Layer 3 Unicast Routing on NX-OS

In This Section

- NX-OS Layer 3 Support
- NX-OS Routing Protocols Overview
- Routing in NX-OS vs. IOS

NX-OS Layer 3 Switching Support

- NX-OS layer 3 features are hardware dependent
  - M cards and F2e/F3 cards on N7K
  - Nexus 5500 with Layer 3 Module
  - Native support in Nexus 3000, 5600, 6000, & 9000
- NX-OS layer 3 features are license dependent
  - Licensing Cisco NX-OS Software Features

NX-OS Layer 3 Interfaces

- Like Catalyst IOS, NX-OS supports…
  - Native layer 3 routed interfaces
    - I.e. **`no switchport`**
  - Switched Virtual Interfaces (SVIs)
    - i.e. VLAN interfaces
    - Must be enabled with **`feature interface-vlan`**

NX-OS Virtual Routing & Forwarding Instances (VRFs)

- Like IOS, NX-OS Virtual Routing & Forwarding Instances are used to create separate logical routing tables
  - Layer 3 interfaces in different VRFs cannot exchange traffic by default
- NX-OS VRFs behave slightly different than IOS, as…
  - All layer 3 interfaces are automatically in VRF table "default"
  - MGMT0 is automatically in vrf "management"
  - VRFs are defined as **`vrf context`**
  - Static routes are defined under the vrf context
  - Dynamic routing is VRF aware, but configured under the same process
  - Exec mode **`routing-context vrf`** can change the default VRF for verifications

NX-OS Routing Protocol Support

- NX-OS supports all dynamic routing protocols
  - RIPv2, EIGRP, OSPF, IS-IS, BGP
- NX-OS also supports static routes & Policy Based Routing (PBR)
- NX-OS protocols are…
  - IPv4 & IPv6 aware
  - VRF aware
  - NSF aware
  - Supported with BFD for fast convergence

NX-OS Dynamic Routing

- Like IOS, NX-OS has both global and interface routing configuration
  - Global routing process affects all links or the routing instance as a whole
    - E.g. EIGRP AS, OSPF SPF Timers, IS-IS Level, etc.
  - Interface commands affect routing protocol behavior of that link
    - E.g. Passive interface, EIGRP route filters, OSPF hello timers, etc.
- Unlike IOS there is no **`network`** command for IGP
  - Protocols are enabled directly at the link level
  - **`network`** command used for BGP NLRI origination

NX-OS Policy Based Routing (PBR)

- Like IOS, NX-OS can use PBR to...
  - Classify inbound traffic by ACL, packet length, etc.
  - Choose next-hop, output interface, etc. outside the normal RIB/FIB
- Unlike IOS, NX-OS PBR...
  - Must be enabled with **`feature pbr`**
  - Needs **`route-map [`*`name`*`] pbr-statistics`** to enable policy statistics

NX-OS Route Redistribution

- Unlike IOS, route-maps are ***required*** to perform redistribution on NX-OS
  - Same route-map match/set logic as IOS
- Redistribution does not include directly connected interfaces
  - Requires **`redistribute direct route-map`**...

Recommended Resources

- Cisco Nexus 7000 NX-OS/IOS Comparison Tech Notes
  - Cisco NX-OS/IOS Layer-3 Virtualization Comparison
  - Cisco NX-OS/IOS EIGRP Comparison
  - Cisco NX-OS/IOS OSPF Comparison
  - Cisco NX-OS/IOS BGP (Basic) Comparison
  - Cisco NX-OS/IOS BGP (Advanced) Comparison

In This Section

- EIGRP Routing on NX-OS

EIGRP on NX-OS vs. IOS Caveats

- Cisco NX-OS/IOS EIGRP Comparison
- Feature is disabled by default
  - **feature eigrp**
- Routing process ID is a string, not an AS number
  - **autonomous-system** must be defined under process
- Protocol enabled at the interface level instead of process level
  - No **network** command like in IOS
- Loopback0 is always used as Router-ID
  - Even if there is another loopback with a higher IP address
- No support for unicast neighbors
  - No **neighbor** command under process
- No support for unequal cost load balancing
  - No **variance** command under process

Configuring EIGRP on NX-OS

- Enable global process
  - **router eigrp [name]**
- Define Autonomous System
  - **autonomous-system [num]**
- Enable EIGRP at link level
  - **ip router eigrp [name]**

