Software Defined Networking and Use Cases

Quinton Coelho
Consulting Systems Architect
Perception

All things to all people

A better way to build a network

Hardware doesn’t matter

The answer to *every* network issue

Simplified troubleshooting

Unlimited bandwidth

Unlimited resources

Whatever you want it to be
SDN Is...

“An open solution for VM mobility in the Data-Center”
“A way to reduce the CAPEX of my network and leverage commodity switches”
“A solution to build virtual topologies with optimum multicast forwarding behavior”
“A means to scale my fixed/mobile gateways and optimize their placement”
“A way to distribute policy/intent, e.g. for DDoS prevention, in the network”

“A platform for developing new control planes”
“A solution to automated network configuration and control”
“A means to get assured quality of experience for my cloud service offerings”
“A way to optimize link utilization in my network enhanced, application driven routing”

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

“An open solution for customized flow forwarding control in and between Data Centers”
“A means to do traffic engineering without MPLS”
“A way to build my own security/encryption solution”
“A solution to get a global view of the network – topology and state”

“A way to optimize broadcast TV delivery by optimizing cache placement and cache selection”
“A way to configure my entire network as a whole rather than individual devices”

Simplified Operations – Enhanced Agility – New Business Opportunities
Perception

Evolving way of centralizing network control.
Specialized hardware is still beneficial
Lowest common denominator features
A process of defining network requirements
Ability to automate QoS deployments
Ability to enforce policy for an entire network
Centralized control providing in a repeatable automated fashion what you can already do today.

The latest buzz word
“...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...”


“...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......”

http://www.openflow.org/wp/learnmore/
Customer Insights: Network Programmability

- **Research/Academia**: Experimental OpenFlow/SDN components for production networks
- **Massively Scalable Data Center**: Customize with Programmatic APIs to provide deep insight into network traffic
- **Cloud**: Automated provisioning and programmable overlay, OpenStack
- **Service Providers**: Policy-based control and analytics to optimize and monetize service delivery
- **Enterprise**: Virtual workloads, VDI, Orchestration of security profiles

**Network Programmability Requirements Across Segments:**
- Automation, Monitoring & Flow Programmability
Classes of Use-Cases
“Leveraging APIs and logically centralized control plane components”

- Custom Routing (incl. business logic)
  Online Traffic Engineering

- Custom Traffic Processing
  (Analytics, Encryption)

- Consistent Network Policy,
  Security, Thread Mitigation

- Virtualization and Domain Isolation
  (Device/Appliance/Network)

- Federating different Network Control Points
  (LAN-WAN, DC-WAN, Virtual-Physical, Layer-1-3)

Automation of
Network Control
and Configuration
(Fulfillment and Assurance)
Virtual & Physical
Network Programmability Models
Implementation Perspective: Evolve the Control-Plane Architecture

1. Programmable APIs

   - Applications
   - Vendor-specific APIs
   - Control Plane: CLI, SNMP, Netflow, ...
   - Data Plane

2a. Classic SDN

   - Applications
   - Vendor-specific APIs
   - Control Plane: OpenFlow
   - Data Plane

2b. Hybrid “SDN”

   - Applications
   - Vendor-specific APIs
   - Control Plane: OpenFlow
   - Data Plane

3. Network Virtualization/Virtual Overlays

   - Applications
   - Vendor-specific APIs
   - Virtual Control Plane
   - Virtual Data Plane
   - Overlay Protocols (e.g. VXLAN)

Openstack and Network Overlays Apply to All Models (Physical/Virtual)
Custom Features Can Be Built
Implementing Customer Use Cases

Approach 1

APPs

Network

Approach 2

Controller

OpenFlow
Device

Device w/
OpenFlow

Device

Other
Agents

Approach 3

Virtual Overlays

Network

Cisco Approach: Flexibility to Choose—The Power of “AND”
Cisco ONE - Open Network Environment
Harnessing Network Value

