The Open Data Center: Evolution in Revolution

Jimmy Ray Purser
Chief Geek, TechWiseTV
Mar 2013
Headlines

“Google revamps networks with OpenFlow”
- ZDnet

“Cisco on board with SDN, OpenFlow”
Enterprise networking planet, April 2012

“Cisco talks up software-defined networking battle plans”
Wired.com, April 2012

“Cisco still dragging its feet on OpenFlow”
NetworkWorld, Feb 2012

“Cisco offers more SDN hints, Confirms Insieme”
Network Computing, April 2012

“SDN needs a bigger definition”
Lippis report, 2012
What is 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…”

What is Software-defined network (SDN)?

“…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…”

Source: www.opennetworking.org
Evolution of the Intelligent Network

Preserve what’s working
- Resiliency
- Scale
- Rich feature-set

Evolve for emerging requirements
- Operational Simplicity
- Programmability
- Application Awareness

Augment the Network for the next wave of application requirements
Customer Insights

Research/Academia
- Experimental OpenFlow/SDN components for production networks
  - Network “Slicing”
  - Network flow management

Massively Scalable Data Center
- Customize with Programmatic APIs to provide deep insight into network traffic
  - Scalable Multi-tenancy

Cloud
- Automated provisioning and programmable overlay
  - Agile service delivery
  - Scalable Multi-tenancy

Service Providers
- Policy-based control and analytics to optimize and monetize service delivery

Enterprise
- Virtualization of workloads, VDI, Orchestration of security profiles
  - Private Cloud Automation

Diverse functionality required across segments
Industry Standards

802.1 Overlay Networking Projects, Cisco Innovations:
FEX Architecture

Technical Advisory Group Chair, Working Groups:
Config, Hybrid, Extensibility, Futures/FPMOD/OF2.0

Open Source Cloud Computing project

Open Network Research Center at Stanford University

Working Groups: Quantum API
Donabe
Cisco Innovations: OpenStack API for Nexus OpenStack Extensions

Overlay Working Groups:
NVO3, L2VPN, TRILL, L3VPN, LISP, PWE3

API Working Groups:
NETCONF, ALTO, CDNI, XMPP, SDNP, I2AEX

Controller Working Groups:
PCE, FORCES
Let’s start at the Cloud!
OpenStack

• Open source Cloud Computing Software framework

• Founding contributors
  • NASA Nebula Project
  • Rackspace Cloud

• Target Users
  • Cloud Service Providers
  • Enterprises – Private Cloud
  • Researchers

• Full Open Source
  • Under Apache 2.0 license
  • Anyone can run it, built on it and submit changes back.
OpenStack

NASA Ozone
- Written in Python
- Team evaluated Eucalyptus and moved away from it
  - complex JAVA code
  - Stability concerns
- AWS-compatible APIs
- Makes use of euca2ools

Rackspace Cloud Files
- Written in C++/C/Python
- Object store constructs
- Runs on commodity hardware

OpenStack Compute

OpenStack Storage

Apache 2.0 license

OpenStack: Compute
- 1,000,000 hosts
- 60,000,000 guests

Goals

OpenStack: Storage
- 100 Petabytes per cluster
- 100,000 requests / second
- Hundreds of billions of objects
OpenStack – Cloud layer

- Euca2ools
- OpenStack API
- OpenStack Client Tools
- OpenStack Dashboard
- Nova-API Server
- Swift Object Storage
- Nova-Object Store
- Nova-Compute Node
- Nova-Network Node
- Nova-Volume Node
- Glance

