Your Time Is Now
Network Function Virtualization (NFV) using IOS-XR

Rajendra Chayapathi
Paresh Shah
BRKSPG-2724
Agenda

• Role of NFV in EPN
• IOS XRv 9000
• Benefits & Use Cases
  • Virtual Route Reflector
  • Virtual Provider Edge
• Deployment & Troubleshooting
• Service Orchestration for NFV
• Summary
NFV decouples the network functions such as NAT, IPS, DNS, RR etc. from proprietary hardware appliances, so they can run in software.

It utilizes standard IT virtualization technologies that run on high-volume service, switch and storage hardware to virtualize network functions.

Customer Demand is Changing

- Rapid Deployment of New Business Applications
- Big Data & Analytics
- On-Demand Bandwidth & Capacity
- Anywhere/Anytime Secure Accessibility
- User Experience, Delivered
- Multi-Vendor Offerings; No Lock-In
- PAYG Models

The New Customer Requirements

- On-Demand Solutions
- Multi-Platform
- Security & Compliance
- Seamless Connectivity
Entering a New Era in the SP Network Evolution

Discontinuity #1:
TDM limits new services, forces architectural shift

Discontinuity #2:
Commoditization of IP services plus high traffic growth limits profitability, forces architectural shift

IP NGN Era
IP unleashes new wave of innovation and service revenues

Evolved Programmable Network (EPN) Era
Network Function Virtualization
Software Defined Networking
Service Orchestration
NFV in Evolved Programmable Network (EPN) Era

Network Function Virtualization

Software Defined Networking

Service Orchestration

Open and Dynamic
Optimal Resource Utilization
Accelerated Innovation
New Services & Revenues
Reduced Cost & Complexity
Elastic & Flexible
Network Functions Virtualization

**Network infrastructure/Service Functions**

**Virtualized compute platforms**

- Key Enabler: Cloud technology
  - Hypervisor & x86 compute hardware
  - Network Programmability APIs
  - Network Automation / Orchestration
Network Functions Virtualization

Where is SDN?

- SDN complementary, but not mandatory
- NFV is not SDN, though they have commonalities
  - Complementary / Orthogonal concepts
## SDN & NFV

### Comparison

<table>
<thead>
<tr>
<th>SDN</th>
<th>NFV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDN Controller:</strong> Open Daylight, Open SDN Controller(OSC) etc.</td>
<td><strong>Virtual network functions:</strong> vFW, vRR, vCPE, vPE etc.</td>
</tr>
<tr>
<td><strong>OpenFlow, NETCONF/Yang, Path computation element protocol (PCEP)</strong></td>
<td><strong>VM to Host (socket, Taps etc.)</strong></td>
</tr>
<tr>
<td><strong>Involves end to end networking</strong></td>
<td><strong>Involves single network entity</strong></td>
</tr>
<tr>
<td><strong>New network architecture</strong></td>
<td><strong>Virtualization of existing architecture</strong></td>
</tr>
</tbody>
</table>
Network Virtualization

- Applications and running using virtualized Hardware end CPUs
- Guest O/S running independently in each VM
- HyperVisor - isolated application providing VMs on the Host
- Basic host operating system
- Virtualization capable CPUs
Network Virtualization
ETSI Architecture Framework for NFV
Virtualizing Network Functions
X86 versus Custom Network Processing Unit (NPU)

**Network Forwarding (L0-3)**
- Mostly predictable traffic
- Stateless functions
- High throughput / BW
- IPv6/v4, MPLS, VPNs, Optical

**Network Services (L4+)**
- Unpredictable traffic
- Stateful functions
- Low to Med Throughput
- BGP Route reflector, Firewall, DPI