CHECK

POLICY
Orchestration
ANALYTICS

Program for Optimized Experience

Programmability

Harvest Network Intelligence

SET

NETWORK

GET

Intelligence
Cisco’s Differentiation: Multi-layered Programmability

Flexibility in Deriving Abstractions

- Management and Orchestration
- Network Services
- Control Plane
- Forwarding Plane
- Transport

Application Developer Environment

OpenFlow/SDN

Open Network Environment

Harvest
Network Intelligence

Program for Optimized Experience
Cisco Open Network Environment – Announced June 2012

Industry’s Most Comprehensive Networking Portfolio
- Hardware + Software
- Physical + Virtual
- Network + Compute

Applications

Multi-layer API
- Platform APIs

Controller
- SDN
  - Controller SW (OpenFlow, onePK)
  - OpenFlow 1.x support

Virtual Overlay
- Virtual Overlays
- Open Clouds with Nexus 1000V
  - Multi-hypervisor
  - Multi-service
  - Multi-cloud
  - Openstack support

One Platform Kit (onePK)
- Programmatic APIs for Network
- HW (IOS, IOS-XR, NX-OS)
Cisco Open Network Environment Building Blocks

Platform APIs
- onePK
  - Comprehensive Developer Kit
  - IOS, IOS-XR
  - and NX-OS

Controllers & Agents
- SDN Controller Software
- OpenFlow Agent

Overlay Virtual Networks
- Nexus 1000V
  - OpenStack REST API
- Multi-Hypervisors
  - VXLAN Gateway
- Services Chaining

Industry’s Broadest Approach for Network Programmability
Cisco onePK (one Platform Kit)
Rapid Application Development

Developer Environment
• Language of Choice
• Programmatic Interfaces
• Rich Data Delivery via APIs

Comprehensive Service Sets
• Flexible Apps;
• New Services Monetization Opportunity

Flexible Application Deployment
• On a Service Blade
• On an External Server
• Directly on the Device

Comprehensive and Consistent Platform Support:
• IOS/XE, NX-OS, IOS-XR
onePK Application Hosting Options

Process Hosting
- Network OS
- Container
- onePK Apps

Blade Hosting
- Network OS
- Blade
- Container
- onePK Apps

End-Point Hosting
- Network OS
- External Server
- onePK Apps

Write Once, Run Anywhere
**onePK APIs are Grouped in Service Sets**

<table>
<thead>
<tr>
<th>Base Service Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Path</strong></td>
<td>Provides packet delivery service to application: Copy, Punt, Inject</td>
</tr>
<tr>
<td><strong>Policy</strong></td>
<td>Provides filtering (NBAR, ACL), classification (Class-maps, Policy-maps), actions (Marking, Policing, Queuing, Copy, Punt) and applying policies to interfaces on network elements</td>
</tr>
<tr>
<td><strong>Routing</strong></td>
<td>Read RIB routes, add/remove routes, receive RIB notifications</td>
</tr>
<tr>
<td><strong>Element</strong></td>
<td>Get element properties, CPU/memory statistics, network interfaces, element and interface events</td>
</tr>
<tr>
<td><strong>Discovery</strong></td>
<td>L3 topology and local service discovery</td>
</tr>
<tr>
<td><strong>Utility</strong></td>
<td>Syslog events notification, Path tracing capabilities (ingress/egress and interface stats, next-hop info, etc.)</td>
</tr>
<tr>
<td><strong>Developer</strong></td>
<td>Debug capability, CLI extension which allows application to extend/integrate application’s CLIs with network element</td>
</tr>
</tbody>
</table>
Cisco ONE Software Controller  A JAVA/OSGI Application

Industry’s Most Extensible Controller Architecture

Multiple published APIs for popular languages and software (Eg: OpenStack)

Modular architecture allows rapid adoption of evolving controller functionality while minimizing operational disruption

