Software Defined Network (SDN) for Service Providers

Santanu Dasgupta
Sr. Consulting Engineer – Global Service Provider HQ

SANOG 21
January 28th, 2013
“...In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications...”

OpenFlow

“...open standard that enables researchers to run experimental protocols in campus networks. Provides standard hook for researchers to run experiments, without exposing internal working of vendor devices......”
“A platform for developing new control planes”

“A solution to build a very large scale layer-2 network”

“A means to do traffic engineering without MPLS”

“A way to optimize link utilization in my network, through new multi-path algorithms”

“A way to avoid lock-in to a single networking vendor”

“A solution to build virtual networks with specific topologies for my multi-tenant Data-Center”

“A way to configure my entire network as a whole rather than individual devices”

“A means to scale my fixed/mobile gateways and optimize their placement”

“A way to define virtual networks with specific topologies for my multi-tenant Data-Center”

“A way to configure my entire network as a whole rather than individual devices”

“A way to distribute policy/intent, e.g. for DDoS prevention, in the network”

“A solution to get a global view of the network – topology and state”

“An open solution for VM mobility in the Data-Center”

“An open solution for customized flow forwarding control in the Data-Center”

“Develop solutions software speeds: I don’t want to work with my network vendor or go through lengthy standardization.”

“A way to reduce the CAPEX of my network and leverage commodity switches”

“A way to build my own security/encryption solution, avoiding RSA”

“A way to scale my firewalls and load balancers”

“A means to do traffic engineering without MPLS”

“A way to optimize link utilization in my network, through new multi-path algorithms”

“A solution to build a very large scale layer-2 network”

“A means to do traffic engineering without MPLS”

“A way to define virtual networks with specific topologies for my multi-tenant Data-Center”

“A way to configure my entire network as a whole rather than individual devices”

“A means to scale my fixed/mobile gateways and optimize their placement”

“A way to define virtual networks with specific topologies for my multi-tenant Data-Center”

“A way to configure my entire network as a whole rather than individual devices”

“A way to distribute policy/intent, e.g. for DDoS prevention, in the network”

“A solution to get a global view of the network – topology and state”

“An open solution for VM mobility in the Data-Center”

“A platform for developing new control planes”

“Develop solutions software speeds: I don’t want to work with my network vendor or go through lengthy standardization.”

“A way to reduce the CAPEX of my network and leverage commodity switches”

“A way to build my own security/encryption solution, avoiding RSA”

“A way to scale my firewalls and load balancers”

“A means to do traffic engineering without MPLS”

“A way to optimize link utilization in my network, through new multi-path algorithms”

“A solution to build a very large scale layer-2 network”

“A means to do traffic engineering without MPLS”

“A way to define virtual networks with specific topologies for my multi-tenant Data-Center”

“A way to configure my entire network as a whole rather than individual devices”

“A means to scale my fixed/mobile gateways and optimize their placement”

“A way to define virtual networks with specific topologies for my multi-tenant Data-Center”

“A way to configure my entire network as a whole rather than individual devices”

“A way to distribute policy/intent, e.g. for DDoS prevention, in the network”

“A solution to get a global view of the network – topology and state”
Towards the Open Network Environment for SDN
Implementation Perspective: Evolve the Control-Plane Architecture

- Enable modularization and componentization of network control- and data-plane functions, with associated open interfaces. This allows for optimized placement of these components (network devices, dedicated servers, application servers) and close interlock between applications and network functions.

