Cisco live!

June 25–29, 2017 · Las Vegas, NV

Advanced Concepts of DMVPN (Dynamic Multipoint VPN)

Mike Sullenberger – Distinguished Engineer

BRKSEC-4054

Cisco live!

Cisco Spark

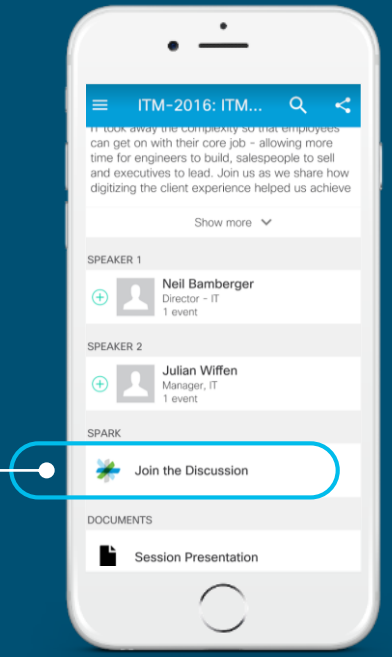
Questions?

Use Cisco Spark to communicate with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

Cisco Spark spaces will be available until July 3, 2017.



cs.co/ciscolivebot#BRKSEC-4054

VPN and IWAN Breakout Sessions

• BRKRST-2362 - IWAN - Implementing Performance Routing (PfRv3) • BRKCRS-2000 - Intelligent WAN (IWAN) Architecture • BRKSEC-3001 - Advanced IKEv2 Protocol • BRKSEC-2054 - GET Your VPN's Secured with ESON • BRKRST-2042 - Highly Available Wide Area Network Design	**Monday**
• BRKRST-2041 - WAN Architectures and Design Principles • BRKCRS-2002 - IWAN Design and Deployment Workshop • BRKSEC-3054 - IOS FlexVPN Remote Access, IoT and Site-to-Site advanced Crypto VPN Designs • BRKCRS-2007 - Migrating Your Existing WAN to Cisco's IWAN • BRKSEC-3052 - Demystifying DMVPN • BRKRST-3413 - IWAN Serviceability: Deploying, Monitoring, and Operating • BRKRST-2557 - IWAN and NFV Orchestration for Managed Service Providers	**Tuesday**
• BRKRST-3018 - Understanding and Troubleshooting Intelligent Path Control in IWAN • BRKSEC-4054 - Advanced Concepts of DMVPN	**Wednesday**
• BRKSEC-3005 - Cryptographic Protocols and Algorithms - a review	**Thursday**

Agenda

- DMVPN Design Overview
- DMVPN Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- DMVPN Network Segmentation
  - VRF-lite over DMVPN
  - MPLSoDMVPN

DMVPN Design Overview

Agenda

- DMVPN Design Overview
- DMVPN Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- DMVPN Network Segmentation
  - VRF-lite over DMVPN
  - MPLSoDMVPN

What is Dynamic Multipoint VPN?

DMVPN is a Cisco IOS software solution for building IPsec+GRE VPNs in an easy, dynamic and scalable manner

- Uses two proven technologies
  - Next Hop Resolution Protocol (NHRP)
    - Creates a distributed mapping database of VPN (tunnel int.) to real (public int.) addresses
  - Multipoint GRE Tunnel Interface
    - Single GRE interface to support multiple GRE/IPsec tunnels and endpoints
    - Simplifies size and complexity of configuration
    - Supports dynamic tunnel creation

DMVPN Philosophy

- Distributed NHRP database
  - No single point must have all the NHRP information for the DMVPN
  - No single point limits the overall size of the DMVPN
- Don’t drop packets while building dynamic tunnels
  - Pre-build (hierarchical) hub-and-spoke network
  - Forward data packets via pre-built path until direct tunnel is ready
- Dynamic Mesh versus Full Mesh
  - Small nodes participate in large DMVPNs up to their capabilities
  - Doesn’t limit the participation of other (larger) nodes in the DMVPN

DMVPN Major Features

- Configuration reduction and no-touch deployment
- Supports:
  - Passenger protocols (IP(v4/v6) unicast, multicast and dynamic Routing Protocols)
  - Transport protocols (NBMA) (IPv4 and IPv6)
  - Remote peers with dynamically assigned transport addresses.
  - Spoke routers behind dynamic NAT; Hub routers behind static NAT.
- Dynamic spoke-spoke tunnels for partial/full mesh scaling.
- Can be used without IPsec Encryption
- Works with MPLS; GRE tunnels and/or data packets in VRFs and MPLS switching over the tunnels
- Wide variety of network designs and options.