**Better fit for NPU**
- Compute
- Bandwidth

**Better fit for x86**
- Compute
- Bandwidth
NFV across Cisco portfolio

Virtualized Network Operating Systems

**IOS-XR**
Virtualized in IOS XRv, IOS XRv 9000

**NX-OS**
Virtualized in Nexus 1000v

**IOS-XE**
Virtualized in CSR1000v

**ASA**
Virtualized in ASAv
Cisco’s VNF Portfolio
## NFV Use-Cases

<table>
<thead>
<tr>
<th>Use-Cases</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Virtual Route Reflector</td>
<td>Virtualized BGP RR delivered on demand</td>
</tr>
<tr>
<td>2  Virtual PE Router</td>
<td>Fully virtualized PE router delivered as an on demand cloud service</td>
</tr>
<tr>
<td>3  Virtual Private Cloud</td>
<td>Single-tier, 2-tier, 3-tier applications with optional NFV service chaining attached to customer L3 VPN</td>
</tr>
<tr>
<td>4  Virtualized Mobility Service</td>
<td>vEPC, vMME, vRAN</td>
</tr>
<tr>
<td>5  Hosted Collaboration Service</td>
<td>Integrating HCS provisioning with VPN configuration for single click customer deployment</td>
</tr>
<tr>
<td>6  Virtualized Video Headend</td>
<td>Cloud DVR, CDN/streaming as a service</td>
</tr>
<tr>
<td>7  Routing-as-a-service</td>
<td>Using CSR to deliver routing/BNG as a cloud service</td>
</tr>
<tr>
<td>8  Virtual BNG in the cloud</td>
<td>High-scale (multi-million subscribers) BNG control plane in the cloud</td>
</tr>
<tr>
<td>9  Virtual Managed Services</td>
<td>Using CSR, ASA to deliver managed services to enterprise customers (attached to customer L3VPN)</td>
</tr>
</tbody>
</table>
IOS XR

- Time tested for years
  - CRS-1, CRS-3, CRS-X, ASR 9000, NCS 6000
- High-scale control plane
- MicroKernel-based
- Modular Software
- Process Restartability & Redundancy
- Remediation through add-on patches
IOS XRv

- IOS XR on x86 Virtualized environment
- Full Platform Independent IOS XR
  - Same IOS XR software feature set
  - Manageability
  - Control Plane
  - Routing

Physical Hardware:
CPU, ASICs, NIC, Consoles, Memory, HDD

Host OS

HyperVisor

IOS XRv

Guest OS (32bit Linux)
IOS XRv - One Physical hardware -- Multiple Instances

- Physical Hardware:
  - CPU, ASICs, NIC, Consoles, Memory, HDD

- Host OS:

- HyperVisors:
  - IOS XRv #1
  - IOS XRv #2
  - Other Guest OS

- Guest OS (32bit Linux)
IOS XRv 9000

- Virtualized IOS XR with Control and Data plane Separation
  - Linux Containers for Admin, Control and Data Planes
  - 64 Bit Kernel

- Scalability through Flexible resource Allocation
  - Data plane scalability.
  - Control Plane scalability
IOS XRv 9000

Forwarding Plane
- Virtual Forwarder
- Software Based H/W assist
- Common code base as - nPower-X ASIC

Routing & Management Plane
- XR Route Processor Functionality
- XR Line Card Functionality
- Support for Physical & Virtual Data-Plane

Admin Plane
- Infra management
- SMU management
- VM/LXC Lifecycle Management
- Upgrade/Downgrade
- Light Weight

MTRIE - Policer - Intf
L3FIB - QoS - L2FIB - ACL

HyperVisor
Host
Cisco IOS XRv 9000

Right sizing Scale and Throughput through Control and Data Planes

Present Mode of Operation

Routers + LCs

IOS XR

RP (Control Plane)

LC (Data Plane)

NxLCs: 1xCPU

Future Mode of Operation

Virtual Routers

Routers/Compute

Compute

Virtual Routers

IOS XRv 9000

LC (Data Plane)

LC (Data Plane)

LC (Data Plane)

N x NPU: MxCPU

Compute Server (Control Plane)

Compute Server (Control Plane)

Compute Server (Control Plane)
Cisco IOS XRv 9000

Design Trade Offs

Possible to degrade overall performance by improving performance for one particular metric

Physical XR Router
IOS XRv 9000

Virtual Router X

Performance:
ACE, TM, & Queues

Performance:
L2/L3 FIB & Labels

Features

Topology

Convergence

Feature Parity

Performance at Low Scale
IOS XRv 9000 Positioning
Completing the XR Edge Portfolio
Benefits & Use Cases
IOS XRv & IOS XRv 9000

Benefits

Lower Opex
• Easy provisioning, configuration & deployment for VMs

Lower Capex
• IOS XRv on standard compute resources
• Multiple XRs on same device

Elastic
• Dynamic resource allocation & de-allocation

Greener
• Low power consumption → Lower carbon footprint

Flexible Growth
• CP & DP Separation and independent resource allocation

SDN Ready
• Independent control and forwarding
Use Cases

- Education and Training
- Network Simulation
- Network Deployment

Universities – Enterprise – Public Sector – Service Providers
Use Cases

Network Simulation & modeling

- Test & Try new control-plane capabilities
- Evaluate network against failures
- Design & plan changes and new features

Equipment Cost
Setup Time
Cumbersome to change

Low Cost
Easy Orchestration
Quick setup & changes

Lab validation
IOS XRv 9000
Use Cases
Cisco PCE controller (XTC)
Use Cases

Network Deployment (vRR & vPE)

- **Consumption based** model - Network growth to match needs
- Redundant devices provisioning without added **cost**
- Service segregation on same hardware
- Grow and **scale** VM’s server resources to match needs
Virtual Route Reflector (vRR)
&
Virtual Provider Edge (vPE)
IOS XRv 9000 as vRR

