = Standard Southbound API
SDN = Centralization of control plane
Original Definition of SDN

“What is SDN?

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

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

The above definition includes *How*.

Now many different opinions about *How*.

⇒SDN has become more general. Need to define by *What*?

What = Why We need SDN?

1. **Virtualization**: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.

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 size, quantity

5. **Automation**: To lower OpEx minimize manual involvement
   - Troubleshooting
   - Reduce downtime
   - Policy enforcement
   - Provisioning/Re-provisioning/Segmentation of resources
   - Add new workloads, sites, devices, and resources
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, and routing, security

9. **Service Integration**: Load balancers, firewalls, Intrusion Detection Systems (IDS), provisioned on demand and placed appropriately on the traffic path
10. **Openness**: Full choice of “How” mechanisms
   ⇒ Modular plug-ins
   ⇒ Abstraction:
     - Abstract = Summary = Essence = General Idea
       ⇒ Hide the details.
     - Also, abstract is opposite of concrete
       ⇒ Define tasks by APIs and **not by how** it should be done.
       E.g., send from A to B. Not OSPF.

Ref: Open Data Center Alliance Usage Model: Software Defined Networking Rev 1.0,”
SDN Definition

- SDN is a *framework* to allow network administrators to *automatically* and dynamically manage and control a *large number* of network devices, *services*, topology, traffic paths, and packet handling (quality of service) policies using high-level languages and APIs. Management includes provisioning, operating, *monitoring*, optimizing, and managing FCAPS (faults, configuration, accounting, *performance*, and security) in a *multi-tenant* environment.

- Key: Dynamic ⇒ Quick
  Legacy approaches such as CLI were not quick particularly for large networks
Examples Alternative APIs

- Southbound APIs: XMPP (Juniper), OnePK (Cisco)
- Northbound APIs: I2RS, I2AEX, ALTO,
- Overlay: VxLAN, TRILL, LISP, STT, NVO3, PWE3, L2VPN, L3VPN
- Configuration API: NETCONF
- Controller: PCE, ForCES

Washington University in St. Louis  http://www.cse.wustl.edu/~jain/cse570-13/ ©2013 Raj Jain
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 (John’s mobile phone)
- Client sets up a connection with the server ⇒ Client is online
- **Presence**: Server maintains contact addresses and may let other contacts know that this client is now on-line
- **Messaging**: When a client sends a “chat” message to another clients, it is forwarded to these other clients
- Messages are “pushed” (⇒ real-time) as opposed to “polled” as in SMTP/POP emails.

Washington University in St. Louis  
http://www.cse.wustl.edu/~jain/cse570-13/  
©2013 Raj Jain
XMPP is 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.
- A number of 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
- Arista switches can be managed by XMPP, Juniper uses XMPP as a southbound protocol for SDN

XMPP in Data Centers

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

Ref: https://github.com/ArchipelProject/Archipel/wiki/Architecture-%26-Concepts
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 computer power or network knowledge than a router may have.
- 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 path computation element (PCE)
- PCE may be centralized or may be distributed in many or every router.

What is the 1 Gbps route to New York not going through Boston?
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/](http://datatracker.ietf.org/wg/pce/)
Ref: [http://en.wikipedia.org/wiki/Path_computation_element](http://en.wikipedia.org/wiki/Path_computation_element)
Forwarding and Control Element Separation (ForCES)

- IETF working group since July 2001
- Control Elements (CEs) prepare the routing table for use by forwarding elements (FEs).
- Each CE may interact with one or more FEs
- There may be many CEs and FEs managed by a CE manager and a FE manager
ForCES (Cont)

- Idea of control and data plane separation was used in BSD 4.4 routing sockets in early 1990s. It allowed routing tables to be controlled by a simple command line or by a route daemon.

- ForCES protocol supports exchange of:
  - Port type, link speed, IP address
  - IPv4/IPv6 unicast/multicast forwarding
  - QoS including metering, policing, shaping, and queueing
  - Packet classification
  - High-touch functions, e.g., Network Address Translation (NAT), Application-level Gateways (ALG)
  - Encryptions to be applied to packets
  - Measurement and reporting of per-flow traffic information

Ref: http://datatracker.ietf.org/doc/rfc3654/?include_text=1
Sample ForCES Exchanges

Security exchange
List of CEs and their attributes
List of FEs and their attributes

Security exchange
FE ID, attributes
CE ID

Security exchange
CE ID, attributes
FE ID

Security exchange
FE ID, attribute
Initial Configuration
Add these new routes
Give me stats
Stats
Port x down
New forwarding table