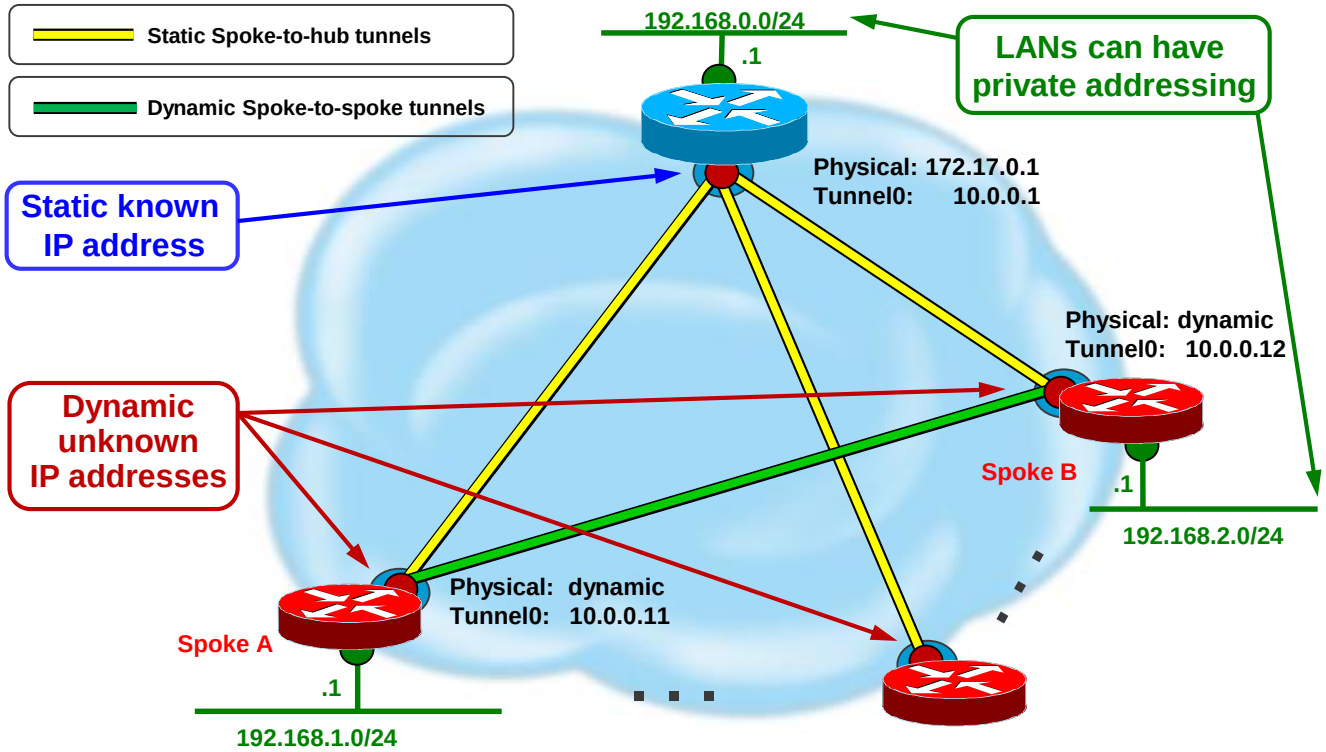
DMVPN Phases

Phase 1 – 12.2(13)T	Phase 2 – 12.3(4)T (Phase 1 +) IWAN 1.0	Phase 3 – 12.4.(6)T (Phase 2 +) IWAN 2.0
• Hub and spoke functionality • mGRE or p-pGRE interface on spokes, mGRE on hubs • Simplified and smaller configuration on hubs • Support dynamically addressed CPEs (NAT) • Support for routing protocols and multicast • Spokes don't need full routing table – can summarize on hubs • No touch deployment	• Spoke to spoke functionality • mGRE interface on spokes • Direct spoke to spoke data traffic reduces load on hubs • Hubs must interconnect in daisy-chain • Spoke must have full routing table – no summarization • Spoke-spoke tunnel triggered by spoke itself • Routing protocol limitations	• More network designs and greater scaling • Same Spoke to Hub ratio • No hub daisy-chain • Spokes don't need full routing table – can summarize • Spoke-spoke tunnel triggered by hubs • Remove routing protocol limitations • NHRP routes/next-hops in RIB (15.2(1)T)

DMVPN How it works

- Spokes build a dynamic permanent GRE/IPsec tunnel to the hub, but not to other spokes. They register as clients of the NHRP server (hub).
- When a spoke needs to send a packet to a destination (private) subnet behind another spoke, it queries via NHRP for the real (outside) address of the destination spoke.
- Now the originating spoke can initiate a dynamic GRE/IPsec tunnel to the target spoke (because it knows the peer address).
- The dynamic spoke-to-spoke tunnel is built over the mGRE interface.
- When traffic ceases then the spoke-to-spoke tunnel is removed.

DMVPN Example



DMVPN and IPv6

- IPv6 Passenger over DMVPN (IPv4 or IPv6) Transport
  - IPv6 Passenger Addresses:
    - NHRP requires IPv6 Unicast Global
    - Routing Protocol requires IPv6 Link-local
    - NHRP automatically registers both Unicast Global and Link-local Addresses
  - IPv4 **or** IPv6 infrastructure transport network (separate mGRE tunnel interfaces)
  - Both IPv4 **and** IPv6 (dual stack) can be over the same DMVPN mGRE tunnel
- (IPv4 and/or IPv6) Passenger over DMVPN IPv6 Transport
  - Use IKEv2 for IPsec encryption key management
  - Standard IPv6 configuration on Outside (WAN) interface
  - IPv4 **and** IPv6 transports require separate DMVPNs (mGRE tunnels)
  - DMVPN IPv4 ←→ DMVPN IPv6 spoke to spoke via hub
  - WAN interface may support both IPv4 and IPv6 (dual stack)

DMVPN and IPv6 – Configuration

```
crypto ikev2 keyring DMVPN
  peer DMVPNv6
    address ::/0
    pre-shared-key cisco123v6
  peer DMVPNv4
    address 0.0.0.0 0.0.0.0
    preshared-key cisco123v4
!
crypto ikev2 profile DMVPNv6
  match identity remote address ::/0
  authentication local pre-share
  authentication remote pre-share
  keyring DMVPN
  dpd keepalive 30 5 on-demand
crypto ikev2 profile DMVPNv4
  match identity remote address 0.0.0.0 0.0.0.0
  authentication local pre-share
  authentication remote pre-share
  keyring DMVPN
  dpd keepalive 30 5 on-demand
!
crypto ipsec profile DMVPNv6
  set transform-set DMVPN
  set ikev2-profile DMVPNv6
!
crypto ipsec profile DMVPNv4
  set transform-set DMVPN
  set ikev2-profile DMVPNv4
…
interface Serial1/0
  ip address 172.16.1.1 255.255.255.252
  ipv6 address 2001:DB8:0:FFFF:0:1:0:1/126
```