• Traditional Role of RR
  • BGP peering
  • Solve N*N full-mesh BGP interconnect
  • Distribute BGP routes to PEs

• Not in packet path
• Focal point for iBGP sessions
IOS XRv 9000 as vRR

RR role expanding - centralized provision, services, and applications

Per Service
Per Address Family
Redundant
Optimized Placement
Scalable
Easy Provisioning
IOS XRv 9000 as vRR

Primary
IPv4 RR
Vpnv4 RR
IPv6 RR
L2vpn vRR

Backup

Primary
IPv4 vRR
Vpnv4 vRR
IPv6 vRR
L2vpn vRR

Backup

8 Physical Devices

2 Physical Devices
Virtualized RRs per AFI

Without Compromising

Performance (Multi-Core)
Independent Operation
High Availability
Same BGP Implementation (XR)
IOS XRv 9000 as vPE

Forwarding Performance (Multi-Core)

Consumption Based Growth

Control Plane & High-Performance Data Plane

High Availability

L3VPN
Customer A

L3VPN
Customer B

L3VPN
Customer C
vRR & vPE using IOS XRv 9000

- Performance (Multi-Core)
- Independent Operation
- High Availability
- IOS XR Based Implementation
- Elasticity & Flexibility
- Portability & Agility
- Route Scalability (32/64b OS)
- Management & Orchestration
- Lower Opex/Capex
# vRR & vPE using IOS XRv 9000

<table>
<thead>
<tr>
<th>Physical Router</th>
<th>VRR on UCS Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Power consumption ~425W</td>
<td>Max Power consumption ~410W</td>
</tr>
<tr>
<td>Total power for 8 instance ~3.4kW</td>
<td>Total power for 8 instances ~820W</td>
</tr>
<tr>
<td>Power/Year = 29,785 KWh</td>
<td>Power/Year = 7,182 KWh</td>
</tr>
<tr>
<td>Power Cost/Year = $5,659 (19c/kWh)</td>
<td>Power Cost/Year = $1,364 (19c/KWh)</td>
</tr>
</tbody>
</table>

Power Calculations based on
ASR9001 (Max Power)
UCS C240 M3 SFF with Intel E5-2643 v2 3.30 GHz/130W 6C/25MB Cache/DDR3 1866MHz with 96 GB Mem, 4 HDD with RAID, and 1 Adapters.
Physical Network Device vs NFV
Consumption Based Deployment

Physical Network Device
- Under-Provisioned
- Over-Provisioned

Network Function Virtualization
- Consumption based capacity growth

CPU, Memory, Gbps vs Time
- Capacity Demand
- Capacity Deployed

Flexible Growth

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
IOS XRv 9000 Deployment
Deployment Models

**XRv 9000 Standalone**
- Any server
- Any Hypervisor
- Flexible

**XRv 9000 Appliance**
- Baremetal
- Plug and play
- Convenient

**XRv 9000 with CSP2100**
- P&P or User initiated Cisco Orchestrator
- Cloud Ready

New! Onboarded on Cisco NFVi
IOS XRv 9000 Hardware/Software Requirements

• Hardware
  • Any x86-based server capable of virtualization
    • e.g. Intel® CPUs with VT-x support

• Hypervisor
  • hypervisor agnostic
  • VMWare ESXi 5.5/6.0, QEMU/KVM 1.0

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU (Cores)</td>
<td>2 (1 Control Plane, 1 Data Plane)</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>8GB</td>
</tr>
<tr>
<td>Hard Disk</td>
<td>45GB</td>
</tr>
<tr>
<td>Serial Port</td>
<td>1 (for console)</td>
</tr>
<tr>
<td>NIC Port (E1000/VirtIO/Niantic 10G)</td>
<td>4 (2 reserved, 1 traffic, 1 mgmt)</td>
</tr>
<tr>
<td></td>
<td>19 (2 reserved, 16 traffic, 1 mgmt)</td>
</tr>
</tbody>
</table>
IOS XRv 9000 Features

- MP-BGP/eBGP, BGP 3107, FlowSpec
- OSPF/ISIS etc.
- BFD (Bidirectional Forwarding Detection)
- Application Hosting
- LDP/MPLS, 6PE, 6vPE, RFC 3107 (3 labels), L3VPN
- IPv4 ACL (chained), uRPFv4/v6, LPTS
- Netconf/YANG & SNMP
- Hierarchical QoS policing, WRED
- EFD (Early Fast Discard)
- IOS XR Manageability & Control Plane
  - PIE/SMU Upgrades
  - LPTS/CoPP
- Gratuitous ARP
- VRRP
- IPv4 & IPv6 uRPF
- Lawful Intercept
- IPSLA
- Telemetry
- Link Aggregation Group (LAG)
- Network Service Header (NSH)