Verifying EIGRP on NX-OS

- Common verifications
  - **show run eigrp**
  - **show ip eigrp**
  - **show ip eigrp interfaces brief**
  - **show ip eigrp neighbors**
  - **show ip eigrp topology [all-links]**
  - **show ip eigrp topology w.x.y.z/len**
  - **show ip route eigrp**
  - **debug ip eigrp packet hello**
  - **debug ip eigrp neighbor w.x.y.z**

EIGRP Features – Passive Interface

- Passive interface stops the sending of EIGRP hellos
  - Result is no adjacency forms on the link
  - Does not stop link from being advertised into EIGRP topology
- Typical use case is for SVIs at L2/L3 boundary
  - Routing protocol peerings should happen North and East/West, not South
- Configured at the link level
  - **ip passive-interface eigrp [name]**

EIGRP Features – Authentication

- Authenticates EIGRP hello packets before adjacency occurs
  - Neighbors don't form unless authentication is correct
- Syntax is identical to IOS
  - Define key chain globally
    - **key chain MYKEYCHAIN**
      - **key 1**
        - **key-string MYPASSWORD**
  - Apply key chain at interface level
    - **ip authentication mode eigrp A md5**
    - **ip authentication key-chain eigrp A MYKEYCHAIN**

EIGRP Features – Distribute Lists

- Distribute lists are used to filter routing updates
  - Can be applied inbound and outbound at link level
  - References a prefix-list that matches prefix/len
- EIGRP has no hierarchy, unlike OSPF/IS-IS
  - Implies distribute-list filtering can be applied anywhere
- Implementing Distribute Lists
  - Define prefix-list
    - **ip prefix-list FILTER permit 10.0.0.0/24**
  - Apply at interface level
    - **ip distribute-list eigrp A prefix-list FILTER out**

EIGRP Features – Default Routing

- Default route is a fallback for all prefixes without a longer match
  - I.e. prefix doesn't match any route of /1 - /32
- Default route advertisement is configured under process level
  - **default-information originate**

EIGRP Features – Summarization

- Summarization used to combine prefixes into shorter match advertisement
  - E.g. two /24's combine to a single /23
- Secondary result is limiting the QUERY domain
  - Helps to improve EIGRP convergence and scaling
- EIGRP has no hierarchy, unlike OSPF/IS-IS
  - Implies summarization can be applied anywhere
- Summarization is applied at interface level
  - **ip summary-address eigrp A 10.0.0.0/23**

EIGRP Features – Stub Routing

- Like summarization, used to limit the QUERY domain
  - Don't send QUERY to neighbor with stub flag set
- Configured under the process level
  - **eigrp stub [direct | summary ...]**

Configuring IPv6 EIGRP on NX-OS

- Enable IPv6 under EIGRP process
  - **address-family ipv6 unicast**
- Define AS number
  - **autonomous-system [num]**
- Enable IPv6 EIGRP at link level
  - **ipv6 router eigrp A**

Verifying IPv6 EIGRP on NX-OS

- Common verifications
  - **show run eigrp**
  - **show ipv6 eigrp**
  - **show ipv6 eigrp interfaces brief**
  - **show ipv6 eigrp neighbors**
  - **show ipv6 eigrp topology [all-links]**
  - **show ipv6 eigrp topology A:B::C:D/len**
  - **show ipv6 route eigrp**
  - **debug ipv6 eigrp packet hello**
  - **debug ipv6 eigrp neighbor A:B::C:D**

In This Section

- OSPF Routing on NX-OS

OSPF on NX-OS vs. IOS Caveats

- Cisco NX-OS/IOS OSPF Comparison
- Feature is disabled by default
  - **feature ospf**
- Protocol enabled at the interface level instead of process level
  - No **network** command like in IOS
- Protocol process must be globally enabled
  - Interface level command does not generate global process automatically
- Loopback0 is always used as Router-ID
  - Even if there is another loopback with a higher IP address
- Reference bandwidth is 40Gbps
  - Can cause suboptimal path selection when combined with older platforms

OSPF on NX-OS vs. IOS Caveats (cont.)