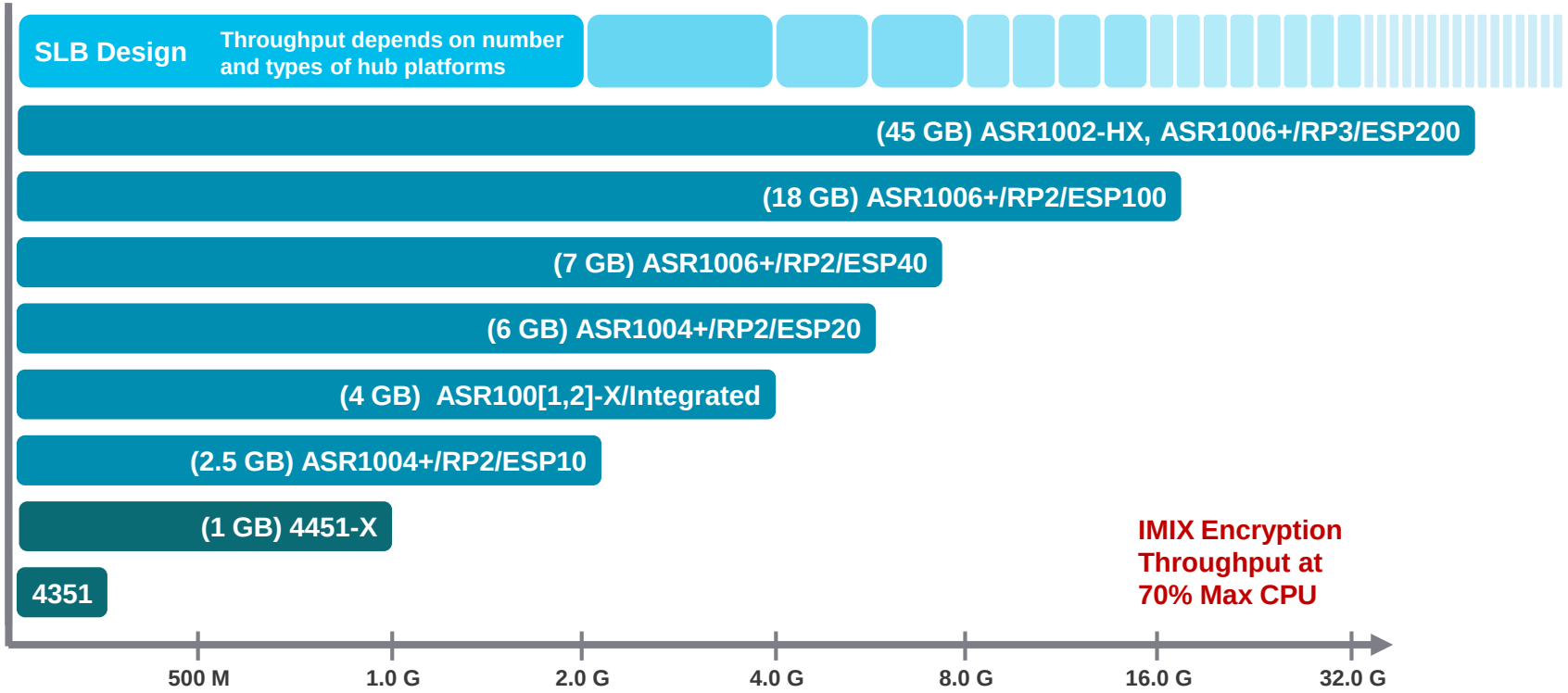
```
interface Tunnel0
  ip address 10.0.0.11 255.255.255.0
  …
  ip nhrp network-id 100000
  ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
  …
  ipv6 address 2001:DB8:0:100::B/64
  …
  ipv6 nhrp network-id 100006
  ipv6 nhrp nhs 2001:DB8:0:100::1 nbma 172.17.0.1 multicast
  …
  tunnel source Serial1/0
  tunnel mode gre multipoint
  tunnel protection ipsec profile DMVPNv4
!
interface Tunnel1
  ip address 10.0.6.11 255.255.255.0
  …
  ip nhrp network-id 100000
  ip nhrp nhs 10.0.6.1 nbma 2001:DB8:0:FFFF:1::1 multicast
  …
  ipv6 address 2001:DB8:0:106::B/64
  …
  ipv6 nhrp network-id 100006
  ipv6 nhrp nhs 2001:DB8:0:106::1 nbma 2001:DB8:0:FFFF:1::1 multicast
  …
  tunnel source Serial1/0
  tunnel mode gre multipoint ipv6
  tunnel protection ipsec profile DMVPNv6
!
ip route 0.0.0.0 0.0.0.0 Serial1/0
ipv6 route ::/0 Serial1/0
```

DMVPN and IPsec

- IPsec integrated with DMVPN, but not required
- Packets Encapsulated in GRE, then Encrypted with IPsec
  - Supports both IKEv1 (ISAKMP) and IKEv2
- NHRP controls the tunnels, IPsec does the encryption
- Bringing up a tunnel
  - NHRP signals IPsec to setup encryption
  - ISAKMP/IKEv2 authenticates peer, generates SAs
  - IPsec responds to NHRP and the tunnel is activated
  - All NHRP and data traffic is Encrypted
- Bringing down a tunnel
  - NHRP signals IPsec to tear down tunnel
  - If encryption is cleared or lost IPsec can signal NHRP to clear the tunnel
- ISAKMP/IKEv2 Keepalives monitor remote crypto peers*

*** BFD over DMVPN**

DMVPN Encryption Scaling



Routing over DMVPN

- Supports all routing protocols, except ISIS
- Hubs are routing neighbors with spokes
  - Receive spoke network routes from spokes
  - Advertise spoke and local networks to **all** spokes
    - Phase 1 & 3: Can Summarize (except OSPF)
    - Phase 2: Cannot summarize (OSPF limited to 2 hubs)
- Hubs are routing neighbors with other hubs
    - Phase 1: Can use different interface and routing protocol than hub-spoke tunnels
    - Phase 2: Must use same tunnel interface and routing protocol as hub-spoke tunnels
    - Phase 3: Can use different tunnel interface and routing protocol than hub-spoke tunnels
- Spokes are only routing neighbors with hubs, **not** with other spokes
    - Phase 3: Spoke-spoke NHRP routes are added by NHRP directly to routing table (15.2(1)T)