IOS XRv 9000 Operational Enhancements

Visibility & Telemetry
- Operational Data, Deep analytical hooks
- Policy-based, flexible, Push Model

Programmability
- Data accessible via published model driven interfaces
- Machine friendly
- Enables automation @ scale

Application Hosting
- Ability to run 3rd party off the shelf applications built with Linux tool chains
- Run custom applications inside an LXC container on the 64-bit Linux host

Flexible Platform and Packaging
- RPM Packages: EIGRP, MGBL, MPLS, K9SEC, LI, BGP etc.
- Automated package dependency checkers
- Automated Provisioning at Bootup
IOS XRv 9000 Telemetry & Programmability

- Traffic analysis, fault prediction, “gray” failure
- Flexible, efficient, extensible data & transport
- Real-time inspection
- Big data aggregation and analysis
- Automated remediation and policy enforcement

Deeper instrumentation + smarter analytics tools = Real-time action-ability
Active feedback & Auto-remediation
**IOS XRv 9000 Performance**

- **Single Core**
  - 2015: 8 Gbps
  - 2016: 40 Gbps
  - 2017+: 160 Gbps

- **Multi-Core**
  - 2016: 80 Gbps
  - 2017+: ? Gbps

- **Multi-Socket**
  - 2016: 120 bps
  - 2017+: IMIX traffic packet size with features enabled

- Features:
  - IMIX traffic packet size
  - Vanilla IPv4

**Notes:**
- IMIX traffic packet size with features enabled in 2017+.
IOS XRv 9000 Deployment
Pass-through vs Device Emulation

**Physical NIC**

**XRv9000 VM**

**Physical device+ driver**

**Emulated device**

**Guest Driver virtIO / E1000**

**XRv9000 VM**

**Emulated**

**Physical NIC**

**High Performance**
IOS XRv 9000 Deployment

- Physical Interfaces
  - Eth0
  - Eth1
  - Eth2

- Virtual Bridges
  - Mgmt Bridge
  - Data Bridge-1

- Virtual Interface
  - Tap0

- Hypervisor Interface
  - Mgmt

- Virtual Machine

Steps:
1. Copy XRv 9000 image (.ova/.iso/.vmdk) to server
2. Create Disk running image
3. Create Virtual (Tap) interfaces
4. Start simulation
Creating TAP and Bridge

cisco@epn-sjcj-ucs1:~$ sudo tunctl -t Tap1
Set 'Tap1' persistent and owned by uid 0
cisco@epn-sjcj-ucs1:~$ sudo ifconfig Tap1 up

cisco@epn-sjcj-ucs1:~$ sudo tunctl -t Tap2
Set 'Tap2' persistent and owned by uid 0
cisco@epn-sjcj-ucs1:~$ sudo ifconfig Tap2 up
<create Tap3/Tap4>

cisco@epn-sjcj-ucs1:~$ sudo brctl addbr vbridge1

cisco@epn-sjcj-ucs1:~$ sudo brctl addbr vbridge2

cisco@epn-sjcj-ucs1:~$ sudo brctl addif vbridge1 Tap1 eth4

cisco@epn-sjcj-ucs1:~$ sudo brctl addif vbridge2 Tap2 eth5

cisco@ubuntu-EPN-4:~$ sudo brctl show vbridge1
bridge name  bridge id     STP enabled interfaces
vbridge1     8000.b6c7102ae0f6 no Tap1
             8000.b6c7102ae0f6 no eth4
Launching the IOS XRv 9000 Virtual Machine

cisco@epn-sjcj-ucs1:~$ sudo /usr/bin/qemu-system-x86_64 -m 16384 -smp cores=4,sockets=1 -name XRv-Test-Launch -drive file=./xrv9k.raw,media=disk,index=1 -drive file=./xrv9k-fullk9-x.iso-6.0.0,media=cdrom,index=2 -serial telnet:0.0.0.0:12345,server,nowait -device e1000,netdev=mgmt-intf -netdev tap,ifname=Tap1,script=no,downscript=no,id=mgmt-intf -device e1000,netdev=data-intf -netdev tap,ifname=Tap4,script=no,downscript=no,id=data-intf -display none -enable-kvm -boot once=d

16G Memory
4 CPU Cores
XRv9K Instance
XRv9K Disk
XRv Image File
Console port
Ethernet (Mgmt)
Ethernet (GigE)
Accessing the IOS XRv Virtual Machine

cisco@epn-sjcj-ucsl:~$ telnet 0.0.0.0 12345
[Linux-initrd @ 0x456bc000, 0x3a93367c bytes]
Starting udev
Populating dev cache
Configuring network interfaces... done.