Extensible protocol support ensures continuous adoption of emerging standards
Use Cases
Getting Properties and Statistics

- **Element**
  - **System**: CPU, Memory, Platform, Serial #, Versions, Uptime, Location, OIR, CLI Changes
  - **Interfaces**: Port, Slot, BW, MTU, TX/RX, BPS, PPS, Errors, Other Stats, Config, Link Changes
  - **Discovery**: CDP, Topology Graph, Edges, Nodes, Topology Changes

- **Application**
Setting Properties and Statistics

- **Element**
  - System
  - Interfaces
  - Discovery

- **Location**
  - IP address, MTU, Clear Stats, Shut/No Shut

- **Application**
  - Filters
Use Case: Campus Network “Slicing”
Partition network for multiple user-communities—“Sandbox” R&D dept.

Solution

- OpenFlow experimental support (v1.0)
- Experimental controller software
- Integrated slicing management
- Programmatic Interfaces (Eg. REST)

Consistent Policy Management for Maximum Flexibility and Innovation
Use Case: Agile Service Delivery for Service Providers
Monetize Via Real-time Network Adaptation and Maintain SLA

Request for Telepresence Session

POLICY

ANALYTICS

Service Provider Network

onePK

Business Center

HD video

Adapt to Meet SLA

Content and Application Provider

Monetize Via Real-time Network Adaptation and Maintain SLA

Adaptive Architecture Optimizes Resource Utilization
Problem: Misconfigurations cause network outages, degrade performance, impact SLAs.
Value proposition: Get, set, and detect configuration changes via cross-platform API

1. Network begins with mismatched parameters on either side of link (e.g. MTU)
2. Application checks parameters on either side and identifies mismatches (red lines)
3. Application sets parameters to match (lines turn green)
4. Application registers for events related to parameters change.
5. Users logs into console and manually changes parameter. Topology indicates change.
Example: Dynamic Bandwidth/QoS Allocation

1. Customer requests premium access to cloud service
2. Policy server pushes customer policy to OnePK on 9k
3. SP Policy Server uses OnePK API to program higher bandwidth QoS policy for specific flow [Customer IP <---> Cloud Service IP]
4. Customer traffic matching the policy is given premium QoS treatment

Using OnePK API, SPs can build such custom apps to create differentiated, revenue generating services
Example: Customer Encryption

Problem: Customers want custom encryption on specific traffic types
Value proposition: Punt traffic of interest, encrypt, and re-inject.

1. Policy APIs on ingress router are set to punt telnet and syslog to app
2. App encrypts punted traffic and re-injects into data path.
3. Policy APIs on egress router punt telnet and syslog to app
4. App decrypts punted traffic and re-injects into data path.
5. Traffic that does not match policy passes through unencrypted.
Use Case: Custom Forwarding - Transit Selection
Utilizing Topology Independent Forwarding

Business Application Driven Requests Flow Based Traffic Steering with Flowspec Granularity

Cisco ONE Controller

HTTP Request

MPLS

Public Internet

Edge Router

Internet2/Other Service

L2/L3/Label/Lambda…
Summary
Cisco Vision: Exposing The Entire Network Value
Programmatic Control across Multiple Network Planes

Program Policies for Optimized Experience

Application Developer Environment
Analysis and Monitoring, Performance and Security
Network Elements and Abstraction

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

Harvest Network Intelligence
Evolutionary step for networking:
Complement/evolve the Network Control Plane where needed

Centered around delivering open, programmable environment for real-world use cases
  - No one-size-fits-all
  - Cisco will support Network Virtualization, APIs and Agents/Controllers
  - Joint evolution with industry and academia

Technology-agnostic
  - Not predicated on a particular technology or standard
  - Draw from Cisco technologies and industry standards

Delivered as incremental functionality
  - Many customers will use hybrid implementations
  - Build upon existing infrastructure with investment protection
Thank you.