# The Future of Networks:
# Software Defined Networking

Kevin Woods

Network Operating Systems Technology Group, Cisco Systems Inc.

Sept 26, 2012

1

Market Transitions Driving Greater Demands on the Network

Market Transitions

Cloud Computing

Massive scale

Mixed physical and virtual topologies

Dynamic and distributed applications

How to Harness Network Value

How to drive business Agility?

How to drive Operational Simplicity?

IS THE NETWORK READY ?
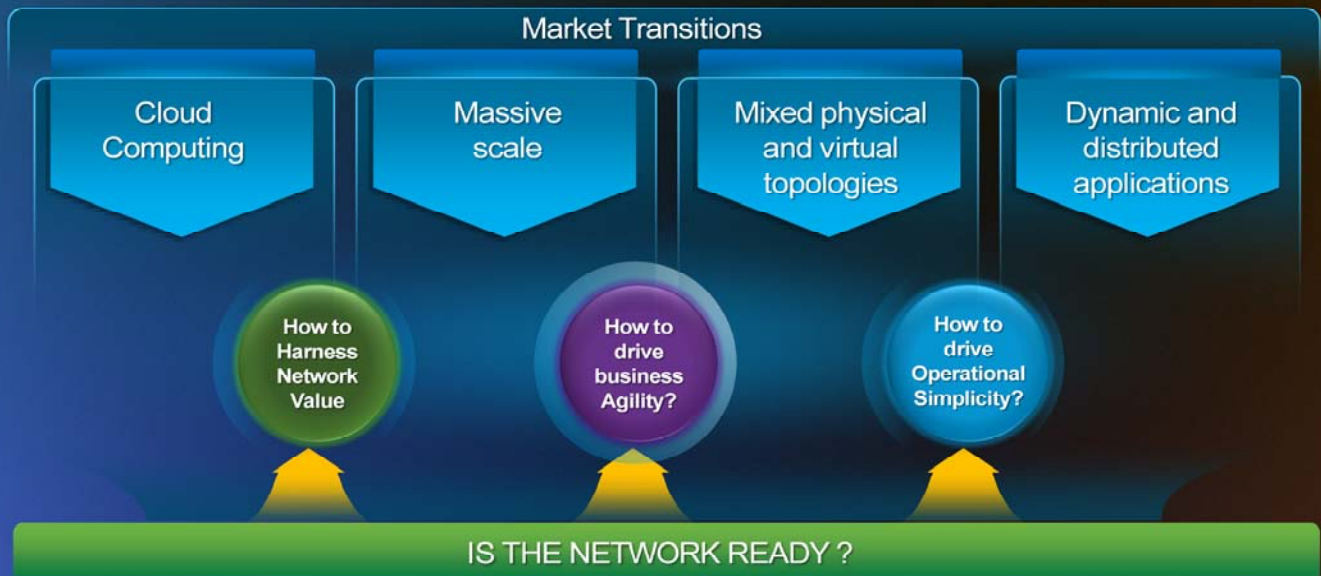
# Towards an Open Network Environment for SDN
## Implementation Perspective: Evolve the Control-Plane Architecture



Traditional Control Plane Architecture
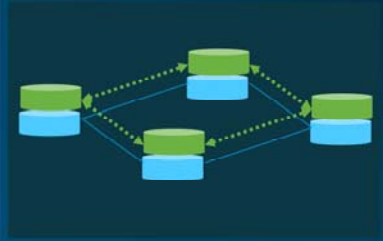
Evolved Control Plane Architecture (Examples)

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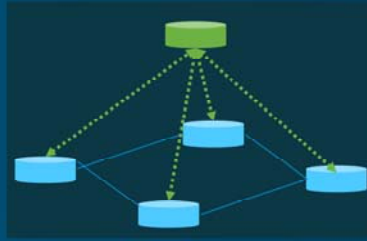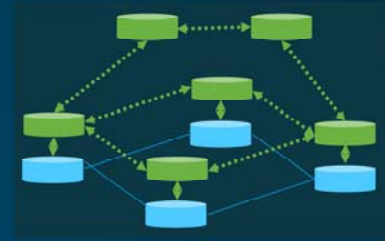
- Enable modularization and componentization of network control- and data-plane functions, with associated open interfaces. This allows for optimized placement of these components (network devices, dedicated servers, application servers) and close interlock between applications and network functions.

