TOMORROW
starts here.

Cisco Connect
Praha, Česká republika
24.–25. 3. 2015
NFV pro poskytovatele služeb
Tech-SP2

Jiri Chaloupka - Systems Engineer
CCIE# 39800
Virtual Networking 101 – Why Need it?

1. 70% of workloads virtualized today

2. vMotion moves VMs across physical ports—the network policy must follow vMotion in a timely fashion

3. Must view or apply network/security policy to locally switched traffic

4. May need to maintain segregation of duties while ensuring non-disruptive operations
Market Trends
Industry/Market Trends Affecting Virtual Networking

- Multi-Hypervisor, Container, Opensource
- Increasing Physical/Virtual Integration in Networking
- L4-7 Market in Transition with Virtual Services
- New Markets (SP) for Virtual Networking
- Increasing Adoption of Public and Private Clouds
- Hybrid Cloud gaining Momentum

Virtual Networking is Key Component of Data Center and Cloud
Industry Initiatives and Terminology
Key Industry Initiatives (1)

SDN, Openflow, Opendaylight

“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 protocol that specifies **interactions between de-coupled control and data planes**…….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/

**Open source** project formed by industry leaders and others under the Linux Foundation. “…OpenDaylight’s mission is to facilitate a community-led, industry-supported open source framework, including code and architecture, to accelerate and advance a common, robust Software-Defined Networking platform…”

http://www.opendaylight.org/
Open source software for building public and private Clouds; includes **Compute (Nova)**, **Storage (Swift)** and **Networking (Neutron)** services.

http://www.openstack.org

“Open vSwitch (OVS) is a production quality open source software switch designed to be used as a vswitch in **virtualized server environments**.......Open vSwitch supports standard management interfaces (e.g. sFlow, NetFlow, IPFIX, RSPAN, CLI), and is open to programmatic extension and control using **OpenFlow and the OVSDB management protocol**”

http://openvswitch.org

“**NETCONF** is an IETF **configuration management protocol** defined in RFC 6241. Provides multiple operations for interacting with configuration & operational data. **YANG** is a **modeling language** defined in RFC 6020. Used by NETCONF to define objects and data in requests & replies. **TOSCA** is an example of another Modeling Language coming from the OASIS initiative
Network Consumption is Evolving

Box-Box configuration to policy based automated controller based networking

Minimal Automation

ACI

Physical + Virtual full integrated solution

SW Overlay with HW GW

Virtual overlay over standalone hardware. Need HW gateways for bare-metal servers

Managing virtual resources are important element of this transition
Network Functions Virtualisation (NFV) – Initial Goals

NFV = Transition of network infrastructure services to run on virtualised compute platforms – typically x86

Enablers
- Hypervisor and cloud computing technology
- Improving x86 h/w performance
- Optimised packet processing and coding techniques
- Network industry standardising on Ethernet
- SDN based orchestration

Value Proposition
- Shorter innovation cycle
- Improved service agility
- Reduction in CAPEX and OPEX

Applications
- Potentially all network functions
## A. Perceived Benefits for NfV - Architecture

<table>
<thead>
<tr>
<th>Motivation</th>
<th>Description</th>
</tr>
</thead>
</table>
| Reduction of the number of network elements to manage and deploy | • **Integration of network functions** into a single system reduces the number of appliances / NE to manage / configure  
• Fewer hardware types to deploy / plan for  |
| Service Elasticity                             | • Deployment of VMs much faster than appliances  
• Easy **scale up / scale down** of services  
• Flexible service portfolio (mixing VNFs) |
| Operational efficiencies through virtualization | • Can leverage virtualization advantages from data center (vMotion, dynamic resource scheduling, power management etc) also for VNFs |
| Reduced complexity for High Availability       | • VMs have a **smaller failure domain.**  
• **Stateless deployments** become more acceptable, so less complexity through stateful redundancy deployments  
• ISSU simplified by deploying a NEW VM and failing over |
## B. Perceived Benefits for NfV - CAPEX

<table>
<thead>
<tr>
<th>Motivation</th>
<th>Description</th>
</tr>
</thead>
</table>
| Deployment of standard **x86-based servers** | • Servers considered **cheaper** than routers / appliances  
• Servers already deployed in branch / DC / PoP |
| Cost reduction through **economies of scale** | • Deployment of huge **server farms** in DCs can lead to better resource utilization |
| Simplified Performance Upgrades           | • Capability to increase performance without forklift upgrades               |
## C. Perceived Benefits for NfV - OPEX

<table>
<thead>
<tr>
<th>Motivation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction of branch visits</td>
<td>• Changes / upgrades in the service can be made in software</td>
</tr>
<tr>
<td></td>
<td>• No longer need to swap appliances on-site for service upgrades, appliance failures</td>
</tr>
<tr>
<td>Automated network operations</td>
<td>• Virtualization places focus on automation and elasticity, thus reducing management</td>
</tr>
<tr>
<td>Flexible VNF-based operation</td>
<td>• <strong>Software upgrades</strong> can be done independently per VNF</td>
</tr>
<tr>
<td></td>
<td>• VNFs can be placed flexibly in branch, PoP or DC</td>
</tr>
</tbody>
</table>
Network Functions Virtualization in Enterprise/SP

Traditional Data Center/ Managed Service

Network Appliance (box) per function on vendor provided hardware. Slow innovation and slow to deploy

Services with NFV

Virtualized network functions easily orchestrated on any server. Fast scale up/scale down and also available in aaS offering
## Service Provider NFV Use Cases by ETSI

<table>
<thead>
<tr>
<th>ETSI Formalized NFV Use Cases</th>
<th>Potentially Virtualized Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Functions Virtualization Infrastructure as a Service</td>
<td>vNAT, vFW, vLB, vRR, vVPN, vRouter</td>
</tr>
<tr>
<td>Virtual Network Function as a Service (VNFaaS)</td>
<td>vCPE, vPE</td>
</tr>
<tr>
<td>Virtual Network Platform as a Service (VNPaaS)</td>
<td>vPrivateCloud</td>
</tr>
<tr>
<td>VNF Forwarding Graphs</td>
<td>VPE-F,</td>
</tr>
<tr>
<td>Virtualization of Mobile Core Networks and IMS</td>
<td>vEPC (vS/P-GW, vMME, vPCRF, vSGSN, vGGSN, vGiLan) vIMS (vP/S/I-CSCF, vMGCF, vAS)</td>
</tr>
<tr>
<td>Virtualization of Mobile Base Station</td>
<td>vMAC, vRLC, vPDCP, vRRC, vCOMP, vBBU</td>
</tr>
<tr>
<td>Virtualization of the Home Environment</td>
<td>vBNG, vRGW, vSTB</td>
</tr>
<tr>
<td>Virtualization of CDNs</td>
<td>vCDN,</td>
</tr>
<tr>
<td>Fixed Access Network Functions Virtualization</td>
<td>vOLT, vDSLAM, vONU, vONT, vMDU, vDPU</td>
</tr>
</tbody>
</table>

Source: [http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf)
ETSI NFV Terminology

- **NF**: A Network Function (NF) is a building block within an operator’s network infrastructure, which has well defined external interfaces and a well defined functional behaviour. In practical terms a Network Function is today often a network node.