Mon Feb 8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Hardware profile: vpe
Mon Feb 8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Host has 16Gb RAM / 4 vCPUs
Mon Feb 8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Management plane: 1Gb RAM / 0 vCPUs
Mon Feb 8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): XR control plane: 7Gb RAM / 2 vCPUs
Mon Feb 8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Data plane core assignment: 2-3
Mon Feb 8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Control plane core assignment: 0-1
IOS XRv 9000 Bring-up

Accessing the IOS XRv Virtual Machine

Mon Feb 8 23:49:45 UTC 2016: Install finished on sda
Rebooting XRv9k system after installation ...
[ 99.990922] reboot: Restarting system

Welcome to the Cisco IOS XRv9k platform
Please wait for Cisco IOS XR to start.
Copyright (c) 2014-2015 by Cisco Systems, Inc.

Cisco IOS XR console will start on the 1st serial port
Cisco IOS XR aux console will start on the 2nd serial port
Cisco Calvados console will start on the 3rd serial port
Cisco Calvados aux will start on the 4th serial port

!!!!!!!!!!!!!!!!!!!!!! NO root-system username is configured. Need to configure root-system username.
!!!!!!!!!!!!!!!!!!!!!!

--- Administrative User Dialog ---
Enter root-system username:
IOS XRv 9000 Deployment

Show Commands

RP/0/RP0/CPU0:ios#show ver
Tue Feb 9 00:10:36.484 UTC
Cisco IOS XR Software, Version 6.0.0
Copyright (c) 2013-2015 by Cisco Systems, Inc.

Build Information:
Built By : alnguyen
Built On : Thu Dec 24 00:54:24 PST 2015
Build Host : iox-lnx-009
Workspace : /auto/srcarchive16/production/6.0.0/xrv9k/workspace
Version : 6.0.0
Location : /opt/cisco/XR/packages/
cisco IOS-XRv 9000 () processor
System uptime is 16 minutes

RP/0/RP0/CPU0:ios#
Show Commands

RP/0/RP0/CPU0:ios# show platform
Tue Feb 9 00:09:33.310 UTC
Node name   Node type   Node state   Admin state   Config state
-----------------------------------------------------------------------------------
0/RP0         R-IOSXRV9000-RP   OPERATIONAL   UP           NSHUT
RP/0/RP0/CPU0:ios#

RP/0/RP0/CPU0:ios# show ipv4 int br
Tue Feb 9 00:12:04.600 UTC
Interface                      IP Address      Status          Protocol  Vrf-Name
GigabitEthernet0/0/0/0         unassigned     Shutdown        Down     default
MgmtEth0/RP0/CPU0/0            unassigned     Shutdown        Down     default
RP/0/RP0/CPU0:ios#

Tap1=MgmtEth0
Tap2=Future
Tap3=Future
Tap4=Giga0/0/0/0

Single RP. No LineCard
IOS XRv 9000 Deployment
Containers and 3rd Party Network NameSpace

```bash
[sysadmin-vm:0_RP0:~]$ ssh 10.0.2.16
Last login: Tue Feb  9 01:21:24 2016 from 10.11.12.15
[host:~]$ virsh list
Id    Name                           State
----------------------------------------------------
5299  sysadmin                       running
12065 default-sdr__uvf--2            running
15153 default-sdr--1                running
```

```bash
[host:~]$ virsh list
Id    Name                           State
----------------------------------------------------
5299  sysadmin                       running
12065 default-sdr__uvf--2            running
15153 default-sdr--1                running
```

```bash
RP/0/RP0/CPU0:ios# show ipv4 int br
Thu Feb 11 15:55:05.581 UTC
Interface                      IP-Address      Status          Protocol Vrf
Name                             Up              Up       default
Loopback0                        1.2.3.4         Up              Up       default
Loopback2                        110.2.2.2       Up              Up       default
Loopback3                        110.3.3.3       Up              Up       default
GigabitEthernet0/0/0/0           200.1.1.1       Up              Up       default
MgmtEth0/RP0/CPU0/0              unassigned      Shutdown        Down     default
RP/0/RP0/CPU0:ios#
```

```bash
[sysadmin-vm:0_RP0:~]$ ssh 10.0.2.16
Last login: Tue Feb  9 01:21:24 2016 from 10.11.12.15
[host:~]$ virsh list
Id    Name                           State
----------------------------------------------------
5299  sysadmin                       running
12065 default-sdr__uvf--2            running
15153 default-sdr--1                running
```

```bash
[host:~]$ virsh list
Id    Name                           State
----------------------------------------------------
5299  sysadmin                       running
12065 default-sdr__uvf--2            running
15153 default-sdr--1                running
```

```bash
RP/0/RP0/CPU0:ios# show ipv4 int br
Thu Feb 11 15:55:05.581 UTC
Interface                      IP-Address      Status          Protocol Vrf
Name                             Up              Up       default
Loopback0                        1.2.3.4         Up              Up       default
Loopback2                        110.2.2.2       Up              Up       default
Loopback3                        110.3.3.3       Up              Up       default
GigabitEthernet0/0/0/0           200.1.1.1       Up              Up       default
MgmtEth0/RP0/CPU0/0              unassigned      Shutdown        Down     default
RP/0/RP0/CPU0:ios#
```

```bash
[xr-vm_node0_RP0_CPU0:~]$ ip netns exec tppns ifconfig | more
Gi0_0_0_0 Link encap:Ethernet   HWaddr 00:50:56:b9:44:0c
inet addr:200.1.1.1  Mask:255.255.255.0
lo:0 Link encap:Local Loopback
inet addr:1.2.3.4  Mask:255.255.255.255
lo:2 Link encap:Local Loopback
inet addr:110.2.2.2  Mask:255.255.255.255
lo:3 Link encap:Local Loopback
inet addr:110.3.3.3  Mask:255.255.255.255
```
Checking License Status