Ref: http://datatracker.ietf.org/doc/rfc3746/?include_text=1
Washington University in St. Louis http://www.cse.wustl.edu/~jain/cse570-13/ ©2013 Raj Jain
Application Layer Traffic Optimization (ALTO)

- IETF working group to optimize P2P traffic
  ⇒ Better to get files from nearby peers
- Provide guidance in peer selection
- ALTO Server: Has knowledge of 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 from 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 a ordered list
- Also need a protocol for ALTO server discovery

Ref: J. Seedorf and E. Berger, “ALTO Problem Statement,” 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
Washington University in St. Louis http://www.cse.wustl.edu/~jain/cse570-13/ ©2013 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
Current SDN Debate: What vs. How?

- SDN is easy if control plane is centralized but not necessary. Distributed solutions may be required for legacy equipment and for fail-safe operation.
- Complete removal of control plane may be harmful. Exact division of control plane between centralized controller and distributed forwarders is yet to be worked out.
- SDN is easy with a standard southbound protocol like OpenFlow but one protocol may not work in all cases.
  - Diversity of protocols is a fact of life.
  - There are no standard operating systems, processors, routers, or Ethernet switches.
- If industry finds an easier way to solve the same problems by another method, that method may win. E.g., ATM vs. MPLS.
SDN Controller Functions

Northbound APIs
- RESTful API
- OSGi Framework

Network Service Functions
- Slicing Manager
- Topology Manager
- Host Tracker

Network Orchestration Function

Management Function

Controller API (Java, REST)

Service Abstraction Layer (SAL)
- PCEP
- SMTP
- XMPP
- BGP
- OpenFlow V1.0
- OpenFlow V1.1
- OpenFlow V1.4

Network Elements
- Network Element
- Network Element
- Network Element

Overlay Tunnels (VxLAN, NVGRE, …)

Washington University in St. Louis
http://www.cse.wustl.edu/~jain/cse570-13/
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 undependably.
- 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.

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

Native Operating System
Java VM
Execution Environment
Modules
Life Cycle
Services

Bundle A
Register
Service Registry
Get
Bundle B
Listen

Ref: http://www.osgi.org/Technology/WhatIsOSGi
Washington University in St. Louis
http://www.cse.wustl.edu/~jain/cse570-13/
©2013 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
  ⇒ Bundles can be installed and uninstalled on the fly.
OpenDaylight SDN Controller Platform (OSCP)

- Multi-company collaboration under Linux foundation
- Many projects including OpenDaylight Controller
- **NO-OpenFlow** (Not Only OpenFlow): 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

Ref: [https://wiki.opendaylight.org/view/Main_Page](https://wiki.opendaylight.org/view/Main_Page)
OpenDaylight Tools

1. **Applications**: Provides Virtual Network Segments (VNS) for each tenant
   1. OpenDaylight Network Virtualization (ONV):
   2. OpenDaylight Virtual Tenant Network (VTN)

2. **Services**:
   1. Defense4All: Security

3. **Northbound APIs**:
   1. REST
   2. **Dlux**: Northbound API using AngularJS, an extension of HTML by Google for dynamic views
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
   3. Affinity Metadata Service
Affinity Metadata Service

- API to create an abstract topology and implementation independent description of infrastructure needs and behaviors of network workloads
- Allows intent to be specified in application and service terms independent of where and how the workloads attach to the network.
- SDN controllers and application can use “affinity” information to automatically provision the VMs and network for the user
- Users don’t need to know about bridges, routers, VLANs, and tunnels

Ref: https://wiki.opendaylight.org/view/Project_Proposals:Affinity_Metadata_Service
Washington University in St. Louis http://www.cse.wustl.edu/~jain/cse570-13/ ©2013 Raj Jain
Summary

1. SDN is the framework to automatically manage and control a large number of network devices and services in a multi-tenant environment.

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, ForCES, PCE, ALTO.