- **VNF**: A Virtual Network Function (VNF) provides exactly the same functional behaviour and interfaces as the equivalent Network Function, but is deployed in a virtualised environment.

- **NFVI**: The NFV-Infrastructure (NFVI) is the totality of all hardware and software components which build up the environment in which VNF are deployed, managed and executed.

- **NFVO**: The NFV-Orchestrator (NFVO) is a software to operate, manage and automate the distributed NFV Infrastructure. The Orchestrator has control and visibility of all VNF running inside the NFV-Infra.

- **VNFM**: The VNF Manager lifecycle management of VNFs and the associated NFVI resources

- **VIM**: The Virtualised Infrastructure Manager manages the NFVI components and specialist VIMs are permitted (e.g. compute, storage and network)
Cisco Overall Strategy

Service Orchestration
Automation, provisioning and interworking of physical and virtual resources

NFV
Network functions and software running on any open standards-based hardware

SDN
Control & Data Plane separation…Centralised Control…abstraction & programmability

Traditional
Distributed control plane components, physical entities

Cisco Is Executing on Plan to Integrate All
Domain Specific Controllers

Cross-Domain Orchestration is key element

- A domain is a function or set devices where the management may be performed by a domain controller
- Domain include but not limited to:
  - Compute
  - Storage
  - Network services
  - CPE
  - VPN
- Controller’s can be shared between domains or completely independent entities
Cisco Networking Architectural Evolution

Agility
Optimize
Revenue

Video
Business
Cloud
Mobility

Evolved Programmable Network

Evolved Services Platform

Applications and Services

APIs

Service Orchestration Apps
Service Catalog
VM / Storage Control

Seamless Experience

Policy
Dynamic Scale

Real-Time Analytics
Fully Virtualized
On-Demand Services Anywhere

Automated
Always “ON”
Intelligent Convergence

Application Interaction

Agility

Optimize

Revenue

Video

Business

Cloud

Mobility
Cisco Networking Architecture Evolution

Applications, ESP and EPN

Applications & Services

Evolved Services Platform

Evolved Programmable Network

Applications

- Service Provider Applications – OSS/BSS, Orchestration etc.
- End User Applications
- External ISPs / Content Providers

Controller NB APIs

Controller/Orchestration

- Cross-Domain Orchestration – multi-domain – E2E service
- Controllers – Multi-Domain i.e. network, compute, storage
- Service Provisioning – Configuration and management

Device/Network Level APIs

Network

- Simplified distributed control plane
- Augmented by central control
- Integrated Packet/Optical/Virtual transport

Applications

Evolution:

1. Applications & Services
2. Evolved Services Platform
3. Evolved Programmable Network
Cisco Networking Architecture Evolution

APIs/Protocols – Consolidation is happening
Evolved Service Platform (ESP)
Cisco Products contained within ESP
Cisco ESP/EPN Architecture

High-Level Mapping to SDN and NFV

Network Functions Virtualisation
(NFVO, VNF Manager, Virtual Infrastructure Manager [compute, storage, network])
Evolved Programmable Networks (EPN)
Network Simplification, Optimisation and Automation

- IP and/or MPLS Data plane
- Control Plane Simplification
- Network Programmable
- Virtualization and Cloud
- Service Chaining/Forwarding
- IP/Optical integration
Network Functions Virtualization

Network Requirements and today's approaches

- **OSS/BSS, subsystem and N/W control**
- **Wireless Gateways**
- **Wireline GWs**
- **Core, Metro switching**
- **Appliances (L4-L7)**
- **Business CPE**
- **Home CPE**

- **CPU**
- **Variable CPU / FPGA / NPU**
- **Distributed: Lots CPUs + NPUs**
- **Distributed: CPUs + Lots of NPUs**
- **Distributed: CPUs + NPU**
- **Centralized: CPU + NPU**
- **Centralized: CPU or SoC**
- **Variable CPU / FPGA / NPU**

**CPU Reqs**
- High
- Low

**Bandwidths**
- 0
- 10Mbps
- 100Mbps
- 1Gbps
- 10Gbps
- 100Gbps
- 1Tbps
- 10Tbps
- 100Tbps
- 1Pbps
Cisco’s Comprehensive VNF Offerings

**Network Infrastructure**
- Available
  - Virtual CE / CPE Router (CSR1Kv)
  - Nexus 1000V
  - VPP based vPE-F
  - Network Analysis Module (NAM)
  - AppNav and AVC (CSR1Kv)

**Gateways**
- Available
  - Virtual BRAS (CSR1Kv)
  - DHCP (CSR1Kv)
  - IP SLA (CSR1Kv)
  - VXLAN (L2,L3), OTV, VPLS, LISP (CSR1Kv)
  - Virtual SGW / SGSN (QvPC)
  - Virtual PGW / GGSN (QvPC)
  - eWAG/ ePDG (QvPC)

- Available (QvPC-SI)
  - Virtual BGP Flowspec Controller (XRv)
  - Virtual IOS
  - Virtual NX-OS
  - Flex-XR Virtual Router
  - CML / VIRL
  - DDoS
Cisco NFV Orchestration Solution

Current Architecture

- **Prime Service Catalog (PSC)**
- **SP's OSS/BSS or Prime Order Fulfillment**
- **Network Services Orchestrator (NSO)** (Based on Tail-F NCS)
- **Elastic Services Controller (ESC)**
- **OpenStack**
- **VM & Storage Orchestrator**
- **Virtual Topology Controller (VTC)**
- **Virtual Topology Forwarder (VTF)**
- **SDN System/Controller**
- **OpenStack APIs**
- **OVS**
- **VNF**
- **DCI**
- **ESC API**
- **REST API**
- **Netconf/Yang**
- **MP-BGP**
- **Restconf**
- **JCloud (Future)**
- **Virtual Topology System (VTS)**