- Areas appear as dotted-decimal in configuration
  - Area 0 and 0.0.0.0 are the same
- Adjacency logging is not on by default
  - Needs **log-adjacency-changes** under process
- No support for distribute-lists
- Secondary addresses are advertised by default
  - Can be disabled with **ip router ospf <instance> area <#> secondaries none**
- Loopbacks are always advertised as /32
  - Can be disabled with **ip ospf advertise-subnet**

Configuring OSPF on NX-OS

- Enable global OSPF process
  - **feature ospf**
  - **router ospf [process-id]**
- Enable interface OSPF process
  - Interface level **ip ospf [process-id] area [area-id]**

Verifying OSPF on NX-OS

- Common Verificaitons
  - Verify OSPF is enabled
    - **show ip ospf**
    - **show ip ospf interface [brief]**
  - Verify OSPF adjacencies
    - **show ip ospf neighbor**
    - **debug ip ospf adj**
  - Verify OSPF database
    - **show ip ospf database [router | network | summary | …]**

Troubleshooting OSPF Adjacencies

- Where can problems arise?
  - Transport problems
  - Attribute negotiation problems
- Useful troubleshooting commands
  - show ip ospf neighbor
  - show ip ospf database
  - debug ip ospf adj
  - debug ip packet
    - ***Use with caution***

OSPF Adjacency State Machine

- Normal OSPF Adjacency State Machine Order
  - Down/Attempt
  - Init
  - 2-Way
    - Stop here for DROthers
  - ExStart
  - Exchange
  - Loading
  - Full

OSPF Adjacencies Attributes

- Unique OSPF Adjacency Attributes
  - Router-ID
  - Interface IP Address
- Common OSPF Adjacency Attributes
  - Interface Area-ID
  - Hello interval & dead interval
  - Interface network address
  - Interface MTU
  - Network Type
  - Authentication
  - Stub Flags
  - Other optional capabilities

In This Section

- IS-IS Routing on NX-OS

IS-IS Overview

- Intermediate System to Intermediate System
  - "Router to Router" communication
  - Link-State IGP similar to OSPF
- Typically used in core of SP networks
  - Simple flat network design
  - Highly scalable
  - Supports both IPv4 and IPv6 routing
- Not an IP protocol - Part of the CLNS stack
  - Integrated IS-IS: IP extensions to IS-IS

IS-IS NET Addressing

- Network Entity Title
  - Essentially CLNS Router-ID
- Uses ISO NSAP Addressing Format
  - Maximum 20 bytes
  - Minimum 8 bytes
- NET format
  - AA.AAAA.AAAA.AAAA.AAAA.AAAA.AAAA.SSSS.SSSS.SSSS.NN
  - **A**rea – not link-state area like OSPF
  - **S**ystem-ID - Router-ID inside the area
  - **N**-Selector - always zero

IS-IS Adjacency Levels

- IS-IS uses two "levels" of adjacency
  - Level 2 (L2)
  - Level 1 (L1)
- IS-IS Level 2
  - Inter or intra area adjacency
  - Like area 0 in OSPF
    - Must be contiguous
    - Cisco IOS/XE/XR/NX-OS does not support IS-IS virtual links
- IS-IS Level 1
  - Intra area adjacency only
  - Like a not so totally stubby area in OSPF
    - Intra area routes
    - Default route out
    - Redistribution allowed

Level 1 / Level 2 Routing

- Level 1 / Level 2 (L1/L2) Router
  - Like ABR in OSPF
  - Used as exit point from L1 to L2
- Injects default route into level-1
  - Sets the “attached” bit

IS-IS Level Manipulation

- Process & interfaces default to Level-1-2
  - Forms both L1 and L2
  - Separate LSP databases
  - Double the overhead
- Level can be defined...
  - Global under the process
    - Affects all interfaces
  - Under the interface
    - Affects only that interface

IS-IS Network Type

- Only two network types
- Broadcast
  - Default on multipoint interfaces
  - Uses DIS instead of DR / BDR
- Point-to-point
  - Default on point-to-point interfaces

DIS Election

- Designated Intermediate System
- Like OSPF DR / BDR
  - No backup DIS
- Election is dynamic, preemption can occur
  - Separate election for L1 and L2
  - Occurs by
    - Highest priority
    - Highest SNPA (MAC) address

Forming IS-IS Adjacency

- Level of adjacency must match
  - Area must match if L1 adjacency
- MTU must match
  - IS-IS pads hellos to interface MTU
  - Different behavior than OSPF but same result
- Network type must match
  - Broadcast
  - Point-to-Point