Routing Protocols over DMVPN

EIGRP

- Distance Vector style matches with DMVPN NBMA network style
  - Feasible successor for quick spoke-to-hub convergence
- Good scaling with reasonably fast convergence (hello 5, hold 15)
- Good metric control (automatic and/or manual)
  - Change metrics, route tagging, filtering or summarization at hub and/or spoke
  - Can be used to control load-balancing of spoke ←→ hub(s) traffic
  - Automatic metric increase per DMVPN hop
- Feature additions – spoke-spoke load-balance support
  - Equal Cost MultiPath (15.2(3)T, 15.2(1)S)
  - Add-path (15.3(1)S)

Routing Protocols over DMVPN

BGP

- Base Distance Vector style matches with DMVPN NBMA network style
  - iBGP (recommended)
    - Dynamic Neighbors, MED to control/compare routes;
    - May need iBGP local-as (15.2(2)T, 15.1(3)S)
  - eBGP (okay)
    - AS-Path length to control/compare routes
- Good scaling but with slower convergence (hello 15+, hold 45+)
- Good metric control (manual)
  - Change metrics, route tagging, filtering or summarization at hub and/or spoke
  - Can be used to control load-balancing of spoke ←→ hub(s) traffic
  - Only manual metric increase per DMVPN hop
- Some issues with Equal Cost multi-path (ECMP) route selection
  - Between multiple DMVPNs and preserving correct next-hop
  - Spoke-spoke tunnel load-balancing for spoke sites with multiple spoke routers

Routing Protocols over DMVPN

OSPF

- Link-state style doesn't match as well with DMVPN NBMA network style
- Area issues – DMVPN requires single Area
  - Area 0 over DMVPN – spoke sites can be in different areas
    - But, area 0 is extended over WAN – possible stability issues for Area 0
  - Non-Area 0 over DMVPN – all spoke sites in this same single area
  - Multi-subnet DMVPN can be used to have multiple OSPF areas
    - Increase in complexity of DMVPN and OSPF design
- More difficult metric control
  - Can only change metrics, filter or summarize at area boundaries
  - Automatic metric increase per DMVPN hop
  - Issue for failover path between multiple DMVPNs – slightly reduce Hub vs. Spoke cost
- No issues with Equal Cost multi-path (ECMP) route selection

Dynamic Routing Protocols

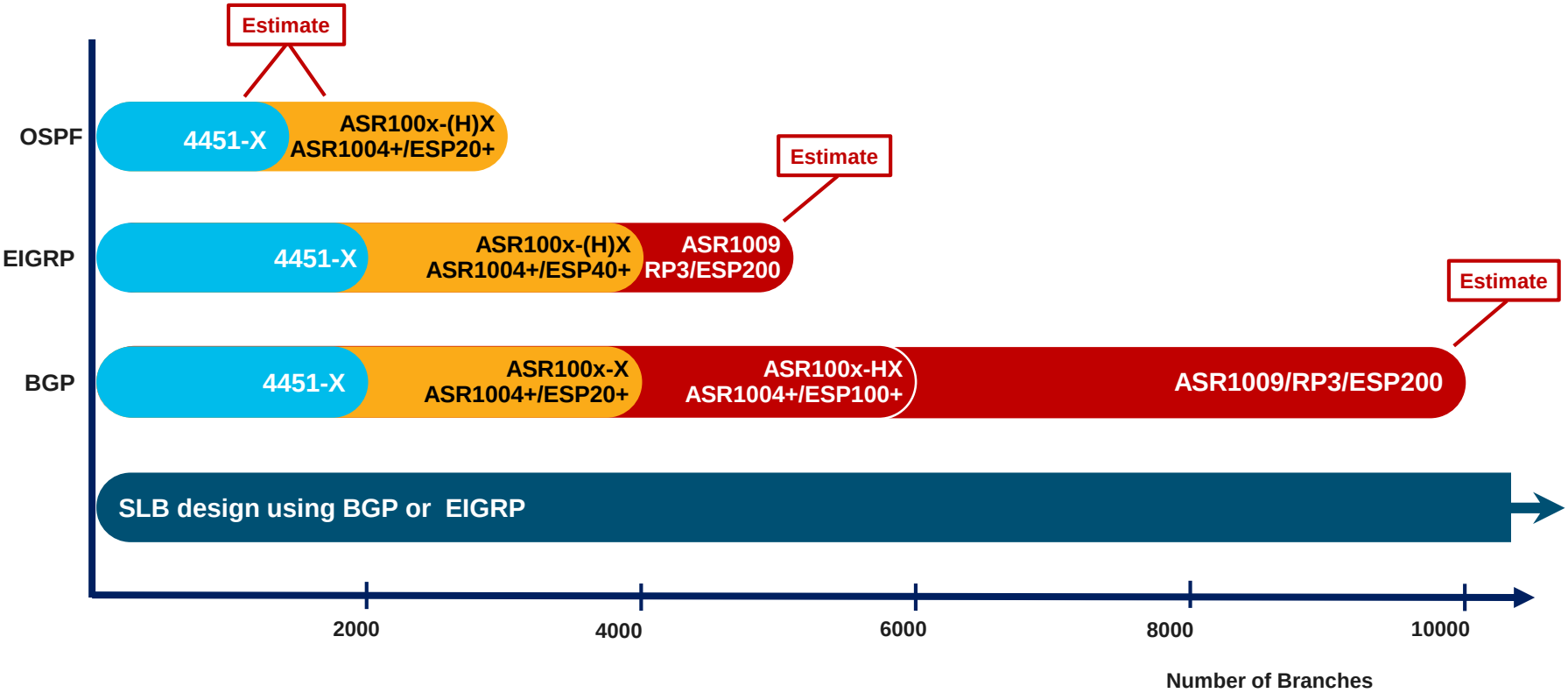
	Routing Protocol Type	Converge (hello/hold)	Route/Metric Control	Scaling (ASR1k neighbors per hub)	Notes
OSPF	Link-State[1] (multicast)	Faster (5/15) to (20/60)	Fair • Summarize and metric control **only** at area border • Automatic per hop metric increase	Low (1500-2000) (dynamic)	Single Area over DMVPN • Area = 0 (spokes in different areas) • Area ≠ 0 (all spokes in same area)
EIGRP	Distance Vector (multicast)	Faster (5/15) to (20/60)	Good • Summarize and metric control at **any** node • Automatic per hop metric increase	Medium (4000-6000) (dynamic)	Spokes: Stub/Stub-site • Suppress EIGRP Queries
BGP	Distance Vector (unicast)	Slower (15/45) to (60/180)	Good • Summarize and metric control at **any** node • Manual metric control	High (6000-10000) (iBGP: dynamic) (eBGP: static)	iBGP • Hubs: route-reflector; iBGP local-AS; Dynamic neighbors; • Metric: MED, Local-pref eBGP • Metric: AS Path-length, Local-pref