A Framework enabled by multiple products & architecture

SDN subsystem / SDN Controller

SDN Virtual Forwarder

OSS

Virtual Topology Controller (VTC) + Virtual Topology Forwarder (VTF) = Virtual Topology System (VTS)
Cisco NFV Orchestration Solution

NSO - Declarative Model, Real time handling, Single network-wide interface

Cisco Network Services Orchestrator (NSO)
- Service Manager
- Device Manager
- Network Element Drivers (NED)

- Mgmt. Apps / OSS/BSS or Tenant Portal
- Network Engineer
- REST, Java, NETCONF
- Network-wide CLI, Web UI

Life Cycle management – Create, Modify, Delete

End-to-End Transactions Configurations

Service YANG Models

Device YANG Models
NSO Main Feature 1: Model-Based Architecture

YANG data models for
- Network services
- Network topology
- Network devices

YANG data models drive
- Northbound APIs
- User interfaces
- Southbound command sequence

Benefits:
- Can be used for all types of services and all types of networks
Transactional guarantees:

- Help ensure fail-safe operations (automated handling of exceptions)
- Keep accurate copy of network configuration state in NSO at all times

Benefits:

- Automation can be based on accurate real-time view of service and network state
- Much higher degree of automation possible
NSO Main Feature 3: FastMap* Algorithm

**FastMap:**
- Only the CREATE operation needs to be specified
- UPDATE, DELETE and REDEPLOY operations are automatically generated and compute minimal change set needed

**Benefits:**
- Reduces service implementation code by two orders of magnitude
- Supports modifications of services at runtime

* Patent No.: US 8,533,303 B2
NSO – Function Packs

Initially
- Mobility
- Virtualized Managed Services

- **Function Pack** is a package that addresses a specific use-case and is built on top of NSO.
- Function Packs are not part of NSO and can be added at run-time.
- Function Packs are customizable by customers and system integrators
Service Designer for creating Service Models

- The service designer offers a Palette with High Level Resource Facing Service primitives and a Canvas for a user to drag-n-drop items from the Palette and assemble them, which, creates the service blueprint.
Cisco NFV Orchestration Solution
YANG Model Driven Methodology in Practice – virtual Managed Services example
ESC Modularity

Elastic Services Controller

- vSphere
- Public Cloud (AWS, Azure)
- Openstack/KVM* Ubuntu
- Linux Containers

Elastic Services Controller

- VNF Lifecycle management
- Service Monitoring, Elasticity and Recovery

API / Netconf / Yang

Northbound Orchestration System

- Cisco Network Services Orchestrator
- Openstack Heat Orchestration
- Any 3rd Party NFV Orchestrator
- API and Yang Model driven Integration

* Available in subsequent releases
Cisco NFV Orchestration Solution

ESC functionality

Elastic Services Controller

List of Events
- VM Alive
- Service Alive
- Upper load threshold crossed
- Lower load threshold crossed
- Service Dead
- VM Dead

List of Actions
- Notify (callback)
- Advertise Service
- Withdraw Service
- Restart VM
- Scale up (add a VM)
- Scale down (remove a VM)
- Individually customizable action(s) for every event

Simple Rules
- Service Alive => Advertise
- VM Dead => withdraw
- Upper load => scale up

Complex Rules
- Service Alive => Advertise, Notify
- Upper load => Scale up, Notify, Advertise
- Service Dead => Withdraw, Notify, Restart

Agent and Agentless mode supported
Cisco NFV Orchestration Solution

Current Architecture

Prime Service Catalog (PSC)

SP’s OSS/BSS or Prime Order Fulfillment

User Self-Service Portal

NFV Orchestrator

VM and Service Lifecycle Manager

VM & Storage Orchestrator

Network Services Orchestrator (NSO) (Based on Tail-F NCS)

Elastic Services Controller (ESC)

OpenStack

SDN System/Controller

Virtual Topology Controller (VTC)

Virtual Topology Forwarder (VTF)

DCI

OSS

A Framework enabled by multiple products & architecture

SDN sub-system / SDN Controller

SDN Virtual Forwarder

OVS

VNF

VTF

Virtual Topology Controller (VTC) + Virtual Topology Forwarder (VTF) = Virtual Topology System (VTS)
Cisco NFV Orchestration Solution
Virtual Topology System (VTS) - Software Overlay based Forwarding Functionality

- VTS = VTC (Controller) + VTF (Forwarder)
- VTC: programs VTF via Restconf/Yang
  (Specification Published in IETF)
- VTC: Use IOS-XRV for route distribution (BGP)
- VTF: Software overlay based on Cisco VPP
- VTF: L2/L3 support (v4 and v6)
- VTF: Multi-tenanted (Control and Data)
- VTF: Multiple Encapsulation MPLSoGRE, VXLAN
- VTF: Performance ~ 10G per Core
- DCI (Data Centre Interconnect): Connects network with DC tenants/NFV PODs

Cisco VPP = Vector Packet Processing
VTF Technology – Vector Path Processing (VPP)

- vPE-forwarder is based on Cisco’s VPP technology

- What is Vector Packet Processing?
  - Highly optimized packet processor for general-purpose CPU’s
  - Very fast
    - Constructs super frames of packets and processes them in one shot - exploits temporal locality of application flows. Benefits from I-cache, D-cache hits.
    - Direct PCI pass-through allows send/receive packets with zero operating system overhead
    - near line rate processing on 10G interfaces
  - 64-bit, multi-threaded
  - Portable
  - VPP is a user space process - fault protected & easy upgrades
  - Multi-tenant forwarding contexts for IPv4 and IPv6
  - Shipping on several Cisco products (ASR 9000, CGSE, CSR1000v)
Cisco NFV Orchestration Solution

Current Architecture

Prime Service Catalog (PSC) → REST API → SP’s OSS/BSS or Prime Order Fulfillment

Network Services Or orchestrator (NSO) (Based on Tail-F NCS)

- Elastic Services Controller (ESC)
- OpenStack
- VNF
- VTP
- OVS
- x86 Server
- SDN System/Controller
- Virtual Topology Controller (VTC)

SDN Subsystem / SDN Controller

Virtual Topology Controller (VTC) + Virtual Topology Forwarder (VTF) = Virtual Topology System (VTS)

A Framework enabled by multiple products & architecture

User Self-Service Portal

NFV Orchestrator

VM and Service Lifecycle Manager

VM & Storage Orchestrator

OpenStack APIs

ESC API

REST API / JCloud (Future)

Netconf/Yang

Restconf

MP-BGP
Adding VNFs in the cloud

Customer Orders VPN Service

- ISR CPE
  - ISR CPE Shipped to Customer Site, connected & Powered ON
- PnP Functionality Zero Touch Provisioning
  - Provide Day 1 Configuration

Tenant Portal

SP’s OSS/BSS

REST API

Network Services Orchestrator (NSO)

Elastic Services Controller (ESC)

Virtual Topology Controller (VTC)

OpenStack

PnP Server

Internet Gateway

If more VNFs are needed for a Service Chain?