Configuring IS-IS on NX-OS

- Enable global IS-IS process
  - **feature isis**
  - **router isis [process-id]**
- Define Network Entity Title
  - **net [nsap]**
- Enable interface IS-IS process
  - Interface level **ip router isis [process-id]**

Verifying IS-IS on NX-OS

- Common Verificaitons
  - Verify IS-IS is enabled
    - **show isis**
    - **show isis interface [brief]**
  - Verify IS-IS adjacencies
    - **show is-is adjacency**
    - **debug isis adjacency**
    - **debug isis iih**
  - Verify IS-IS
    - **show isis database [detail]**

IS-IS Path Selection

- Link cost depends on OS
  - E.g. IOS vs. XR vs. NX-OS
  - Can be manually modified
- Neighbors must agree on metric *style*
  - Narrow
    - Default
  - Wide
    - Needed for MPLS TE and IPv6
  - Transition
- Level 1 paths preferred over Level 2 paths
  - Like OSPF Intra-Area over Inter-Area

IS-IS Optimizations

- IS-IS by default runs both Level-1 and Level-2
  - Redundant database information
  - CLOS fabric is a flat link state flooding domain
  - IS-IS should use L1 or L2, but not both
- Ethernet is network type Broadcast by default
  - In physical P2P designs, IS-IS DIS / OSPF DR is redundant
  - IS-IS should run as network-type point-to-point

IS-IS Optimizations (cont.)

- Convergence is timers based
  - Hello interval and multiplier, similar logic to OSPF
  - BFD should be used for faster neighbor failure detection
- IS-IS supports two types of authentication
  - Global authentication of LSPs
  - Interface authentication of Hellos

IS-IS Optimizations (cont.)

- IS-IS Overload Bit
  - Signals other routers not to include this node as transit in the SPT
  - Useful for performing a maintenance window on a spine or leaf
- IS-IS supports two types of authentication
  - Global authentication of LSPs
  - Interface authentication of Hellos

In This Section

- BGP Routing on NX-OS

BGP in the VXLAN Underlay

- What are we trying to accomplish?
  - BGP is the application to advertise the VXLAN L2 VNI and L3 VNI information
  - MAC to L2 VNI to VTEP mapping
  - IP to L3 VNI to VTEP mapping
- What other considerations do we have?
  - How are the BGP peerings formed?
  - What are the BGP NLRI advertisement rules?
  - How does next-hop processing affect VXLAN?

Basic BGP Workflow

- Establish TCP Transport
- Establish BGP Peerings
- Negotiate Address Families
- Advertise NLRI
- Apply BGP Policy

Establishing TCP Transport

- Unlike IGP, BGP does not use its own transport
  - Uses TCP Port 179
- Within our scope, typically implies either...
  - Peers are directly connected
  - IGP transport is already established
- TTL is a transport consideration
  - iBGP, EBGP, Multihop EBGP

Establishing BGP Peerings

- BGP peers must agree upon...
  - AS numbers
    - Global, local, private, confed sub-as, etc.
  - Update source
    - Loopback is MPLS tunnel destination
  - Address Families
    - IPv4 Unicast, L2VPN EVPN, etc.
  - Misc.
    - Authentication, TTL Security, etc.

Negotiating Address Families

- BGP transport is independent of NLRI
  - E.g. IPv4 transport can be used to advertise IPv6 NLRI
- AFI/SAFIs define which NLRI is exchanged
  - IPv4 Unicast, VPNv4 Unicast, L2VPN EVPN, etc.
- AFI/SAFIs must match in the capabilities exchange
  - I.e. peerings can't form if AFI/SAFI mismatches

Advertising NLRI

- Once peering is established and AFI/SAFIs are negotiated, BGP updates are exchanged
- Updates (NLRI) can be originated multiple ways
  - Network statement, Redistribution, Conditional Advertisement, Conditional Route Injection, etc.
- Key NLRI attributes
  - Prefix/len
  - Next-hop
  - VPN Route Distinguisher (RD)
  - VPN Route Target (RT)

NLRI Advertisement Rules

- Advertisement rules change depending on peering type
  - EBGP
  - iBGP
  - iBGP RR Client
  - iBGP RR Non-Client
  - Confed EBGP
- Next-hop rules change depending on peering and AFI/SAFI
  - EBGP to iBGP in IPv4/IPv6 Unicast
  - EBGP to iBGP in VPNv4/VPNv6 Unicast
  - iBGP to iBGP
  - Multihop EBGP in VPNv4/VPNv6 Unicast