- 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

Control/Network/Services-plane component(s)    Data-plane component(s)

# Towards Programmatic Interfaces to the Network
## Approaching Today's Application Developer Dilemma

- Many Network Applications today:
  - OTT – for speed and agility
  - Avoid network interaction – complex and slow innovation
- New Model for Network Applications
  - Keep speed and agility
  - Full-duplex interaction with the network across multiple planes – extract, control, leverage network state

App — "Fast" — App

"Slow"    "New"

CLI(s)

Service
Edge
Appliance
Service
CPE
Service
Service
Core
Mobile
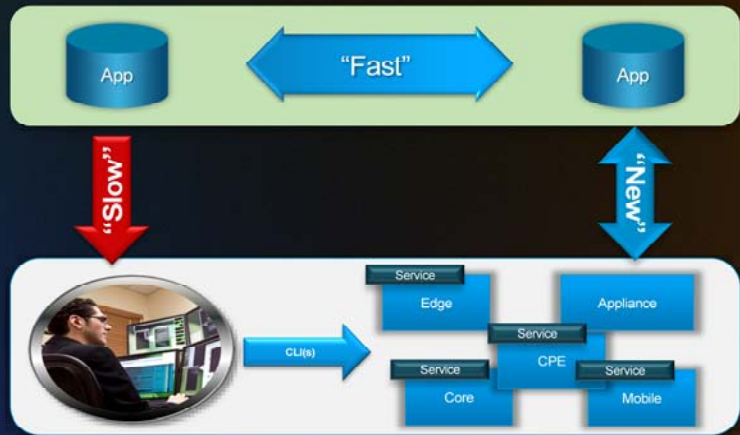
**A New Programming Paradigm is Needed**

5

# Market Requirements:
## What Our Customers Are Asking Us For

| Research/Academia | Massively Scalable Data Center | Service Providers | Cloud | Enterprise |
|---|---|---|---|---|
| • Standards-compliant<br>• Network partitioning "slices" for experimentation | • Drive better server utilization<br>• Deeper insight to network traffic<br>• Customize network device behavior | • Rapid services introduction<br>• Better SLA management<br>• Monetization<br>• Manage and optimize WAN topology<br>• Orchestration | • Multi-tenancy at scale<br>• Automation<br>• Deeper insight into network traffic<br>• Physical and Virtual services | • Investment protection<br>• Reduced complexity<br>• Business agility<br>• Leverage network value<br>• Physical and virtual |

## Diverse requirements with similar motivations

6

# Industry Standards & Forums

**802.1 Overlay Networking Projects** — IEEE

ONRC RESEARCH — Open Network Research Center at Stanford University

**Technical Advisory Group Chair, Working Groups:** Config, Hybrid, Extensibility, Futures/FPMOD/OF2.0 — ONF OPEN NETWORKING FOUNDATION

W3C

ITU International Telecommunication Union

openstack — **Initiatives** Quantum, Donabe, OpenStack Quantum Plugin, API for Nexus, OpenStack Extensions

Open Source Cloud Computing project — cloudstack open source cloud computing

IETF — **Overlay Working Groups:** NVO3, L2VPN, TRILL, L3VPN, LISP, PWE3 **API Working Groups/BOFs** NETCONF, ALTO, CDNI, XMPP, SDNP, I2AEX **Controller Working Groups:** PCE, FORCES

MEF

# Common Concepts – The Open Network Environment
## Approaching a Definition

- 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

**Open Network Environment**

| Programmatic APIs | Agents and Controllers | Network Virtualization Infrastructure |

Simplified Operations

| Enhanced Agility | Network Monetization |

8
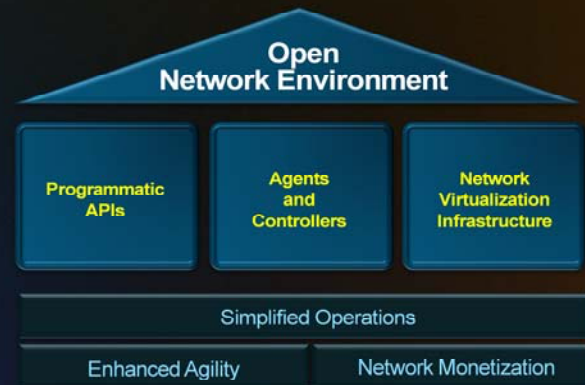
Exposing The Entire Network Value
Programmability at multiple layers of the network

