引领创新：
思科SDN战略 ONE

桑毅宏
January 10, 2013
## Agenda

<table>
<thead>
<tr>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Of Definition</td>
</tr>
<tr>
<td>Cisco Strategy</td>
</tr>
<tr>
<td>Virtual Overlays</td>
</tr>
<tr>
<td>Controller &amp; OF Switches</td>
</tr>
<tr>
<td>onePK: One Platform Kit</td>
</tr>
</tbody>
</table>
Of Definition
Business Trends

... driving SDN evolution

- User, device and application mobility
- Cloud computing and service
- Consumerization of IT
- Changing traffic patterns (in data centers)
- Agile service creation and delivery
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 "Slicing"

Network Flow Management

Scalable Multi-Tenancy

Agile Service Delivery

Private Cloud Automation

Diverse Programmability Requirements Across Segments
Most Requirements are for Automation & Programmability
SDN Enables.. Optimization and Monetization

Simplification…
Of network provisioning

Service Creation
New monetization opportunities

“Virtual Networks”
Connect multi vendor networks
Headlines

“Google revamps networks with OpenFlow”
—ZDnet

“Prediction: OpenFlow Is Dead by 2014; SDN Reborn in Network Management”
—Mike Fratto, Network Computing

“Will OpenFlow commoditize networks? Impact Cisco margins?”
—Several media publications, Bloggers

“.We share a more pragmatic view, noting Cisco (for example) is likely to view SDN as a TAM expansion opportunity…”
—Deutsche Bank Research note, Wired, April 2012

“Hype around SDN/OpenFlow getting way out of Control. Where have I seen this before…”
—Ethereal mind, Blogger

“SDN needs a bigger definition”
—Lippis report, 2012
SDN – Evolving Definition
### Basic Definitions

<table>
<thead>
<tr>
<th>What Is Software Defined Network (SDN)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>“…In the SDN architecture, the <strong>control and data planes are decoupled</strong>, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications…”</td>
</tr>
<tr>
<td>Source: <a href="http://www.opennetworking.org">www.opennetworking.org</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What Is OpenFlow?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open protocol that specifies <strong>interactions between de-coupled control and data planes</strong></td>
</tr>
<tr>
<td>Note: OF is not mandatory for SDN</td>
</tr>
<tr>
<td>Note: North-bound Controller APIs are vendor-specific</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What is OpenStack?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Opensource software</strong> for building public and private Clouds; includes Compute (Nova), Networking (Quantum) and Storage (Swift) services.</td>
</tr>
<tr>
<td>Note: Applicable to SDN and non-SDN networks</td>
</tr>
<tr>
<td>Source: <a href="http://www.openstack.org">www.openstack.org</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What is Overlay Network?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overlay network is created on existing network infrastructure (physical and/or virtual) using a network protocol. Examples of overlay network protocol are: GRE, VPLS, OTV, LISP and VXLAN</td>
</tr>
<tr>
<td>Note: Applicable to SDN and non-SDN networks</td>
</tr>
</tbody>
</table>

© 2011 Cisco and/or its affiliates. All rights reserved.
Cisco Strategy
Cisco Perspective on SDN

• Cisco continues to pursue broader strategy (Cisco ONE), which includes SDN
  - Programmatic device APIs
  - Network overlay virtualization
  - Network functional abstractions (controller & openflow for SDN)

Cisco's portfolio already includes several key components of an SDN solution

• OpenFlow is a protocol, not an architecture
  OpenFlow primarily define a protocol for packet forwarding
  OpenFlow is not complete for production (e.g. lacks: high availability, security, L3-forwarding model, management infrastructure, testing and certification framework, hybrid deployment capability)

• Migration to SDN will be evolutionary
  Cisco will take a use-case driven approach that draws on several key elements of Cisco's product and technology portfolio
  Cisco will in the near term engage with specific customers on OpenFlow as a prototype technology
Networking Standards are Critical

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

- **Open Source Cloud Computing project**

- **802.1 Overlay Networking Projects, Cisco Innovations:**
  - FEX Architecture

- **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

**Note:** Very little standardization in hypervisor technologies (e.g. live migration, config, APIs)
## Industry Standards

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

### 802.1 Overlay Networking Projects, Cisco Innovations:
- FEX Architecture

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

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

### Controller Working Groups:
- PCE, FORCES

### Open Source Cloud Computing project
- Open Network Research Center at Stanford University

<table>
<thead>
<tr>
<th>Org</th>
<th>Cisco Leadership/Contributions for Open &amp; Programmable Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONF</td>
<td>• Harden OpenFlow spec</td>
</tr>
<tr>
<td></td>
<td>• Technical Advisory Group Chair</td>
</tr>
<tr>
<td></td>
<td>• Hybrid Group Chair</td>
</tr>
<tr>
<td>Openstack</td>
<td>Major contributions to Quantum networking APIs</td>
</tr>
<tr>
<td>IETF</td>
<td>Programmability: IRS</td>
</tr>
<tr>
<td></td>
<td>Overlay protocols: LISP, OTV, VXLAN</td>
</tr>
</tbody>
</table>