More scalable and flexible service chaining enabled with VTC & high-performance VTF
Service Chaining Methods, Today
Static, Rigid, Complex ...

- Layer4-7 services - mostly physical appliances
  - Firewall, ADC, DPI, WOC, WAF, Monitoring, Various Optimizers, etc.

- Service chains are very primitive
  - Hop-by-hop steering techniques: VLAN stitching, PBR, etc.

- Service deployment, policies, tenancy, etc., are bound to topology
  - Provision VLANs across the network to insert a new service

- Service provisioning and capacity scaling is very static

- Troubleshooting service paths is a nightmare!
vPath Architecture Principles
Service Chaining for Virtual and Hybrid Environments

- Service Integration Architecture
  - Traffic steering on overlays
  - Infrastructure Programming
  - Metadata Communication
  - Multi-tenancy
  - Service scale-out
  - Dynamic Service Chains
  - Integrate vPath unaware services
  - OAM for visibility/troubleshooting
  - RESTful External Interface

* Not at IETF to avoid SDN and control plane discussion
**vPath 3.0 Service Function Types**

*Transparent And Terminated Service Functions*

**Transparent Service Function (No location)**
- Act on interesting traffic flowing through them. Require no redirection of traffic but all traffic must flow through them – typically inserted between switches.
- Eg.: Transparent Firewalls, Load balancers, Cisco SCE (DPI)

**Bump-in-a-wire**
- Service functions that are transparent to traffic, aka bridged or stealth mode operation. They have no location (IP or MAC) and simply bridge the traffic from an incoming interface to an outgoing interface.
- Eg.: Transparent Firewalls, Load balancers, Cisco SCE (DPI)

**Proxy Service Function**
- Service functions that are transparent to traffic. They have one or more locations (IP / MAC) associated with them. Require redirection of traffic in order to service them. They don’t alter the 5-tuple of the flows.
- Eg.: Cisco WAAS, Cisco VSG, Openwave Integra, Imperva WAF, etc.

**End-point Service Function**
- Service functions that are not transparent to traffic. They are deployed in the path of the traffic and hence traffic is destined to the location associated with them. Traffic (such as TCP flows) is terminated and re-originated within such services. They alter the five-tuple of the flows. Service functions include NAT, VPN, etc.
- Eg.: Cisco ASA, Citrix VPX, F5, etc.
vPath 3.0 Agent

Infrastructure Programming

- Abstracts the NSH or vPath 3.0 Header (VP3H) interactions via API
- Distributed as a shared library or a KLM, in binary form
- Requires service functions to integrate with the agent first
- Enables programming the forwarders or the infrastructure.
- Example: Simple Offloads, Pre-program 5-tuples with policy
**vPath 3.0 | IETF SFC Architecture**

_Built ground up for service chaining_

---

**Service Overlay**: The overlay network constructed with the vPath 3.0 encapsulation header (VP3H) and an overlay transport header.

**Service Control Plane**: Responsible for service chain configuration, service function discovery, service forwarder programming and management.

**Service Classifier**: Responsible for classifying the workload traffic, selecting a service path, adding the VP3H, and forwarding the encapsulated packet to the next service forwarder hosting the next service function on the service chain, thereby starting the service chain.

**Service Function Forwarder (SFF)**: Responsible for receiving a VP3H encapsulated packet and forwarding to the next local service function on the path, based on the service forwarding table. When all service functions are done, removes VP3H encapsulation and passes it to the network forwarder for forwarding on the underlay.

**Service Overlay**: The overlay network constructed with the vPath 3.0 encapsulation header (VP3H) and an overlay transport header.

**Service Classifier**: Responsible for service chain configuration, service function discovery, service forwarder programming and management.

**Service Control Plane**: Responsible for service chain configuration, service function discovery, service forwarder programming and management.

**Service Function Forwarder (SFF)**: Responsible for receiving a VP3H encapsulated packet and forwarding to the next local service function on the path, based on the service forwarding table. When all service functions are done, removes VP3H encapsulation and passes it to the network forwarder for forwarding on the underlay.

**Service Flow Table**: Maintains the flow state that includes, service paths, per-service decisions, etc.

**Service Function Table**: Describes the service functions, their attributes, location, transport encapsulation, etc.

**Service Forwarding Table**: Describes how to forward traffic along a service path, to each of the service functions in the path.

**Components**

1. Service Control Plane
2. Service Classifier
3. Service Forwarder
4. Service Overlay
vPath 3.0 (VP3H) | Service Encapsulation
Network Service Header (NSH) in IETF

<table>
<thead>
<tr>
<th>Overlay Header</th>
<th>00 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6 7 8 9 30 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NATIVE (0x894F), LLC/SNAP</td>
<td></td>
</tr>
<tr>
<td>VXLAN/VXLAN-GPE</td>
<td></td>
</tr>
<tr>
<td>GRE/NVGRE</td>
<td></td>
</tr>
<tr>
<td>UDP (port# 6633)</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Service Overlay Packet Format

- **Overlay Header**
- **Base Header**
- **Context Headers**
- **Network Service Header (NSH | VP3H)**
- **Original Payload**
  (packet/frame requiring services)

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base Header</strong></td>
<td>Provides SFC Path information and header de-referencing data</td>
</tr>
<tr>
<td><strong>Context Headers</strong></td>
<td>Provide the Metadata</td>
</tr>
</tbody>
</table>
Adding VNFs in the cloud

Customer Orders VPN Service

PnP Functionality
Zero Touch Provisioning

ISR CPE

ISR CPE Shipped to Customer Site, connected & Powered ON

PnP Server

Provide Day 1 Configuration

PnP Functionality
Zero Touch Provisioning

Elastic Services Controller (ESC)

Virtual Topology Controller (VTC)

OpenStack

Establish VPN: IPSec, IP Overlay (VXLAN, GRE, LISP), L2

CloudVPN Connectivity up

More scalable and flexible service chaining enabled with VTC & high-performance VTF

DCI/PE

Internet Gateway

Customer Orders VPN Service

Tenant Portal

SP’s OSS/BSS

REST API

REST API

More scalable and flexible service chaining enabled with VTC & high-performance VTF
OpenStack Cloud Computing Software

- Freely available, open source software allowing anyone to build their own private or public clouds.
- Open source and open APIs allows the customer to avoid being locked in to a single vendor.
- Built by a growing community of contributors.
- Opportunities for vendors to develop their own solutions and services.