[1] Link-State **is not** a good match for NBMA style (hub-and-spoke) networks like DMVPN

Routing Protocol?

- Which routing protocol should I use?
  - In general you would use the same routing protocol over DMVPN that you use in the rest of your network, or over other WAN networks (like MPLS).
- BUT...
  - EIGRP being an advanced distance vector protocol matches really well with DMVPN network topologies
  - BGP, specifically iBGP, runs well over DMVPN, but is more complicated to setup to have it act more like an IGP than an EGP
  - OSPF can run over DMVPN, BUT lower scaling and Area 0 issues can complicate the network
  - RIP can be used, but has longer hold time and limited metric values
  - IS-IS cannot be used since it doesn’t run over IP

Routing Protocol Scaling



Redundancy

- Active-active redundancy model – two or more hubs per spoke
  - All configured hubs are active and are routing neighbors with spokes
    - Can use Backup NHS feature to activate a subset of configured hubs
    - Can use 'if-state nhrp' and 'backup interface ...' to disable/enable a backup tunnel interface
  - Routing protocol routes are used to determine traffic forwarding
    - Single route: one tunnel (hub) at a time – primary/backup mode
    - Multiple routes: multiple tunnels (hubs) – load-balancing mode (CEF, PfR)
- (ISAKMP/IKEv2)/IPsec
  - Cannot use IPsec Stateful failover (NHRP isn't supported)
  - Invalid SPI recovery is not useful with DMVPN

    no crypto isakmp invalid-spi-recovery
  - ISAKMP/IKEv2 keepalives on spokes for DPD
- BFD over DMVPN for quicker spoke-hub and spoke-spoke failure discovery

Redundancy (cont)

- Can use single or multiple DMVPNs for redundancy
  - Each mGRE interface is a separate DMVPN network using
    - Same: Tunnel source (optional).
    - Different: NHRP network-id and IP subnet, Tunnel key
    - When using same tunnel source → different tunnel keys, same IPsec profile (*name*) and shared
      tunnel protection ipsec profile *name* shared
  - Can "glue" mGRE interfaces into same DMVPN network (Phase 3 only)
    - Same: NHRP network-id and authentication, Tunnel key (optional)
    - Different: Tunnel source and IP subnet
- Spokes – two or more hubs (NHSs)
  - Phase 1: (Hub-and-spoke)
    - p-pGRE interfaces → two or more DMVPN networks, one hub (NHS) on each
  - Phase 1, 2 or 3: (Hub-and-spoke or Dynamic Mesh)
    - mGRE interface → one DMVPN network, two or more hubs (NHSs)

Redundancy (cont.)

- Hubs – interconnect and routing
  - Phase 1: (Hub and spoke only)
    - Interconnect hubs directly over physical link, p-pGRE or mGRE tunnel
    - Can exchange routing through any of these paths
    - Same or different routing protocol as with spokes
  - Phase 2: (Dynamic Mesh)
    - Must interconnect hubs over same mGRE tunnel as spokes, daisy-chain as NHSs
    - Must exchange routing over DMVPN network
    - Must use same routing protocol as with spokes
  - Phase 3: (Dynamic Mesh)
    - Interconnect hubs over same or different mGRE tunnel (same NHRP Network-id)
    - Must exchange routing over a DMVPN network
    - Same or different routing protocol as with spokes

Spoke-Spoke and Spoke-Hub Tunnels
Considerations

- Resiliency
  - BFD over DMVPN for quick spoke-hub and/or spoke-spoke tunnel recovery
  - Can also use ISAKMP/IKEv2 keepalives, but doesn't test data channel – **Spokes only**

    crypto {isakmp keepalive | ikev2 dpd} *initial retry* [on-demand | periodic] **(Recommend: initial=30, retry=5)**

    crypto {isakmp | ikev2} nat keepalive *interval* **(Recommend: interval=30)**
- Path Selection
  - NHRP will always try to build spoke-spoke tunnel
    - No bandwidth/latency measurement of spoke-spoke vs. spoke-hub-spoke paths
    - Can do interesting things with Smart-spoke feature
- Overloading routers
  - CPU or memory → IKE Call Admission Control (CAC) – **Hubs**

    crypto call admission limit ike {sa | in-negotiation} *max-SAs* **(Default: no-limit)**

    crypto ikev2 limit max-in-negotiation-sa *max-SAs* {inbound | outbound} **(Default: inbound: 40, outbound: 400)**

    show crypto call admission statistics
  - Bandwidth → Design for expected traffic
    - Hub-spoke versus Spoke-spoke; Spoke-spoke availability is best effort

Best Practices

- mGRE Tunnel configuration
  - Both Hubs and Spokes
    - tunnel source *interface-name*
    - bandwidth <from WAN-interface> (as starting point, may adjust)
    - ip mtu 1400; ip tcp adjust-mss 1360
- NHRP
  - Spokes
    - ip nhrp holdtime 600*
    - ip nhrp shortcut*
    - ip nhrp nhs {*hub-tunnel-ip* | dynamic} nbma {*hub-nbma-ip* | *hub-fqdn*} multicast (12.4(20)T)
  - Hubs
    - ip nhrp redirect
    - ip nhrp map multicast dynamic*
    - ip nhrp server-only