RP/0/# show license platform summary
Sat Dec 26 05:47:08.537 UTC
Current state: PRODUCTION

|-------------|-------------------------------|---------------------------------|

<table>
<thead>
<tr>
<th>Feature/Area</th>
<th>Entitlement</th>
<th>Last</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td><strong>Product:</strong> Right to Use</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>System</td>
<td><strong>Feature:</strong> BGP Scale up to 4M</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
IOS XRv 9000 Deployment using ESXi

- ISO image Upload
- Datastore ISO File
  - [datastore2] ISO Folder/xrv9k-full-x

- Guest Operating System:
  - Linux as Guest OS

- Number of virtual sockets: 1
- Number of cores per virtual socket: 4
- Total number of cores: 4
- Memory Size: 16 GB
- Allocated minimum 4 CPU
- Allocated minimum 8GB Mem

- Network
  - NIC 1: VM Network
  - NIC 2: XRV9k R1 reserved ifaces
  - NIC 3: XRV9k R1 reserved ifaces
  - NIC 4: VM Network
- Minimum 4 Network interfaces
Create Disk: Thin provisioning, 55GB, IDE

Disk Size: 55 GB

Disk Provisioning:
- Thick Provision Lazy Zeroed
- Thick Provision Eager Zeroed
- Thin Provision

Virtual Device Node:
- SCSI (0:0)
- IDE (0:0)

Network Backing:
- Server (VM listens for connection)
- Client (VM initiates connection)

Port URI: telnet://172.18.231.103:2001

Creating Serial Interface
IOS XRv 9000 Deployment on ESXi

Filesystem type is iso9660, using whole disk
kernel /boot/bzImage root=/dev/ram console=ttyS0 prod=1 install=/dev/sda platfo

<SNIP>
Wed Feb 17 02:13:47 UTC 2016: Copying all ISOs to repository took 68 seconds
[ 340.853307] reboot: Restarting system
Press any key to continue.

<SNIP>

Telnet to the Serial Port
```
telnet <esxi_host_ip> <port_number>
```

Will go through baking process on first boot up & reload
Only happens once, during the first bootup

Create Username and Password

--- Administrative User Dialog ---
Enter root-system username:
Using ESXi Hypervisor

- Creating XRv 9000 Virtual Machine using vSphere GUI

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Custom</td>
</tr>
<tr>
<td>Name and Location</td>
<td>as with any other VM</td>
</tr>
<tr>
<td>Storage</td>
<td>as with any other VM</td>
</tr>
<tr>
<td>Virtual Machine Version</td>
<td>&quot;Virtual Machine Version 8 or 9&quot; *</td>
</tr>
<tr>
<td>Guest Operating System</td>
<td>&quot;Other&quot;, version &quot;Other (64-bit)&quot;</td>
</tr>
<tr>
<td>CPUs</td>
<td>Max 14 cores</td>
</tr>
<tr>
<td>Memory</td>
<td>Min 3 GB, Max 8 GB</td>
</tr>
<tr>
<td>Network</td>
<td>4-11 NICs, First NIC will be MgmtEthernet0/0/CPU0/0 while NIC 3-11 will be GigabitEthernet/TenGigabitEthernet</td>
</tr>
</tbody>
</table>
Using ESXi Hypervisor

- Creating XRv 9000 Virtual Machine using vSphere GUI (Con’t)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCI Controller</td>
<td>LSI Logic Parallel (default)</td>
</tr>
<tr>
<td>Select a disk</td>
<td>&quot;Use an existing virtual disk&quot;</td>
</tr>
<tr>
<td>Select Existing Disk</td>
<td>select XRv 9000 VMDK image</td>
</tr>
<tr>
<td>Advanced Options</td>
<td>Must be an IDE disk</td>
</tr>
<tr>
<td>Ready to Complete</td>
<td>select &quot;Edit the virtual machine settings before completion&quot;</td>
</tr>
</tbody>
</table>

"Virtual Machine Properties" window – add 2 serial ports as:

- Under "Hardware", click "Add…"
- Select "Serial Port"
- Select "Connect via Network"
- Select "Server" and enter a telnet URI with an unused port (e.g., telnet://<host IP address>:5001) - each VM and each serial port must use a unique port number.
- Repeat this to add a second serial port. The first serial port will be the console port, and the second will be the aux port.