OpenStack, a cloud management open platform
OpenStack Releases

- **Austin** – Oct 2010
- **Bexar** – Feb 2011
- **Catus** – April 2011
- **Folsom** – Sept 2012
- **Grizzly** – April 2013
- **Havana** – October 2013
- **Diablo** – September 2011
- **Icehouse** – April 2014
- **Essex** – April 2012
- **Icehouse** – April 2014
- **Kilo** – April 2015
- **Icehouse** – April 2014
- **Havana** – October 2013
- **Diablo** – September 2011
- **Essex** – April 2012
- **Kilo** – April 2015
OpenStack is “Project” Based

Core Projects Shown

<table>
<thead>
<tr>
<th>Compute</th>
<th>Storage</th>
<th>Dashboard</th>
<th>Identity</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Nova”</td>
<td>“Glance, Swift, Cinder”</td>
<td>“Horizon”</td>
<td>“Keystone”</td>
<td>“Neutron”</td>
</tr>
</tbody>
</table>
| - Houses VMs  
- API driven  
- Support for multi-hypervisors | - Instance/VM image storage  
- Cloud object storage  
- Persistent block level storage | - Web app for controlling OpenStack resources  
- Self-service portal | - Centralized policies  
- Tenant mgmt.  
- RBAC  
- Ext. integration (LDAP) | - Networking as a service  
- Multiple models  
- IP address mgmt.  
- Plugins to external HW |

<table>
<thead>
<tr>
<th>Telemetry</th>
<th>Orchestration</th>
<th>Database</th>
<th>Data Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Ceilometer”</td>
<td>“Heat”</td>
<td>“Trove”</td>
<td>“Sahara”</td>
</tr>
</tbody>
</table>
| - Central collection point  
- Metering and monitoring | - Template-based orchestration engine  
- More rapid deployment of applications | -DBaaS  
- Single-tenant DB within instance | - Fast provisioning of Hadoop clusters |
Who is Involved in OpenStack?

You name it – Compute, Storage, Networking vendors, Universities, Gov’t, OpenStack-specific startups
Traditional HW vendors – Cisco, HP, Dell, etc…
Providers – Rackspace, AT&T, Comcast, etc…
Startups – PistonCloud, SwiftStack and many, many more…
Distributions & Support – Red Hat, Canonical, SUSE
Cisco’s Focus on OpenStack - Today

Community
- Neutron – Network Service
- Horizon – Dashboard
- Keystone – Identity
- Swift – Object Storage
- Ceph/Cinder – Block Storage
- Automation – PuppetLabs
- HA Design

Engineering
- Cisco Product Integration
- Nexus Plugins – Neutron
- UCS
- CSR/ASR
- Co-developed solutions (Red Hat, Canonical, SUSE)

Customers
- Cisco Designs on specific releases
- Start simple, build from there – Focus on automation and HA
- Evangelization of what Cisco is doing - Thought Leadership – Help customers know What, When, Where & How
Cisco Top Vendor in Network Contributions

OpenStack Atlanta User Survey
OpenStack Architecture
Nexus Standalone Integration

- 1 Controller, 2 Compute nodes
- Separate Management (eth0) and Data networks (eth1)
- ToR switch connections and config
- Separate Nova availability zones
- Neutron Cisco plugin -> ML2 Driver
- Not running Neutron L3 agent on controller server
- VLAN range managed by vswitch plugin (ovs, n1kv)
- Supported on Grizzly or later releases
- Requires ncclient on control-server for NETCONF

*There is an additional linux bridge on the host which has not be shown for simplicity*
Nexus – Standalone - Nexus ML2 Driver

Features:

- Nexus 9000 running in standalone NX-OS mode
- Existing Nexus ML2 driver we use with other Nexus switches
- Dynamic VLAN provisioning for ethernet and port channel interfaces
- VXLAN gateway: VXLAN segment id configured and mapped to IP multicast groups
OpenStack Plugin Model

- Cisco plugin supports multiple sub-plugins
- Modular L2 (ML2) evolution of Neutron
- Allow multiple plug-ins to exist as sub-plugin drivers
Neutron Model

- **Tenant**
  - **Network**: external
  - **Subnet**
  - **Port**
  - **Security Group**
  - **Security Group Rule**

### Abstraction

<table>
<thead>
<tr>
<th>Abstraction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Network</td>
<td>L2 / Broadcast domain</td>
</tr>
<tr>
<td>Logical Router</td>
<td>L3 domain</td>
</tr>
<tr>
<td>Subnet</td>
<td>Subnet (OpenStack IPAM / DHCP)</td>
</tr>
<tr>
<td>Security Group</td>
<td>Group-based ACL</td>
</tr>
</tbody>
</table>
ML2 Architecture Diagram

- Neutron Server
- ML2 Plug-in
- Type Manager
- API Extensions
- Mechanism Manager
- GRE TypeDriver
- VLAN TypeDriver
- VXLAN TypeDriver
- Cisco
- Cisco Nexus
- Cisco Hyper-V
- Microsoft
- Layer 2 Population
- Linux Bridge
- Open vSwitch
- ...
Nexus 1000V Architecture

- **VSM**: Virtual Supervisor Module
- **VEM**: Virtual Ethernet Module

Diagram shows the architecture with modular switches, supervisor modules, linecards, backplane, hypervisors, and virtual appliances connected to a control plane.
Virtual Supervisor Module (VSM)
- Virtual or Physical appliance running Cisco NXOS (supports HA)
- Performs management, monitoring, and configuration
- Tight integration with Virtual Machine Manager

Virtual Ethernet Module (VEM)
- Enables advanced networking capability on the hypervisor
- Provides each virtual machine with dedicated "switch port"
- Collection of VEMs: 1 vNetwork Distributed Switch

Virtual Ethernet Ports (vEth Ports)
- VM ports attached to the VEM
- Ports are migrated along with the VM during Live migration
- Port state is preserved across migrations

Nexus 1000V Architecture
Nexus 1000V OpenStack Solution Components

- VSM (HA pair)
- Neutron plugin
- Installer integration
Nexus 1000V Solution for OpenStack

Component Overview: VEM, VSM and VXLAN Gateway, VSG

- VSM based centralized management
- Advanced NX-OS features including VXLAN
- Consistent Physical and Virtual interface management
- Tight integration with OpenStack and Vendor Distributions
Nexus 1000V Forwarding model and Features
Nexus 1000V VEM Architecture