Virtual Infra

Physical Infra
OpenStack Community

<table>
<thead>
<tr>
<th>Rackspace Hosting</th>
<th>NASA</th>
<th>Citrix</th>
<th>Dell</th>
<th>NTT Data</th>
<th>PEER 1</th>
<th>SoftLayer</th>
<th>RightScale</th>
<th>Cloud.com</th>
<th>Iomart Group</th>
<th>OC3</th>
<th>OPSCODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puppet Labs</td>
<td>PathtenDB</td>
<td>cloudkick</td>
<td>Zenoss</td>
<td>Limelight</td>
<td>Cloudscaling</td>
<td>AMD</td>
<td>Riptano</td>
<td>Intel</td>
<td>Autonomic Resources</td>
<td>Zuora</td>
<td></td>
</tr>
<tr>
<td>Nicira</td>
<td>Spiceworks</td>
<td>Sonian</td>
<td>CloudSwitch</td>
<td>Cloud Central</td>
<td>Morphlabs</td>
<td>jPath</td>
<td>Vyatta</td>
<td>Infini</td>
<td>Cirrascale</td>
<td>Internap</td>
<td></td>
</tr>
<tr>
<td>EnterpriseDB</td>
<td>Gigaspaces</td>
<td>Chengso</td>
<td>NTT</td>
<td><a href="http://www.opatan.com/">http://www.opatan.com/</a></td>
<td>L&amp;T Infosys</td>
<td>Gloudnet</td>
<td>Gluster</td>
<td>Nephoscale</td>
<td>Korean Telecom</td>
<td>Arista Networks</td>
<td></td>
</tr>
<tr>
<td>B1 Systems</td>
<td>Canonical</td>
<td>Grid Dynamics</td>
<td>Cisco</td>
<td>Extreme Networks</td>
<td>AFORE Solutions</td>
<td>ComputeNext</td>
<td>Cybera</td>
<td>Ultimesofr</td>
<td>StackOps</td>
<td>CBS Corp</td>
<td></td>
</tr>
<tr>
<td>Cloud Cruiser</td>
<td>Zadara Storage</td>
<td>CloudBees</td>
<td>Caldera</td>
<td>Memset</td>
<td>Bull</td>
<td>OW2</td>
<td>Brocade</td>
<td>Abiquo</td>
<td>SmartScale</td>
<td>Mellanox</td>
<td></td>
</tr>
<tr>
<td>SME Storage</td>
<td>Mach Technology</td>
<td>Taxi1 Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2011 Cisco and/or its affiliates. All rights reserved.
“A platform for developing new control planes”

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

“With SDN I can develop solutions to my problems far faster – “at software speeds”. I don’t have to work with my network vendor or go through length standardization”

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

“An means to do traffic engineering without MPLS”

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

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

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

“A solution to build virtual topologies with optimum multicast forwarding behavior”

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

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

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

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

“A way to scale my firewalls and loadbalancers”

“A solution 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”

“SDN – Many things to Many people”
What is Software Defined Networking?

- SDN is an approach that uses programmatic interfaces, combining
  - network overlay virtualization
  - enhanced abstraction of network functions
  - direct access to network infrastructure at both network as well as device level for applications and tools
- SDN is an incremental capability and can be layered onto existing infrastructure
  - OpenFlow is an example of an interface and can be a component of SDN
**Software Defined Networks Implementation Perspective**

- SDN enables 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 a clear control-/data-plane split, improve performance and robustness, enhance manageability, operations and consistency.
Case studies for SDN
Service Providers
## SP SDN use cases

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Details</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Combine application data with network data to control traffic</td>
<td>Use policy servers, analytics, subscriber data to make forwarding decisions. Examples: Subscriber services (mobile, DSL), Security (DDOS-like), Backbone traffic engineering, Self Service policy provisioning.</td>
</tr>
<tr>
<td>2</td>
<td>Run applications in the network</td>
<td>Use compute resources in the network, allow faster delivery of network services, move elements from cloud to network and vice versa (flexibility). Examples: SLA reporting, Traffic monitoring, RR,…</td>
</tr>
<tr>
<td>3</td>
<td>Network To Cloud Services</td>
<td>Tie network services with Cloud services (SP cloud or SP partners: Content Providers, SAAS…): Cloud SLA, Cloud security,…</td>
</tr>
<tr>
<td>4</td>
<td>Services Orchestration</td>
<td>Combine network services with L4L7 services, chained. Tie into service orchestration - Within one platform - Across the Network</td>
</tr>
</tbody>
</table>
Use Case 1: Forwarding Control

Policy: if “match” & “context” then “Set”

Context: subscriber info, location TOD, Analytics (app performance, congestion, …), …

SP Business Problem: “need to easily handle traffic based on my business imperatives”

Requirement: To translate business imperatives into traffic handling decisions, I need to combine network intelligence with other data sources