© 2011 Cisco and/or its affiliates. All rights reserved.
Open Network Environment – Flexibility to Choose
Protocols, APIs and Deployment Models

OnePK Developer Environment

- **Element**: Element Capabilities, Configuration Management, Interface/Ports Events, Location Information
- **Utilities**: Syslog Events and Queries, AAA Interface, Netflow Events, DHCP Events
- **Discovery**: Network Element Discovery, Service Discovery, Topology Discovery
- **Developer**: Debug Capabilities, Tracing Interfaces, Management Extensions
- **Policy**: Interface Policy, Interface Feature Policy, Forwarding Policy, Flow Action Policy
- **Routing**: Protocol Change Events, RIB Table Queries

OpenStack
- Quantum API: Interface descriptions, L2 network provisioning, L3 and IP Addr. Mgmt. - coming

OpenFlow
- Packet classifiers, Marking, Copy/Punt Inject, Statistics

RICHNESS OF FEATURES
Developer portal
Training & Certification
ISVs

© 2011 Cisco and/or its affiliates. All rights reserved.
Top 5 Takeaways: Cisco Open Network Environment

1. Flexible programmability models with scale and performance
2. No compromise security
3. Consistency across physical and virtual environments
4. Multi-hypervisor, multi-protocol, multi-layer
5. Open Standards
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 “Slicing”

Network Flow Management

Scalable Multi-Tenancy

Agile Service Delivery

Private Cloud Automation

Diverse Programmability Requirements Across Segments
Most Requirements are for Automation & Programmability
Network Programmability Models
Physical or Virtual

Current switch/router

Control Plane

Data Plane

Resilient, Scalable, Secure, Rich Features, Evolutionary, Investment Protection

“SDN” Approach

Control Plane

Data Plane

Simpler (fewer nodes to manage) Centralized Topology View

Hybrid Model?

Control Plane

Data Plane

Vendor-specific APIs

Vendor-specific APIs

Vendor-specific APIs

OpenFlow or Vendor-specific

Combined Benefits

* Standards based over time

Openstack & Network Overlays apply to all models (physical / virtual)
Evolution of the Intelligent Network

Preserve What’s Working

- Resiliency
- Scale and Security
- Rich feature-set

Evolve for Emerging Requirements

- Operational Simplicity
- Programmability
- Application aware

Evolve the Network for the Next Wave of Application Requirements
Implementing Customer Use Cases

**Approach 1**
- APIs
- Network
- Tightly-coupled HW & SW

**Approach 2**
- Controller
- Other Agents
- OpenFlow Device
- Device w/ OpenFlow
- Device
- Loosely-coupled HW & SW

**Approach 3**
- Virtual Overlays
- Physical and Virtual
- Logical/overlay Networks

Cisco Approach: Flexibility to Choose—The Power of “AND”
The News: Cisco Open Network Environment

1. Platform APIs
   - onePK (One Platform Kit)
     Comprehensive API & Developer Kit
   Platforms: IOS, IOS-XR, NX-OS

2. Controller/Agents
   - Controller software for SDN research
   - OpenFlow v1.0 Support on Catalyst 3750-X and 3560-X

3. Virtual Overlay Networks
   - Nexus 1000V
     OpenStack and REST API
     Multi-Hypervisors
     VXLAN Gateway
     Security, Services Chaining

Industry’s broadest approach for Network Programmability
Virtual DC / Virtual Overlays

Nexus 1000V – Virtual Networking Platform
Multi-services, Multi-hypervisor, Multi-cloud
Cisco Virtual Networking and Security Solution

*Nexus 1000V, CSR 1000V, ASA 1000V, VSG, and vWAAS Deployment*

**Virtualized/Cloud Data Center**
- Nexus 1000V: Distributed switch, LAN connectivity
- CSR 1000V: WAN gateway, Routing and VPN
- ASA 1000V: Edge firewall, WAN-to-LAN traffic
- VSG: VM-level controls, Zone-based FW
- vWAAS: WAN optimization, Application traffic
- vACE: SLB, App Delivery