- DataPath Agent: For VSM communication
- DataPath: For packet processing (Forwarding and Features)
  - Multi-threading support
- FastPath: Kernel offload module for caching DataPath decisions
  - Decoupled forwarding and feature lookups
FastPath Module Overview

- Multiple priority queues between fast & slow paths
  - Prioritize control packets over others

- Generic functional modules in the FastPath

- Decoupled L2 and Feature lookups

- Features can be offloaded or accelerated in FastPath

- Efficient handling of short lived flows for flow based features
Comparison to Flow-based Forwarding

- Reduces the number of packets punted to slow-path
  - One per mac address instead of one per flow (L2 forwarding require smac,dmac as the minimum flow spec)
- Any change to the forwarding state doesn’t impact the feature state and vice versa
- Live-migration requires re-learning just VM mac address instead of all the flows to and from the VM
- Flow spec is localized for flow-based features

18 flow entries instead
6 mac forwarding entries
## Nexus 1000V for KVM Functionality

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Switching</strong></td>
<td>L2 Switching, 802.1Q VLAN Tagging, Private VLANs</td>
</tr>
<tr>
<td></td>
<td>LACP, Port-channeling, VPC Host Mode</td>
</tr>
<tr>
<td></td>
<td>Multicast/IGMP Snooping, Jumbo-frame support</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Access Control Lists (L2–4 w/ Redirect), Port ACLs, ACL Statistics, RADIUS,</td>
</tr>
<tr>
<td></td>
<td>TACACS+, VSG (in Preview mode)</td>
</tr>
<tr>
<td><strong>VXLAN</strong></td>
<td>Unicast &amp; multicast modes, Port-statistics, ACLs, Netflow, VXLAN to VLAN Gateway</td>
</tr>
<tr>
<td><strong>Provisioning</strong></td>
<td>Integration with OpenStack Neutron APIs &amp; Horizon dashboard, VM Policy</td>
</tr>
<tr>
<td></td>
<td>Provisioning through port-profiles, Ability to create SLA-based network profiles</td>
</tr>
<tr>
<td><strong>Visibility</strong></td>
<td>Netflow, Port-statistics, VM-level interface statistics, vTracker, SPAN, ERSPAN, vNAM integration</td>
</tr>
<tr>
<td><strong>Manageability</strong></td>
<td>Cisco NX-OS CLI, SNMP (v.1, 2, 3), CDP, Syslog, NTP</td>
</tr>
<tr>
<td></td>
<td>ISSU, SSH v2, Telnet, REST-APIs</td>
</tr>
<tr>
<td></td>
<td>Centralized management through VSM</td>
</tr>
</tbody>
</table>
### Key Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| **VSM based Centralized Management** | • Centralized VM and Uplink interface view  
• Network module on the server status  
• VM interface status and statistics |
| **Overlay Support** | • VXLAN for multicast underlay  
• VXLAN for unicast underlay |
| **OpenStack Extensions** | • Network Resource Extension  
• Port Resource Extension |
The Policy Driven Data Center
Its All About Capturing Intent!

Today’s abstractions expose too many domain specific details

User intent is lost!
Application Language Barriers

Developer and infrastructure teams must translate between disparate languages.
Logical Model Overview

Private-Network and subnets are independent between tenants.
Group-Based Policy: GBP
What’s Wrong with OpenStack Networking Today?

Cloud Application Model

- Service A
- Service B
- Service C

Neutron Model

- L2 / Broadcast is the base API!
- Network / routers / subnets
- Based on existing networking models
- No concept of dependency mapping or intent

• No broadcast / multicast
• Resilient / Fault Tolerant
• Scalable Tiers
• Built around loosely coupled services
• Don’t care about IP addresses
Where Can We Do Better

Separation of Concerns

- Separate application requirements from low level APIs
- Separate tenant from operator

Dependency Mapping

- Build self-documenting dependency maps of tiers of an application
  - Service A consumes service B and Service C

Enable Network Services

- Define network service chains between tiers of an application *without* low level configuration
Introducing Group-Based Policy

Intent-based API for describing application requirements
Captures dependencies between tiers of an application
Plugin model
  Supports mapping to Neutron APIs
  Supports “native” SDN drivers
Application Centric Infrastructure

Rapid Deployment of Applications onto Networks with Scale, Security and Full Visibility

NEXUS 9500 AND 9300

APPLICATION CENTRIC POLICY

CONTROLLER

ACI
OpFlex: An OPEN, Extensible Policy Protocol

OPFLEX WAS DESIGNED TO OFFER:

1. Abstract policies rather than device-specific configuration
2. Flexible, extensible definition of using XML / JSON
3. Support for any device including virtual switches, physical switches, network services with strong interoperability across vendors
4. Open, standardized API with an open source reference implementation

Policies:
- Who can talk to whom
- What about
- Ops requirements
OpenStack GBP Architecture

Neutron Driver maps GBP to existing Neutron API and offers compatibility with any existing Neutron Plugin

Native Drivers exist for OpenDaylight as well as multiple vendors (Cisco included)

Open model that is compatible with ANY physical or virtual networking backends
GBP – Example (Horizon)
Why Cisco ACI and OpenStack

1. **GROUP-BASED POLICY SUPPORT**
   - Automation
   - Intent-driven

2. **PHYSICAL + VIRTUAL**
   - Zero-touch Performance
   - Physical server
   - Multi-hypervisor

3. **FABRIC TUNNELS**
   - Automatic VXLAN
   - Distributed L2
   - Distributed L3

4. **SERVICE CHAINING**
   - Service chaining and redirection

5. **TELEMETRY AND OPERATIONS**
   - Health Metrics
   - Visibility
   - Troubleshooting
Two Options for ACI
APIC Driver for OpenStack

- ML2 (modular level 2) driver supporting existing Neutron APIs: network, router, security group, LBaaS, etc.
- Automation of neutron ports for virtual machines
- Relies on OVS in hypervisor
- Shipping today from Cisco
- Available on Openstack IceHouse, Juno, etc.
Group-Based Policy

- OpenStack extensions on top of Neutron exposing a policy API
- Supports policy API to APIC
- Backwards compatible with existing neutron plug-ins (works with Nexus 9000 standalone)
- Available for Openstack Juno
- Open approach
- Enables Openstack customers to deploy, scale and modify policy across teams fast
OpenStack Partners

Support for major OpenStack Distributions

Testing and Integration
Working closely with vendors to test and qualify APIC Plugin on OpenStack distributions

Easy Deployment
Integrating with existing deployment tools used by each distribution