*** Default in 16.3**

Best Practices (cont)

- Routing
  - Phase 2 – RP advertises routes with remote spoke as the next-hop
    - EIGRP: (hubs) no ip [next-hop-self | split-horizon] eigrp *as*, (all) use delay to adjust metric
    - OSPF: (all) ip ospf network broadcast; (spokes only) ip ospf priority 0
    - BGP: iBGP (hubs) route-reflectors; (spokes) neighbor *hub* next-hop-self
  - Phase 1 & 3 – RP advertises routes with the hub as the next-hop
    - EIGRP: (hubs) no ip split-horizon eigrp <as>
    - OSPF: (all) ip ospf network point-multipoint; prefix-suppression (suppress /32 routes)
    - BGP: iBGP (hubs) route-reflectors; (all) neighbor [*hub* | *spoke*] next-hop-self **all**
  - To manipulate path selection through DMVPN use:
    - EIGRP: delay not bandwidth; OSPF: cost; iBGP: MED, Local-pref

Cisco IOS Code and Platform Support

* Recommended

- 3900(E), 2900, 1900, 890, 819, 880
  - 15.3.3M9*, 15.4.3M7*, 15.5.3M5*, 15.6.3M2+
  - 15.4.2T4, 15.5.2T4+, 15.6.2T2+

+ N/A for 881-887

- ASR1002-X, ASR100[4,6,6-X,9-X,13](RP2), 4451-X, 4431, 4300
  - (3.13.7S)154-3.S7*, (3.15.4S)155-2.S4, (3.16.5S)155-3.S5*, (3.17.3S)156-1.S3
  - Denali: 16.3.3, Everest: 16.4.2, 16.5.1b
- ASR100[1,2]-HX, ASR100[6,9]-X(RP3), ASR1013(RP3), 4221
  - Denali: 16.3.3+, Everest: 16.4.2, 16.5.1b

+ N/A for 4221

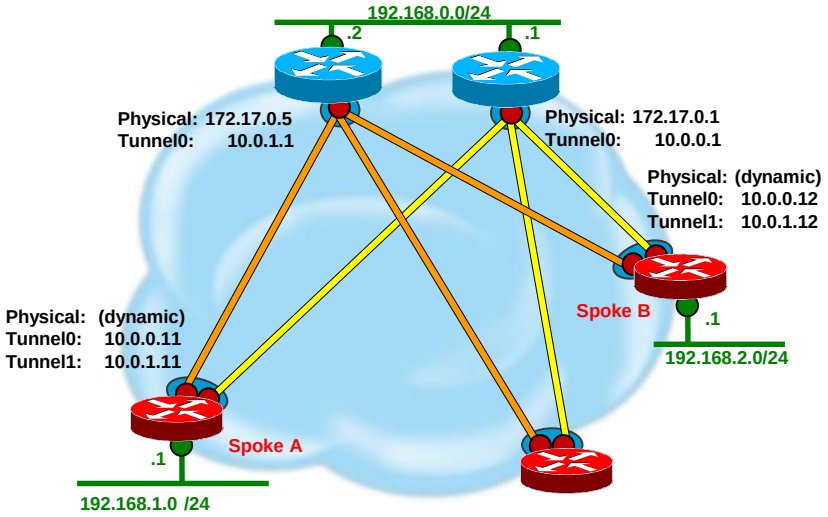
- CSR1000V
  - (3.13.7S)154-3.S7, (3.15.4S)155-2.S4, (3.16.5S)155-3.S5*, (3.17.3S)156-1.S3
  - Denali: 16.3.3, Everest: 16.4.2, 16.5.1b

Basic DMVPN Designs

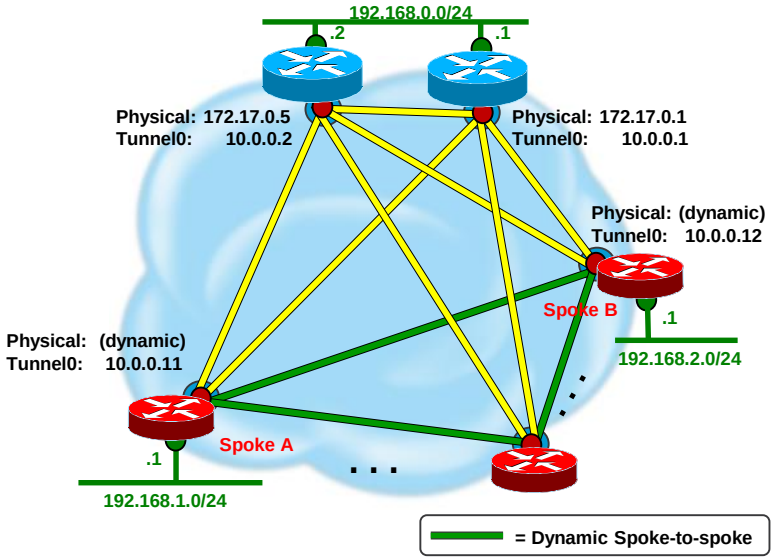
- Hub-and-spoke – Order(n)
  - Spoke-to-spoke traffic via hub
    - Phase 1: Hub bandwidth and CPU limit VPN
    - SLB: Many “identical” hubs; increases CPU and bandwidth limits
- Spoke-to-spoke – Order(n) « Order(n^2)
  - Control traffic; Hub and spoke; Hub to hub
    - Phase 2: (single)
    - Phase 3: (hierarchical)
  - Unicast Data traffic; Dynamic mesh
    - Spoke routers support spoke-hub and spoke-spoke tunnels currently in use.
    - Hub supports spoke-hub traffic and overflow from spoke-spoke traffic.
- Network Virtualization
  - VRF-lite; Multiple DMVPNs (one per VRF)
  - MPLS over DMVPN (2547oDMVPN); Single DMVPN (many VRFs)