SDN Solution: Complement the device’s control functions with alternate control functions to leverage additional data sources
Use Case 2: Applications in The Network

SP Business Problem: Use compute resources in the network
Requirement: Have flexibility to move applications between the Data Center and the network-
- Have access to device API
SDN Solution: Network Programmability- Application virtualization across network and Data Center
Use Case 3: Network To Cloud

**SP Business Problem:** Provide customers differentiated access to Cloud services and SAAS providers

**Requirement:** Map network services to cloud services - Ensure proper SLA, security,…

**SDN Solution:** Network Programmability allows for mapping of network flow to cloud
Use Case 4: Service Orchestration

SP Business Problem: Increase revenue stream by offering value added services
Requirement: To be able to provide service chaining and orchestration based on service profile - Combine with dynamic flow offload
SDN Solution: Service chaining and Service orchestration - Dynamic service chains across virtual, physical, and even 3rd party appliances

At Home

On the Move

At Work

Residential Customer Group A

Residential Customer Group B

Business Internet

Business VPN “Corp X”

DPI
Virus/Malware Scan
Firewall

CGN
Firewall

CGN
DDOS Protection
Firewall

IPSec
Virus/Malware Scan
SBC
Test on OPEN FLOW class
OpenFlow Switch and Control Plane Model

- OpenFlow is based on an **L2-L4 Ethernet switch**, with an internal flow-table pipeline, and a "standardized" interface to add and remove flow entries.

- Forwarding “Instruction set”
- New actions can be done on packet.
  - Atypical matching
  - Field modification
  - Route/switch on new criteria
  - Network slicing
  - New algorithms: multipath, load-balancing

*Partial slide content from ITU*
Flow tables are Match-Action points

Rules : match against packets

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stats</td>
</tr>
</tbody>
</table>

Counters : per-table, per-flow, per-port and queue

1. Forward packet to : (optional)
   1. All : not incoming iface
   2. Controller : encapsulate and send
   3. Local : to the local networking switch stack
   4. Table : perform actions in flow table
   5. In-port : send to given port
   6. Normal : traditional forwarding path
   7. Flood : along the minimum spt
   8. Group (ECMP, FRR, Multicast)

2. Enqueue
3. Drop
4. Modify field (VLAN, MAC, IP, TOS, Ports, TTL)
5. Encap/decap

Version 1.x

+ Multi table
+ Metadata
+ IPv6
+ MPLS
+ Tunnel/Encap

Ingress Port | MAC src | MAC dst | Eth type | VLAN ID | VLAN prilo | IP Src | IP Dst | IP Prot | IP TOS | TCP/UDT sport | TCP/UDP dport
|-----------|--------|--------|---------|--------|-----------|-------|-------|--------|-------|--------------|--------------

+ mask, wildcards

(source OpenFlow)
OF Processing Pipeline

**OF 1.0 model** (single lookup)

- Single Table
- Controller
- Packet IN
- Ingress Port
- Packet+
- Ingress Port + Metadata
- Action Set
- Packet
- 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 Logical Ports

Configuration and determination of physical and logical ports presented to the OpenFlow controller is out of scope of the OpenFlow protocol.
OpenFlow SDN Overview

- **Controller**
  - **OpenFlow Agent**
    - **Flow API**
    - **Platform Dataplane** (e.g. ACL TCAM)
    - **HW API**

- **Data packets**

- **Network Mgmt**
  - **OF-Config/Netconf**

- **Cisco NOS Switch/Router Platform**

- **Data**

- **OF/CSDN Protocol**

- **Interface Management**
OpenFlow is Built on onePK

ONE Agent Framework (proposed agents)

- NETCONF Agent
- CIM Agent
- OpenFlow Agent
- Puppet Agent
- Prime agent
- Custom Agent
- onePK Mgmt Apps

Process boundary

- onePK Client
- onePK Server

Comm libraries

onePK Presentation APIs (C, Java, Python, ...)

- Comm libraries

onePK Abstraction APIs

- Element
- Interface
- Developer
- Utilities
- Discovery
- Policy
- Routing
- Datapath
- Ext...