Customization to ACI
Evaluating ways to expose features that ACI can leverage such as Group Policy and OpFlex
Choice of Deployment Models

**EXISTING 2, 3-TIER DESIGNS**
- Programmability
- Modernized NXOS

**SDN OVERLAY MODEL**
- Integrated Network Virtualization – Routing + Switching
- OpenFlow / Open source
- Third Party Controller

**APPLICATION CENTRIC INFRASTRUCTURE**
- Any Hypervisor
- Physical & Virtual
- Open
Cisco NFV Orchestration Solution

Target Architecture

Prime Service Catalog (PSC)

SP's OSS/BSS or Prime Order Fulfillment

Network Services Orchestrator (NSO)
(Based on Tail-F NCS)

User Self-Service Portal

NFV Orchestrator

VM and Service Lifecycle Manager

VM & Storage Orchestrator

SDN sub-system / SDN Controller

SDN Virtual Forwarder

OSS

A Framework enabled by multiple products & architecture

ESCAPE API

REST API / JCloud (Future)

Other VNF-M e.g. Stratos etc

Inc. Netconf/Yang

Other SD APIC / 3rd Party

Virtual Topology Controller (VTC)

EMS (3rd Party)

MP-BGP

Service Assurance

A Framework enabled by multiple products & architecture

Service Assurance

OpenStack APIs

VNF

VNF

OpenStack

VMWare

Elastic Services Controller (ESC)

ESC API

Netconf/Yang

Other SD APIC / 3rd Party

Restconf /Yang

SDN System/Controller

x86 Server

OVS / Other

VTF
Cisco NFV Orchestration Solution
Use-case: vManaged Services – vMS Service Construct

- **Managed Service**
  - **Orchestration & Management**
    - Evolved Services Platform
    - SP & User Portal
    - Cloud/DC orchestration
    - Service Assurance
    - CPE/WAN Management
  - **VPN Element**
    - CPE/vCPE i.e. Virtual or Physical component
    - Transport technology i.e. MPLS VPN or Overlay based i.e. VXLAN, IPSec, LISP
  - **Service Element**
    - Basic Network service i.e. NAT, FW (as offered in CPE today)
    - Advanced Services i.e. Security, XaaS i.e. HCS, Webex

- **Managed Service**
  - **Orchestration & Management**
    - Evolved Services Platform
    - SP & User Portal
    - Cloud/DC orchestration
    - Service Assurance
    - CPE/WAN Management
  - **VPN Element**
    - CPE/vCPE i.e. Virtual or Physical component
    - Transport technology i.e. MPLS VPN or Overlay based i.e. VXLAN, IPSec, LISP
  - **Service Element**
    - Basic Network service i.e. NAT, FW (as offered in CPE today)
    - Advanced Services i.e. Security, XaaS i.e. HCS, Webex
Adding VNFs in the cloud

Customer Orders VPN Service

ISR CPE Shipped to Customer Site, connected & Powered ON

ISR CPE

PnP Server

PnP Functionality
Zero Touch Provisioning

Provide Day 1 Configuration

Establish VPN: IPSec, IP Overlay (VXLAN, GRE, LISP), L2

CloudVPN Connectivity up

Tenant Portal

SP’s OSS/BSS

REST API

Network Services Orchestrator (NSO)

Elastic Services Controller (ESC)

Virtual Topology Controller (VTC)

OpenStack

OVS/VTF

Internet Gateway

If more VNFs are needed for a Service Chain?

More scalable and flexible service chaining enabled with VTC & high-performance VTF

If more VNFs are needed for a Service Chain?

DCI/PE

XR9600 Server

CSR1Kv

ASAv

DCI/PE

DCI/PE

Internet Gateway

If more VNFs are needed for a Service Chain?
Managing On-prem NFV with ISR 4k

- PnP Functionality Zero Touch Provisioning
- Customer Orders VPN Service
- Tenant Portal
- SP's OSS/BSS
- Network Services Orchestrator (NSO)
- Elastic Services Controller (ESC)
- Virtual Topology Controller (VTC)
- OpenStack
- ISR 4k
- Internet Gateway

PnP Server

- Provide Day 1 Configuration
- Spin up local service Containers as necessary for service VMs on-prem

Establish VPN: IPSec, IP Overlay (VXLAN, GRE, LISP), L2

CloudVPN Connectivity up

If more VNFs are needed for a Service Chain?

More scalable and flexible service chaining enabled with VTC & high-performance VTF

If more VNFs are needed for a Service Chain?

OpenStack

OVS/ VTF

CSR1Kv

ASAv

vWAAS

L7 FW

Network Services Orchestrator (NSO)

DCI/PE

Internet Gateway

Spin up local service Containers as necessary for service VMs on-prem

Establish VPN: IPSec, IP Overlay (VXLAN, GRE, LISP), L2

CloudVPN Connectivity up

If more VNFs are needed for a Service Chain?
Managing On-prem NFV with generic X86 server

Customer Orders VPN Service

Provision VNFs

Spin up local service containers as necessary for service VMs on-prem

Establish VPN: IPSec, IP Overlay (VXLAN, GRE, LISP), L2

CloudVPN Connectivity up

If more VNFs are needed for a Service Chain?

More scalable and flexible service chaining enabled with VTC & high-performance VTF

X86 server

CSR

WAAS

Internet Gateway

DCI/PE

CSR

CSR1Kv

ASAv

OVS/ VTF

OpenStack

Elastic Services Controller (ESC)

Virtual Topology Controller (VTC)

Tenant Portal

PnP Server

Network Services Orchestrator (NSO)

REST API

REST API

SP’s OSS/BSS
Adding VNFs in the cloud to the Meraki service

Customer Orders a Meraki CPE

Meraki CMS

API
Info Exchanged on new CPE Provisioning

Call Home and Zero Touch Provisioning

IPSec Tunnel

Meraki CPE Shipped at Customer Site, connected & Powered ON

CloudVPN Connectivity up

Tenent Portal

REST API

SP’s OSS/BSS

REST API

Meraki CMS

Network Services Orchestrator (NSO)

Elastic Services Controller (ESC)

OpenStack

Virtual Topology Controller (VTC)

OVS/VTF

Internet Gateway

More scalable and flexible service chaining enabled with VTC & high-performance VTF

DCI/PE

CSR1kv

ASAv

Meraki CPE

CloudVPN Connectivity up

More scalable and flexible service chaining enabled with VTC & high-performance VTF
NFV use-case

vCPE for the Home Environment (QvBN - Quantum virtual Broadband Node)
CSR 1000v Market Summary – NFV router