- Anticipated benefits include: Closely align the control plane with the needs of applications, enable componentization with associated APIs, improve performance and robustness, enhance manageability, operations and consistency.
Cisco Open Network Environment (ONE)
Flexibility in Deriving Abstractions
Open Network Environment

• Open Network Environment – Complementing the Intelligent Network

  *Preserve what is working:* Resiliency, Scale and Security, Comprehensive feature-set

  *Evolve for Emerging Requirements:* Operational Simplicity, Programmability, Application-awareness

• The Open Network Environment integrates with existing infrastructure

  Software Defined Network concepts are a component of the Open Network Environment

  The OpenFlow protocol can be used to link agents and controllers, and as such is component of SDN as well
Programmatic APIs
Approaching abstractions for Networking

- Abstractions allow the definition of associated APIs
  - Enable API platform kit across all platforms, to integrate with development environments
  - Accelerate development of network applications: Completely integrated stack from device to network
  - Multiple deployment modes (local and remote (blade/server) based APIs)
  - Multiple Language Support (C, Java, Python…)
  - Integrate with customer development to deliver enhanced routing, forwarding..
Agents and Controllers
Agent-Controller pairs: A few well-known and evolving examples

- Agents and Controllers are a component of Open Network Environments and a key component of the evolving “Software Defined Network” concept.
- The Concept of Agents and Controllers exists in the Industry for quite some time.
- Agent-Controller pairs always serve a specific task (or set of tasks) in a specific domain.
OpenFlow Basics

- OpenFlow Components
  
  **Application Layer Protocol**: OF-Protocol
  
  **Device Model**: OF-Device Model (abstraction of a device with Ethernet interfaces and a set of forwarding capabilities)
  
  **Transport Protocol**: Connection between OF-Controller and OF-Device*

- Observation:
  
  OF-Controller and OF-Device need pre-established IP-connectivity

* TLS, TCP – OF 1.3.0 introduces auxiliary connections, which can use TCP, TLS, DTLS, or UDP.

Source: OpenFlow 1.3.0 specification, figure 1
OpenFlow Processing Pipeline

**OF 1.0 model**
(single lookup)

- Controller
- Ingress Port
- Packet IN
- Action Set {}
- Packet OUT
- Packet DROP

**OF 1.1 and beyond model**
(multiple lookups)

1. Find highest-priority matching flow entry
2. Apply instructions:
   i. Modify packet & update match fields 
      (apply actions instruction)
   ii. Update action set (clear actions and/or 
       write actions instructions)
   iii. Update metadata
3. Send match data and action set to 
    next table

Source: OpenFlow 1.3.0 specification, figure 2
OpenFlow Protocol Specification Evolution

Dec 31, 2009  
OF 1.0
• Single Table  
• L2, IPv4 focused matching

Feb 28, 2011  
OF 1.1
• Multiple Tables  
• MPLS, VLAN matching  
• Groups: {Any-,Multi-)cast  
• ECMP

Dec 5, 2011  
OF 1.2
• IPv6  
• Flexible-length matching

April 19, 2012  
OF 1.3.0
• 802.1ah PBB  
• Multiple parallel channels between Switch and Controller

A few topics of ongoing work

High availability model for device and controller (state re-sync etc.)
Hardware friendly switch model – “typed tables” → New Forwarding Abstractions WG
Security model (granular access control)
Layer 3 forwarding model
Enhanced Statistics
Management infrastructure (evolution of OF-CONFIG)
Testing and certification framework
Hybrid device/network deployment capability (→ Hybrid WG)
Virtualization: Network Partitioning
Example: Network Slicing for Research Environments

• Business Problem
  University desires to “slice” the network into multiple partitions:
  Production network – classic control plane
  Several research networks – experimentation with new control algorithms, programs etc.