- Start the VM. Telnet to the configured serial port(s) to interact with and configure the VM
IOS XRv 9000 Deployment using Openstack
IOS XRv 9000 Deployment using Openstack
IOS XRv Troubleshooting
IOS XRv 9000 Troubleshooting

Hierarchy

- Host Hardware
- Host OS
- HyperVisor
- XR Level
- VNF Level
- Hypervisor Level
- Host OS Level
- Hardware Level
**IOS XRv 9000 Troubleshooting**

**Hierarchy**

- **XR Level**
- **VNF Level**
- **Hypervisor Level**
- **Host OS Level**
- **Hardware Level**

**Regular XR Troubleshooting Techniques**

**DPA / DPC Communication and Packet Stats**

```
RP/0/RP0/CPU0:SS_Node1#show controllers dpa statistics global

<table>
<thead>
<tr>
<th>Index</th>
<th>Punt</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1575</td>
<td>ARP</td>
<td>10</td>
</tr>
<tr>
<td>1677</td>
<td>IFIB</td>
<td>104034</td>
</tr>
<tr>
<td>1698</td>
<td>IPv4 BFD</td>
<td>1404379</td>
</tr>
<tr>
<td>1722</td>
<td>IPv4 incomplete TX adjacency</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Inject</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>268</td>
<td>IPv4 from fabric multicast</td>
<td>103716</td>
</tr>
<tr>
<td>270</td>
<td>IPv4 from fabric next-hop</td>
<td>330</td>
</tr>
<tr>
<td>275</td>
<td>Inject to fabric</td>
<td>104047</td>
</tr>
<tr>
<td>276</td>
<td>Inject to port</td>
<td>1510764</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Drop</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>IPv4 disabled in uIDB</td>
<td>3888</td>
</tr>
<tr>
<td>113</td>
<td>IPv6 disabled in uIDB</td>
<td>60</td>
</tr>
<tr>
<td>236</td>
<td>Preroute PIT lookup missed</td>
<td>1</td>
</tr>
</tbody>
</table>
```
IOS XRv 9000 Troubleshooting

Hierarchy

- **XR Level**
  - Intel Virtualization turned on in BIOS
  - HT should be turned off (for dataplane performance)
  - Power optimization turn off

- **VNF Level**

- **Hypervisor Level**
  - Dedicated pinned CPU \((vCPU = pCPU)\)
  - Non-Uniform-Memory-Access (NUMA)
    - use local memory same node as dedicated CPU, not foreign memory

- **Host OS Level**

- **Hardware Level**
  - Turn Off HyperTreading (HT)
  - Turn Off Power Optimization
server!~$ sudo netstat -pln | grep 12346
tcp 0 0 0.0.0.0:12346 0.0.0.0:* LISTEN 18135/qemu-system-
x

server!~$ top
server!~$ numactl -hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 16 17 18 19 20 21 22 23
node 0 size: 257762 MB
node 0 free: 194589 MB
node 1 cpus: 8 9 10 11 12 13 14 15 24 25 26 27 28 29 30 31
node 1 size: 258045 MB
node 1 free: 247971 MB
**CPU 1 & 8 allocated (NUMA 0 and 1)**

Memory from NUMA Node 0

```bash
cisco@ubuntu-EPN-4:~$ sudo numastat -p 18135
```

### Per-node process memory usage (in MBs) for PID 18135 (qemu-system-x86)

<table>
<thead>
<tr>
<th>Node 0</th>
<th>Node 1</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huge</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Heap</td>
<td>0.71</td>
<td>1.37</td>
</tr>
<tr>
<td>Stack</td>
<td>0.07</td>
<td>0.03</td>
</tr>
<tr>
<td>Private</td>
<td>9735.30</td>
<td>372.94</td>
</tr>
<tr>
<td>Total</td>
<td>9736.07</td>
<td>374.34</td>
</tr>
</tbody>
</table>
server!~$ sudo taskset -pc 1 18135
pid 18135's current affinity list: 0-31
pid 18135's new affinity list: 1

server!~$ sudo taskset -pc 2 18135
pid 18135's current affinity list: 1
pid 18135's new affinity list: 2
IOS XRv 9000 Troubleshooting