Cisco Network Operating System (IOS, IOS-XE, IOS-XR, NX-OS) (Platform PI Code)

Cisco Network Operating System (IOS, IOS-XE, IOS-XR, NX-OS) (Platform PD Code)
OnePK!!! Almost as Awesome as my Laptime!!!
Evolving How We Interact With The Network Operating System
Evolving How We Interact With The Network Operating System

Traditional Approach

IOS
- Monitoring
- Policy
- Interface
- Discovery
- Routing
- Data Plane

Events

Actions

App EEM (TCL)

New Paradigm

App
- C
- Java

Anything you can think of
Introducing One Platform Kit - onePK

Flexible development environment to:
- Innovate
- Extend
- Automate
- Customize
- Enhance
- Modify

Applications That YOU Create

onePK

Any Cisco Router or Switch
onePK Architecture

C, JAVA Program

onePK API Presentation

onePK API Infrastructure

IOS / XE
(Catalyst, ISR, ASR1K)

NXOS
(Nexus Platforms)

IOS XR
(ASR 9K, CRS)
onePK Application Hosting Options

Process Hosting

- Container
- onePK Apps
- Network OS

Blade Hosting

- Blade
- Container
- onePK Apps
- Network OS

End-Point Hosting

- External Server
- onePK Apps
- Network OS

Write Once, Run Anywhere
Custom Routing
Initial Setup: Default routing using EIGRP
Custom Routing
Routing for Dollars: Application driven routes installed in network
Custom Routing

Tracing the application installed route – using the developer and element services
Custom Routing
Tracing the application installed route – using the developer and element services

Gateway of last resort is not set

10.6.8.8/8 is variably subnetted, 6 subnets, 2 masks
C 10.1.1.0/24 is directly connected, Ethernet0/0
L 10.1.1.4/32 is directly connected, Ethernet0/0
D 10.40.1.0/24 [98/268/1856] via 40.10.1.12, 2vld, Serial2/0
D 10.50.1.0/24 [98/319/3586] via 40.10.1.12, 2vld, Serial2/0
D 10.60.1.0/24 [98/370/5856] via 40.10.1.12, 2vld, Serial2/0
D 10.70.1.0/24 [98/319/3586] via 40.10.1.12, 2vld, Serial2/0

20.6.8.0/24 is subnetted, 10 subnets
D 20.10.1.0 [98/370/5856] via 40.10.1.12, 2vld, Serial2/0
D 20.20.1.0 [98/472/9856] via 40.10.1.12, 2vld, Serial2/0
D 20.30.1.0 [98/370/5856] via 40.10.1.12, 2vld, Serial2/0
D 20.40.1.0 [98/423/7856] via 40.10.1.12, 2vld, Serial2/0
D 20.50.1.0 [98/423/7856] via 40.10.1.12, 2vld, Serial2/0
D 20.60.1.0 [98/423/7856] via 40.10.1.12, 2vld, Serial2/0
D 20.70.1.0 [98/472/9856] via 40.10.1.12, 2vld, Serial2/0
D 20.80.1.0 [98/423/7856] via 40.10.1.12, 2vld, Serial2/0
D 20.90.1.0 [98/620/8586] via 40.10.1.12, 2vld, Serial2/0
D 20.100.1.0 [98/472/9856] via 40.10.1.12, 2vld, Serial2/0

30.6.8.0/24 is subnetted, 5 subnets
D 30.10.1.0 [98/524/1856] via 40.10.1.12, 2vld, Serial2/0
D 30.20.1.0 [98/472/9856] via 40.10.1.12, 2vld, Serial2/0
D 30.30.1.0 [98/472/9856] via 40.10.1.12, 2vld, Serial2/0
D 30.40.1.0 [98/524/1856] via 40.10.1.12, 2vld, Serial2/0
D 30.50.1.0 [98/524/1856] via 40.10.1.12, 2vld, Serial2/0

40.6.8.0/24 is variably subnetted, 4 subnets, 2 masks
C 40.10.1.0/24 is directly connected, Serial2/0
L 40.10.1/32 is directly connected, Serial2/0
C 40.20.1.0/24 is directly connected, Serial2/3
L 40.20.1/32 is directly connected, Serial2/3
108.6.0.0/24 is subnetted, 1 subnets
A 108.1.1.0 is directly connected, 00:01:56, Serial2/3