Basic DMVPN Designs

Dual DMVPN Single Hub
Single mGRE tunnel on Hub,
two p-pGRE tunnels on Spokes



Single DMVPN Dual Hub
Single mGRE tunnel on all nodes

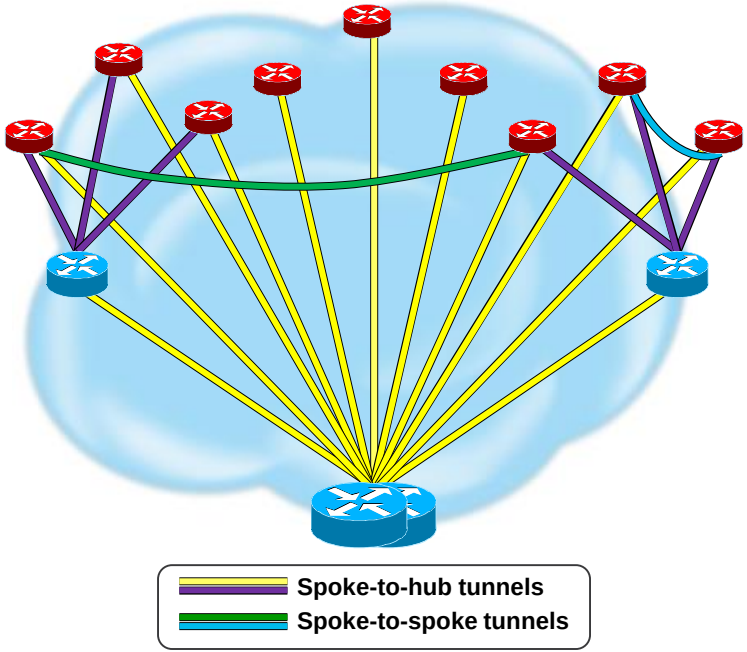


Multiple DMVPNs versus Single DMVPN

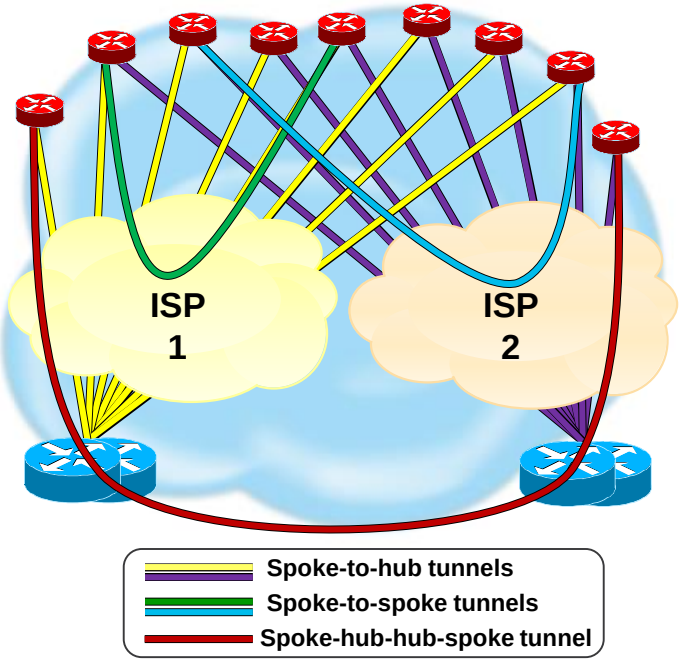
- Multiple DMVPNs
  - Best for Hub-and-spoke only
    - Easier to manipulate RP metrics between DMVPNs for Load-sharing
      - EIGRP – Route tags, Delay; iBGP – Communities, MED; OSPF – Cost
    - Performance Routing (PfR) selects between interfaces
  - Load-balancing over multiple ISPs (physical paths)
    - Load-balance data flows over tunnels → Better statistical load-balancing
- Single DMVPN
  - Best for spoke-spoke DMVPN
    - Can only build spoke-spoke within a DMVPN not between DMVPNs*
    - Slightly more difficult to manipulate RP metrics within DMVPN for Load-sharing
      - EIGRP – Route tags, delay; iBGP – Communities, MED; OSPF – Can't do
  - Load-balancing over multiple ISPs (physical paths)
    - Load-balance tunnel destinations over physical paths → Worse statistical load-balancing