Hypervisor Common mistakes

• Multiple XRv 9000 using same disk image
  – Each instance needs a separate disk

• Multiple XRv using same console

$ qemu-system-x86_64 -serial telnet::2345,server,nowait <...snip...>

inet_listen_opts: bind(ipv4,0.0.0.0,2345): Address already in use
inet_listen_opts: FAILED
chardev: opening backend "socket" failed: Address already in use
qemu: could not open serial device 'telnet:0.0.0.0:2345,server,nowait': Address already in use
### IOS XRv 9000 Troubleshooting

#### Host Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>top</strong></td>
<td>Displays system and process status.</td>
</tr>
</tbody>
</table>

```
top - 09:26:59 up 13 days, 58 min,  2 users,  load average:  0.49, 0.47, 0.50
Tasks:  325 total,  2 running, 322 sleeping,  0 stopped,  1 zombie
Cpus:   1.6%us,  0.4%sy,  0.1%ni,  98.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    181507016k total,  8715152k used,  172791864k free,  146552k buffers
Swap:   134143996k total,         0k used, 134143996k free,  4476556k cached

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
  5477 cisco     20   0 50460  26m 3808 R   39  0.0   4843:57 Xtightvnc
  31187 libvirt- 20   0 5972m 1.9g 7640 S    9  1.1 377:28.36 qemu-system-x86
  34605 libvirt- 20   0 5972m 1.8g 7628 S    7  1.0 348:28.40 qemu-system-x86
  24982 cisco     30  10 32068 1844 1488 S   4  0.0   0:05.99 fuzzyflakes
```
IOS XRv 9000 Troubleshooting

Host OS Common Issues

• Missing qemu/KVM package
  – apt-get install qemu-kvm;

• Not able to create TAP interface:
  – apt-get install uml-utilities (required for tap interface for bridge to physical)

• Virtualization not enabled:
  – Ensure VT-x flag (or equivalent) is exposed to operating system
    • egrep -c '(vmx|svm)' /proc/cpuinfo

• Ensure that user is added to ‘kvm’ group
  – sudo addgroup `id -un` kvm

```bash
server!~$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
cisco@ubuntu-EPN-4:~$
```
Service Orchestration for NFV
Service Orchestration for NFV

Network Function Virtualization  Software Defined Networking  Service Orchestration

Service Orchestration

Cisco Network Service Orchestrator (NSO)

Server
IOS-XRv 9000
IOS-XRv 9000
IOS-XRv 9000

Hypervisor

Cloud VM Orchestration
Cisco ESC  vmware  openstack

Cisco live!
NFV with IOS XR

Right Sizing Your Deployment

Choose your Service
- Core / Transport
- Peering
- DCI
- PE
- Subscriber Services
- Virtual PE (vPE)
- Virtual RR (vRR)

Size up your Data Plane
- Core / Transport
- Peering
- DCI
- PE
- Subscriber Services
- Virtual PE (vPE)
- Virtual RR (vRR)

Adapt your Control Plane
- Core / Transport
- Peering
- DCI
- PE
- Subscriber Services
- Virtual PE (vPE)
- Virtual RR (vRR)

Choose between On-box, Hybrid or Virtual CP Based on Use Case

Today's IOS-XR on box Control Plane

Virtualized CP or Expansion CP from Physical System

Multichassis NCS 6000 Multichassis CRS-X
NCS 6000 NCS 5500 ASR 9000 Tomahawk CRS-X
CRS ASR 9000 Typhoon
IOS-XRv 9K
NFV with IOS XR

Completing Portfolio

- CRS Portfolio
- Edge Routing ASR 9000 Portfolio
- NCS 5500 NCS 5000
- NCS 6000 Single & Multi Chassis
- NFV Virtual Router IOS-XRv 9000
Putting it all together…

- **Compute**
- **Network**
- **Storage**
- **Hypervisors**
- **Virtual Functions**

**Key Features**
- SMU-ability
- Opex Saving
- Low Capex
- Carrier Class
- Scalable
- Multi-threaded
- Flexible
- Elastic
- High Availability

**IOS XRv 9000**
- Virtual Router
- Real Performance
Re-Cap

• Role of NFV in EPN
• IOS XRv 9000
• Benefits & Use Cases
  • Virtual Route Reflector
  • Virtual Provider Edge
• Deployment & Troubleshooting
• Service Orchestration for NFV
• Summary
By the same authors…

(Dec 2016)
ISBN: 978-0134463056
Complete Your Online Session Evaluation

• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Lunch & Learn
- Meet the Engineer 1:1 meetings
- Related sessions
Thank You
Your Time Is Now