• Solution
  Network Slicing Manager (built as an app on top of the ONE controller) partitions the network based on e.g. ports, VLANs, aggregate flow matches,..
  Effects of a particular control function of a partition/slice limited to that partition/slice
Open Network Environment: Standards Context
Orchestrating Efforts across multiple SDOs/Fora, including IETF, IEEE, ONF, ITU, …

- Network Configuration (NETCONF WG)
- Application Layer Traffic Optimization (ALTO WG)
- Content Delivery Networks Interconnection (CDNI WG)
- Extensible Messaging and Presence Protocol (XMPP WG)
- Software Driven Networks (SDNP BOF)
- Infrastructure-to-application information exposure (I2AEX BOF)

- OpenFlow Device Configuration (ONF Config WG)
- Path Computation Control (PCE WG)
- Forwarding & Control Separation (FORCES WG)
- OF Hybrid Devices/Networks (Hybrid WG)
- OF Protocol Extensions (Extensibility WG)
- OF Futures: “FPMOD/OF2.0” (Futures group)

- Future Networks
  - Y.FNsdr: SDN Framework
  - Y.FNsdr-fm: Requirements formal specification and verification methods for SDN

- Network Virtualization Overlays (NVO3 WG)
- L2VPN WG
- L3VPN WG
- Locator/ID Separation Protocol (LISP WG)
- Pseudowire Emulation Edge to Edge (PWE3 WG)
- 802.1aq Shortest Path Briding
- 802.1Qbp Equal Cost Multiple Paths
- 802.1Qbg – Edge Virtual Bridging
- 802.1BR – Bridge Port Extension

- SG13 – Future Networks
  - Y.Fnvirtreq - Requirement of network virtualization
SDN For Service Providers
A Few Use Cases
DNS Proximity Guided Resolution

1. DNS Request
2. NPS Request
3. NPS Reply
4. DNS Reply

Network Services Layer
- NPS Database
- NPS/Proximity Server
  - Information collector
  - Algorithms
  - Databases
- Routing Protocols
- Databases

Network Layer
- PE-1
- PE-2
- PE-3
- PE-4

Policy Database
- Geo-location
- Performance data
Multi Layer Path Computation Element

Multi-layer Coordination

R1

Service

ServiceBackhaul

IP/MPLS

R2

Service

Tunnel

Link

R3

Service

ServiceBackhaul

User

Service 

ServiceBackhaul

IP/MPLS

ML-PCE

ML-PCE

Service Control & Admin

Setup Service Wires (PCEP, OF++)

Discovery, Status

Service Control & Admin

Setup Service Instances

Setup Tunnels (PCEP)

L3 Link Topology (BGP-LS)

Setup λ's (PCEP)

DWDM Topology (OSPF/BGP-LS)

Service Backhaul

IP/MPLS

DWDM

© 2011 Cisco and/or its affiliates. All rights reserved.
1. Customer requests premium access (eg, priority QoS, incremental BW) to XYZ cloud service
2. Policy server pushes customer policy changes to Ingress & Egress PE(s) attached to customer sites
3. OnePK application overrides default VPN behavior
4. Customer traffic matching the policy is given premium treatment

Using Programatic Access, SPs can build such custom apps to create differentiated, revenue generating services
Use Case from Verizon – Mobile Video Optimization in RAN

Software Defined Networking Today:
Traffic Steering for Content Management & Distribution

- By subscriber based on subscriber policy
- By application type through content inspection
- By cache asset based on URL
- By destination based on RAN congestion signaling
- By server based on performance and availability
Summary – SDN for SP

Evolution, not Revolution
- Use SDN to Augment capabilities
- Hybrid mode Ops/Architecture
- Use case driven

SDN is an architecture, OpenFlow is a protocol
- SDN is much broader
- OpenFlow is just one component alongside others

No one size fits all
- Diverse segment will have diverse requirement, will require different solutions

Program Policies for Optimized Experience
- App Developer Environment
- Analysis and Monitoring, Performance & Security
- Network Elements & Abstraction

Evolve for Emerging Requirements
- Any Object
  - Switch/Router
  - ASIC
  - Network Fabric
  - Compute
- Any Service
  - Cloud
  - Collaboration
  - Video
  - Security
  - Mobility
- Any Layer
  - L1-7
  - Control/Data Plane
  - Hardware/Software
  - ASICs/OS

Use Case Driven – Making Network More Intelligent
Thank you.