**Physical Infrastructure**
- WAN Router
- Switches
- Servers

**Virtual Infrastructure**
- vPath
- VXLAN
- Nexus 1000V
- Multi-Hypervisor

**Tenant A**
- CSR 1000V
- ASA 1000V
- VSG
- vWAAS

**Department A**
- Nexus 1000V
- VSG

**Department B**
- VSG

© 2011 Cisco and/or its affiliates. All rights reserved.
Virtual Overlay Networks – Extending the lead
Scalable Multi-tenant Cloud Infrastructures – foundation for Secure Hybrid cloud

Nexus 1000V
VXLAN, vPath

OpenStack Quantum API
REST API

Any Hypervisor

VXLAN Gateway

Physical
(VLAN)
Network

ASA 1KV
VSG
vWAAS

Tenant 1
Tenant 2
Tenant 3

Virtual
Workloads

Physical
Workloads

ASA 55xx

Nexus 1000V enhancements
• Multi-hypervisor: VMware, Microsoft, Citrix, RedHat
• OpenStack/REST API
• VXLAN Gateway

Secure Consistent Experience Across Physical and Virtual Environments
## Reference Solutions

<table>
<thead>
<tr>
<th>Solution</th>
<th>Nexus 1000V</th>
<th>Nexus 1010</th>
<th>Virtual Security Gateway</th>
<th>Virtual WAAS</th>
<th>NAM (N1010)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vblock</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>FlexPOD</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtual Desktop</td>
<td>✔</td>
<td></td>
<td>Implicit Support</td>
<td>✔</td>
<td>Implicit Support</td>
</tr>
<tr>
<td>Virtual Multi-tenant DC (VMDC)</td>
<td>✔</td>
<td></td>
<td>Implicit support</td>
<td>✔</td>
<td>Implicit support</td>
</tr>
<tr>
<td>DC-to-DC vMotion</td>
<td>✔</td>
<td></td>
<td>Implicit support</td>
<td>✔</td>
<td>Implicit support</td>
</tr>
<tr>
<td>PCI 2.0</td>
<td>✔</td>
<td></td>
<td>Implicit support</td>
<td>✔</td>
<td>Implicit support</td>
</tr>
<tr>
<td>Hosted Collaboration</td>
<td>✔</td>
<td></td>
<td>Implicit support</td>
<td></td>
<td>Implicit support</td>
</tr>
</tbody>
</table>

*Based on default Citrix configuration*
Controller & OF Switches
Cisco Controller: Overview

- Controller: JAVA based, runs on any LINUX Server Appliance (e.g. UCS)
- Level 1 API: OF 1.x
- Level 2 API: Cisco App
- Level 3 API: OSGI, WebSockets, REST
- Northbound API: Cisco App, User App, 3rd Party App
- Southbound API: Dykstra SPF, Topology Manager, Slice Manager, ARP Handler, Forwarding Rules Manager, L3 Intf, Host Tracker, Device Manager, Infrastructure (Core)

- Cisco Enhanced: Capabilities Exch., Topology Disc., Neighbor info, other differentiations

All levels of API available to Apps for interaction with Controller.
Use Case: Campus Network “Slicing”

Consistent Policy Management for Maximum Flexibility and Innovation

Solution

- OpenFlow experimental support (v1.0)
- Experimental controller software
- Integrated slicing management
- Programmatic Interfaces (Eg. REST)
onePK: One Platform Kit
Introducing One Platform Kit - onePK

Flexible development environment to:
- Innovate
- Extend
- Automate
- Customize
- Enhance
- Modify
Introducing One Platform Kit – onePK
Enhanced Interactions with the Network Operating System

Traditional Interactions

Network OS

Events

Actions

App EEM (TCL)

+ onePK

App

C

Java

Anything you can think of
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 API Libraries

Initial Service Sets

**Element**
- Element Capabilities
- Configuration Management
- Interface/Ports Events
- Location Information

**Utilities**
- Syslog Events and Queries
- AAA Interface
- Path Trace

**Discovery**
- Network Element Discovery
- Service Discovery
- Topology Discovery

**Developer**
- Debug Capabilities
- Tracing Interfaces
- Management Extensions

**Data Path**
- Packet/Flow Classifiers
- Copy/Punt/Inject
- Statistics

**Policy**
- Interface Policy
- Interface Feature Policy
- Forwarding Policy
- Flow Action Policy

**Routing**
- Read RIB Routes
- Add/Delete Application Routes
- RIB Events (Route up/down)
Yes, onePK is Secure

Security Five Ways

- Code Isolation
- Strong Typing
- AAA (PKI)
- Encryption (TLS)
- App Security
- Admin Security
- Digital Signing Certification Process
- CLI Control Resource Allocation
- Isolation Resource Consumption
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
Example #1: Custom Routing
Routing for Dollars: Application driven routes installed in network
Example #2: Custom 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.
Thank you.