Application Developer Environment

Orchestration

Analysis and Monitoring, Performance and Security

Network Services

Control Plane

Forwarding Plane

Network Elements and Abstraction

Transport

Harvest Network Intelligence

Network Planes

Program Policies for Optimized Experience
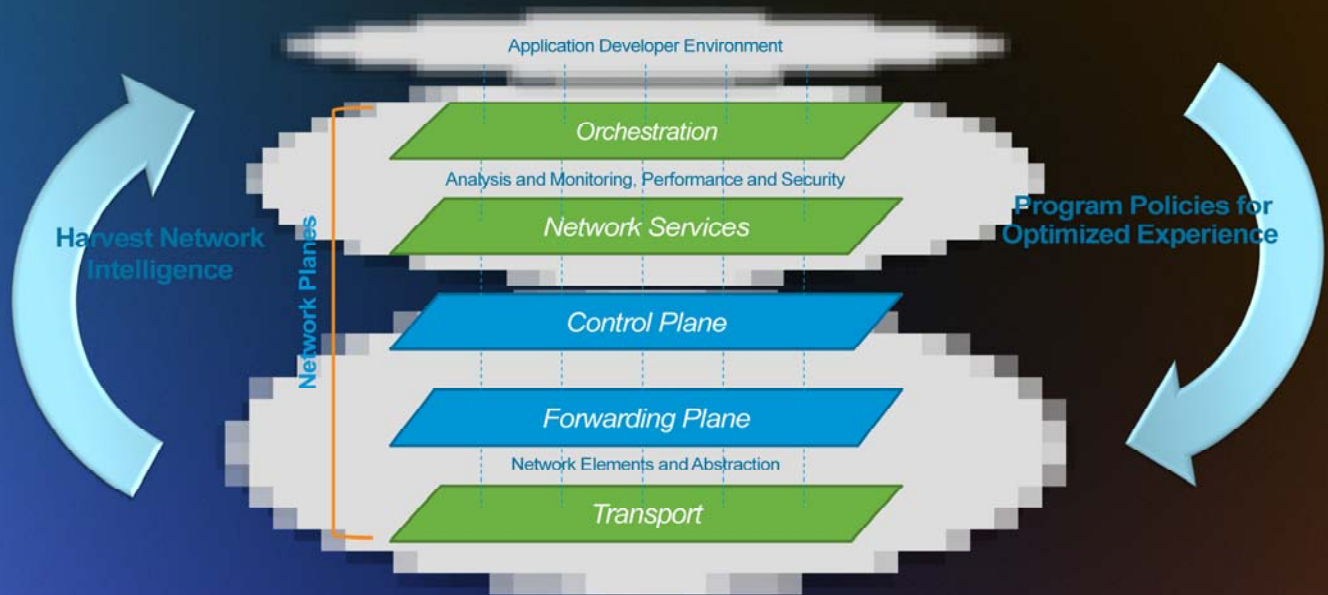
# Programmatic Network Access – Multiple Layers

**Full-Duplex access to the network at multiple layers and networking planes**

- 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 APIs)
- Multiple Language Support (C, Java, …)
- Integrate with customer development environment to deliver enhanced functionality
- Reduced time to market by leveraging common platform for building services

**Harvest Network Intelligence**

**Program for Optimized Experience**

| Layer | Description |
|---|---|
| Application/Development | *Application development frameworks* |
| Management | *Programmatic network automation* |
| Orchestration | *Automated, policy directed service and cloud management, e.g. NetworkService Manager, OpenStack, …* |
| Network Service | *Network wide service access: Optimized paths (PCE), Topology & service selection (NPS/ALTO), MediaTrace, Address mapping, ..* |
| Control | *Common control abstractions: Security, Policy, Routing, ..* |
| Forwarding | *Common forwarding abstractions: Data-Path access, Flow-Forwarding, Tunneling, ..* |
| Transport/Device | *Device configuration, state monitoring, logging, debugging* |

## SDN Engineers!

- Bachelors in Computer Science or related field
- Experience with network devices (routers, switches, load balancers) and a good understanding of network routing protocols
- Proficient in Python and Java
- Solid understanding of the Linux operating system
- Experience with Open Source tools for provisioning or orchestration such as OpenStack, Chef or Puppet
- Experience developing control plane software for switches or routers is preferred
- In-depth understanding of the TCP/IP stack
- Ability to debug issues within custom code quickly
- A passion for developing scalable and optimized software systems

11