- FCS since March 2013
- Over 250 customers world-wide
- Over 2500 instances sold to date
  - ... and rising

- New feature highlights of IOS XE3.13
  - APPX License Package
  - PfR Master Controller support for Cisco CSR 1000V
  - Platform Hardware Throughput Monitor
  - Shared Management Interface for REST API Support
  - Support for Single Root I/O Virtualization (SR-IOV) on VMware ESXi and Microsoft Hyper-V
  - New or Modified REST API Support
# CSR 1000V Features Per Technology Package

<table>
<thead>
<tr>
<th>Technology Package</th>
<th>IOS-XE Features</th>
<th>Virtualization</th>
</tr>
</thead>
</table>
| **IPBase**<br>(formerly Standard) | ▪ **Basic Networking:** BGP, OSPF, EIGRP, RIP, ISIS, IPv6, GRE, VRF-LITE, NTP, QoS  
▪ **Multicast:** IGMP, PIM  
▪ **High Availability:** HSRP, VRRP, GLBP  
▪ **Addressing:** 802.1Q VLAN, EVC, NAT, DHCP, DNS  
▪ **Basic Security:** ACL, AAA, RADIUS, TACACS+  
▪ **Management:** IOS-XE CLI, SSH, Flexible NetFlow, SNMP, EEM, NETCONF | ESXi 5.5 |
| **SEC**<br>(formerly Advanced) | IPBase Plus...  
▪ **Advanced Security:** Zone Based Firewall, IPSec VPN, EZVPN, DMVPN, FlexVPN, SSLVPN, GETVPN | XenServer 6.1 |
| **AppX** | IPBase Plus...  
▪ **Advanced Networking:** L2TPv3, BFD, MPLS, VRF, VXLAN  
▪ **Application Experience:** WCCPv2, AppXNAV, NBAR2, AVC, IP SLA  
▪ **Hybrid Cloud Connectivity:** LISP, OTV, VPLS, EoMPLS  
▪ **Subscriber Management:** PTA, LNS, ISG | KVM (Ubuntu 12.04 LTS, RHEV 3.1, RHEL 6.3) |
| **AX**<br>(formerly Premium) | **ALL FEATURES** | Hyper-V 2012 R2 |
Virtualized IOS XE
- Generalized to work on any x86 system
- Hardware specifics abstracted through a virtualization layer
- Control Plane and Data Plane mapped to vCPUs
- Bootflash: NVRAM: are mapped into memory from hard disk
- No dedicated crypto engine – we leverage the Intel AES-NI instruction set to provide hardware crypto assist.
- Boot loader functions implemented by GRUB
CSR 1000v IOS XE Threads to vCPU Associations

- IOS XE processing threads in the Guest OS are statically mapped to vCPUs threads
- vCPU threads in turn are allocated to physical cores by the hypervisor scheduler

<table>
<thead>
<tr>
<th>CSR footprint</th>
<th>Control Plane</th>
<th>Data Plane PPE</th>
<th>Data Plane HQF</th>
<th>Data Plane Rx processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>vCPU 0</td>
<td></td>
<td>vCPU 0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>vCPU 0</td>
<td>vCPU 1</td>
<td>vCPU 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>vCPU 0</td>
<td>vCPU 1 &amp; 2</td>
<td>vCPU 3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>vCPU 0</td>
<td>vCPU 1-5</td>
<td>vCPU 6</td>
<td>vCPU 7</td>
</tr>
</tbody>
</table>

**NOTE:** vCPU allocations subject to change without further notice
Example: Single Root IO Virtualization - SR-IOV

- Allows a single PCIe devices to appear to be multiple separate PCIe devices
- Enables network traffic to bypass software switch layers
- Creates physical and virtual functions (PF/VF)
  - PF: full featured PCIe
  - VF: PCIe without configuration resources
  - Each PF/VF gets a PCIe requestor ID s.t. IO memory management can be separated between different VFs
- Requires support in BIOS/Hypervisor
- Supported for IOS XE since 3.13 for ESXi and HyperV.
Virtual Router – VNF Market Requirement

• Routing as a Service: L2 Provider Edge, L3 Provider Edge, Business Internet etc…

• Routers as NFV appliances/VNFs
  • Benefit of elastic expansion
  • flexibility
  • common hardware platform
  • Are Perfect fit of Service Providers Low-End Router requirements
  • Separation of Circuit and Service
  • Virtually Ubiquitous Service Instantiation

• Cisco will deliver a product based on IOS-XRv + High performance XR Dataplane
  • Need features at decent performance
  • Expect to deliver feature parity
IOS-XR vRouter vNPU Data Plane

- Feature Rich Edge Services Focused Packet processor for general-purpose CPU’s
- Shares common code & features: with Cisco NG NPU engine (to be used on ASR 9000, NCS Family)
- Cisco vNPU is based on Cisco’s VPP technology
  - Uses VPP drivers and optimized Tx and Rx routines
  - Constructs super frames of packets and processes them in one shot - exploits temporal locality of application flows. Benefits from I-cache, D-cache hits.
  - Direct PCI pass-through allows send/receive packets with zero operating system overhead
- 64-bit, multi-threaded
- Portable
IOS-XRv 9000 : Positioning
Complementing the current HW portfolio
Summary
ASIC or CPU? Real PE or vPE? SDN or NfV?

**CRS:**
- 2004: 130nm NPU, 40Gbps (~11W/Gbps)
- 2010: 65nm NPU, 140Gbps (~5W/Gbps)
- 2013: 40nm NPU, 400Gbps (~2W/Gbps)
- 2015: 20nm...

**ASR9000:**
- 2009: 90nm NPU, 120Gbps per slot
- 2011: 55nm NPU, 360Gbps per slot
- 2014: 28nm NPU, 800Gbps per slot
- ...

**Virtual:Physical (CPU:ASIC) real estate ratio ~15:1**
NFV Orchestration Solution Elements

“All components are modular”

- Prime Service Catalog (PSC)
- Prime Order Fulfillment (POF)
- Network Services Orchestrator (NSO) [Based on Tail-F NCS]
- Elastic Services Controller (ESC)
- Virtual Topology System (VTS)

Optional products:
- DC Overlay SDN system, consisting of a controller called the Virtual Topology Controller (VTC) and a Forwarder called the Virtual Topology Forwarder (VTF)

Core products:
- OpenStack/VMware

- NFV Orchestrator
- VNF Manager
- VIM

“SW suite for automated NFV service lifecycle management & orchestration”
Prosíme, ohodnoťte tuto přednášku

Děkujeme