3. OpenDaylight SDN Controller platform is the leading open source SDN controller project under Linux Foundation.

4. It uses REST APIs and OSGI framework for modularity.
Reading List

- J. Seedorf and E. Berger, “ALTO Problem Statement,”
  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://www.osgi.org/Technology/WhatIsOSGi
- http://www.sdncentral.com/sdn-use-cases/
- http://datatracker.ietf.org/wg/pce/
- https://wiki.opendaylight.org/view/Main_Page
Wikipedia Links

- [Software-defined networking](http://en.wikipedia.org/wiki/Software-defined_networking)
- [Representational state transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)
- [OSGI](http://en.wikipedia.org/wiki/OSGI)
- [XMPP](http://en.wikipedia.org/wiki/XMPP)
- [Path computation element](http://en.wikipedia.org/wiki/Path_computation_element)
References

- OpenDaylight Components and Tools:
  - https://wiki.opendaylight.org/view/OpenDaylight_Network_Virtualization_%28ONV%29:Main
  - https://wiki.opendaylight.org/view/OVSDB_Integration:Design
  - https://wiki.opendaylight.org/view/Project_Proposals:Affinity_Metadata_Service
  - https://wiki.opendaylight.org/view/Project_Proposals:BGP_and_PCEP
  - https://wiki.opendaylight.org/view/Project_Proposals:Defense4All
  - https://wiki.opendaylight.org/view/Project_Proposals:Dlux
References (Cont)

- https://wiki.opendaylight.org/view/Project_Proposals:SNMP4SDN
- https://wiki.opendaylight.org/view/YANG_Tools:Main
- Open Data Center Alliance Usage Model: Software Defined Networking Rev 1.0,”
### Acronyms

- **ACI**  Application Policy Infrastructure  
- **ACL**  Access Control List  
- **AEX**  Application Information Exposure  
- **ALG**  Application Level Gateway  
- **ALTO**  Application Layer Traffic Optimization  
- **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  
- **BUM**  Broadcast, Unknown, and Multicast
Acronyms (Cont)

- 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
- DOVE Distributed Overlay Virtual Ethernet
- DVS Distributed Virtual Switch
- EID Endpoint Identifier
- ETSI European Telecommunications Standards Institute
- FCAPS Faults, configuration, accounting, performance, and security
- FE Forwarding Element
Acronyms (Cont)

- 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
- IP Internet Protocol
- IPv4 Internet Protocol version 4
Acronyms (Cont)

- IPv6: Internet Protocol version 6
- IRTF: Internet Research Taskforce
- IS-IS: Intermediate System to Intermediate System
- ISO: International Standards Organization
- 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
- NFV: Network Function Virtualization
- NTP: Network Time Protocol
- NVGRE: Network Virtualization using Generic Routing Encapsulation
- NVO3: Network Virtualization over L3
- NVP: Network Virtualization Platform
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OF</td>
<td>OpenFlow</td>
</tr>
<tr>
<td>OnePK</td>
<td>Open Network Environment Platform Kit</td>
</tr>
<tr>
<td>ONF</td>
<td>Open Networking Forum</td>
</tr>
<tr>
<td>ONV</td>
<td>OpenDaylight Network Virtualization</td>
</tr>
<tr>
<td>OpEx</td>
<td>Operational Expenses</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>OSCP</td>
<td>OpenDaylight SDN Controller Platform</td>
</tr>
<tr>
<td>OSGi</td>
<td>Open Services Gateway Initiative</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>OVS</td>
<td>Open Virtual Switch</td>
</tr>
<tr>
<td>OVSDB</td>
<td>Open Virtual Switch Database</td>
</tr>
<tr>
<td>PCC</td>
<td>Path Computation Client</td>
</tr>
<tr>
<td>PCE</td>
<td>Path Computation Element</td>
</tr>
<tr>
<td>PCEP</td>
<td>Path Computation Element Protocol</td>
</tr>
<tr>
<td>POP</td>
<td>Post Office Protocol</td>
</tr>
<tr>
<td>PWE3</td>
<td>Pseudowire Emulation Edge to Edge</td>
</tr>
</tbody>
</table>
Acronyms (Cont)

- 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
- 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
Acronyms (Cont)

- URI Uniform Resource Identifier
- vBridge Virtual Bridge
- VIRL Virtual Internet Routing Lab
- VLAN Virtual Local Area Network
- VM Virtual Machine
- VNS Virtual Network Segment
- VPN Virtual Private Network
- vTep Virtual Tunnel End Point
- VTN Virtual Tenant Network
- VxLAN Virtual Extensible Local Area Network
- WAN Wide Area Network
- XML Extensible Markup Language
- XMPP Extensible Messaging and Presence Protocol
SDN Related Organizations and Projects

- Open Networking Foundation (ONF): www.opennetworking.org
- Telecom Industry Association (TIA): www.tiaonline.org
- Internet Engineering Task Force (IETF): www.ietf.org
- Open Data Center Alliance, http://www.opendatacenteralliance.org
- OpenStack Quantum: https://wiki.openstack.org/wiki/Quantum
- OpenDaylight: www.opendaylight.org
SDN Web Sites

- SDN Central, [http://www.sdncentral.com](http://www.sdncentral.com)