DMVPN Combination Designs

Retail/Franchise

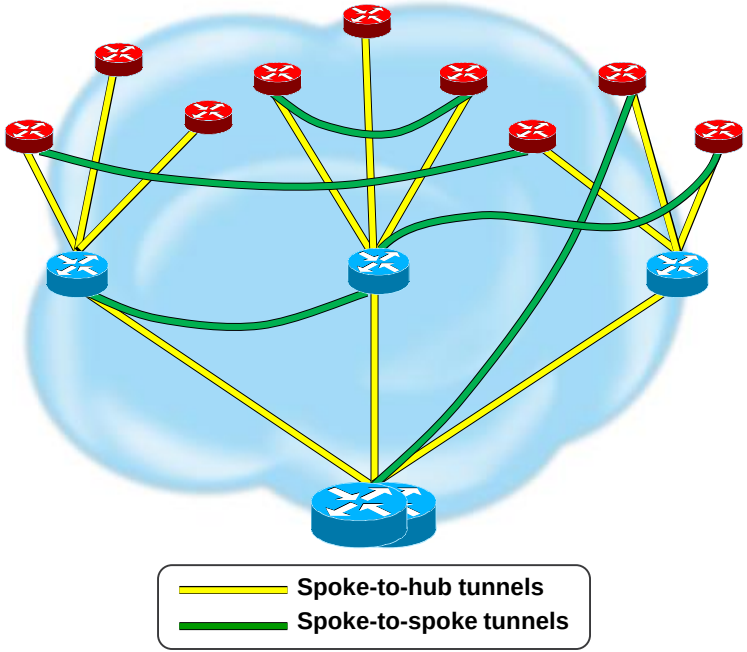


Dual ISP

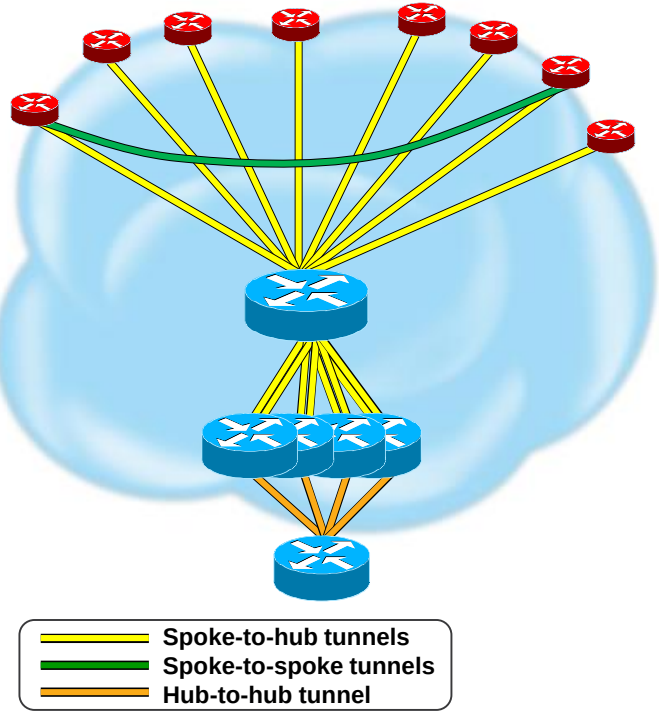


DMVPN Combination Designs (cont)

Hierarchical



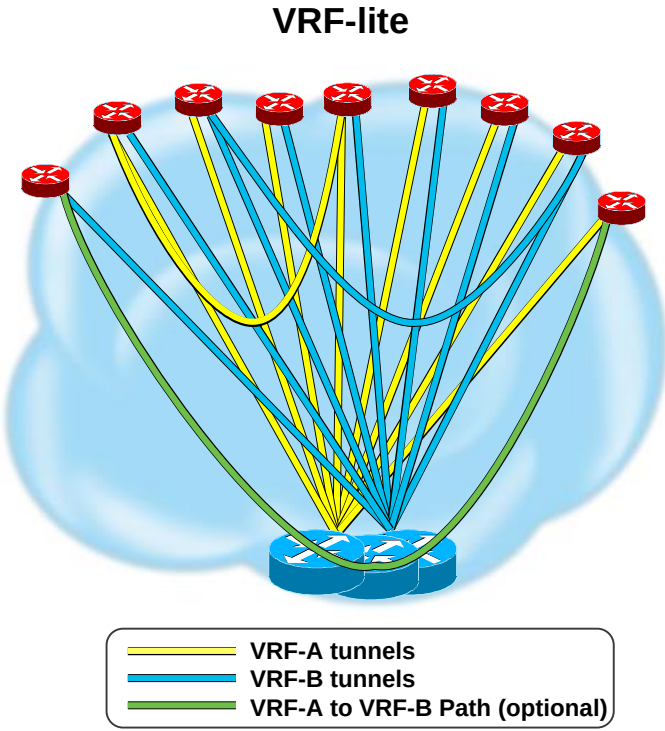
Server Load Balancing



Network Virtualization

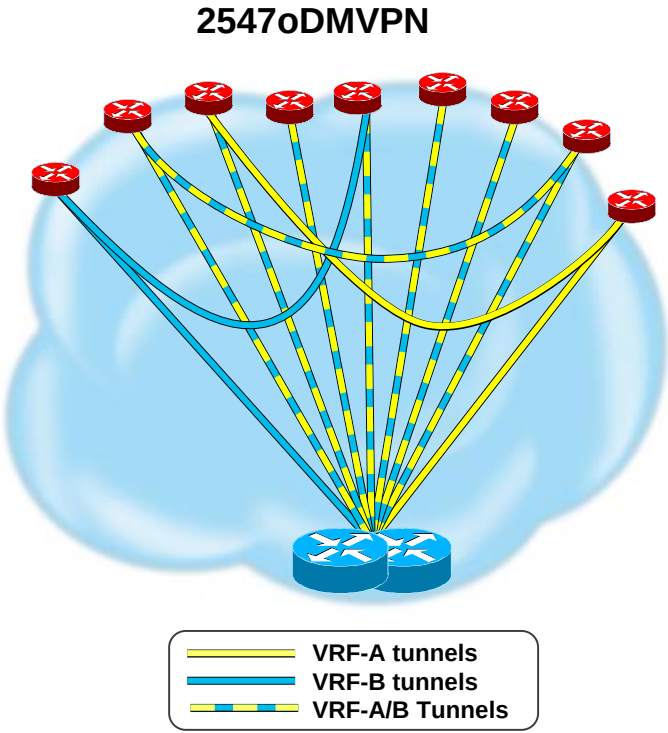
Separate DMVPN mGRE tunnel per VRF (VRF-lite)

- Hub routers handle all DMVPNs
  - Multiple Hub routers for redundancy and load
- IGP used for routing protocol over DMVPNs on Spokes and Hubs
  - Address family per VRF
  - Routing neighbor per spoke per VRF
- BGP used only on the hub
  - Redistribute between IGP and BGP for import/export of routes between VRFs
  - “Internet” VRF for Internet access and routing between VRFs
- Global routing table used for routing DMVPN tunnel packets



Network Virtualization
MPLS over DMVPN – 2547oDMVPN

- MPLS VPN over DMVPN
  - Single DMVPN/mGRE tunnel on all routers
  - Multiple Hub routers for redundancy and load
- MPLS configuration – routers are PEs
  - Spoke to spoke via hub **and** direct shortcut
  - MPLS labels via NHRP, 'mpls nhrp' **(15.4(1)S, 15.4(2)T)**
    - Replaces 'mpls ip'; No LDP
- Routing
  - Global for routing DMVPN tunnel packets
  - IGP for routing outside of DMVPN
  - MP-BGP for routing over DMVPN
    - Redistribute between IGP and BGP for over DMVPN
    - Import/export routes between VRFs and Global (or Internet VRF)
    - One routing neighbor per spoke

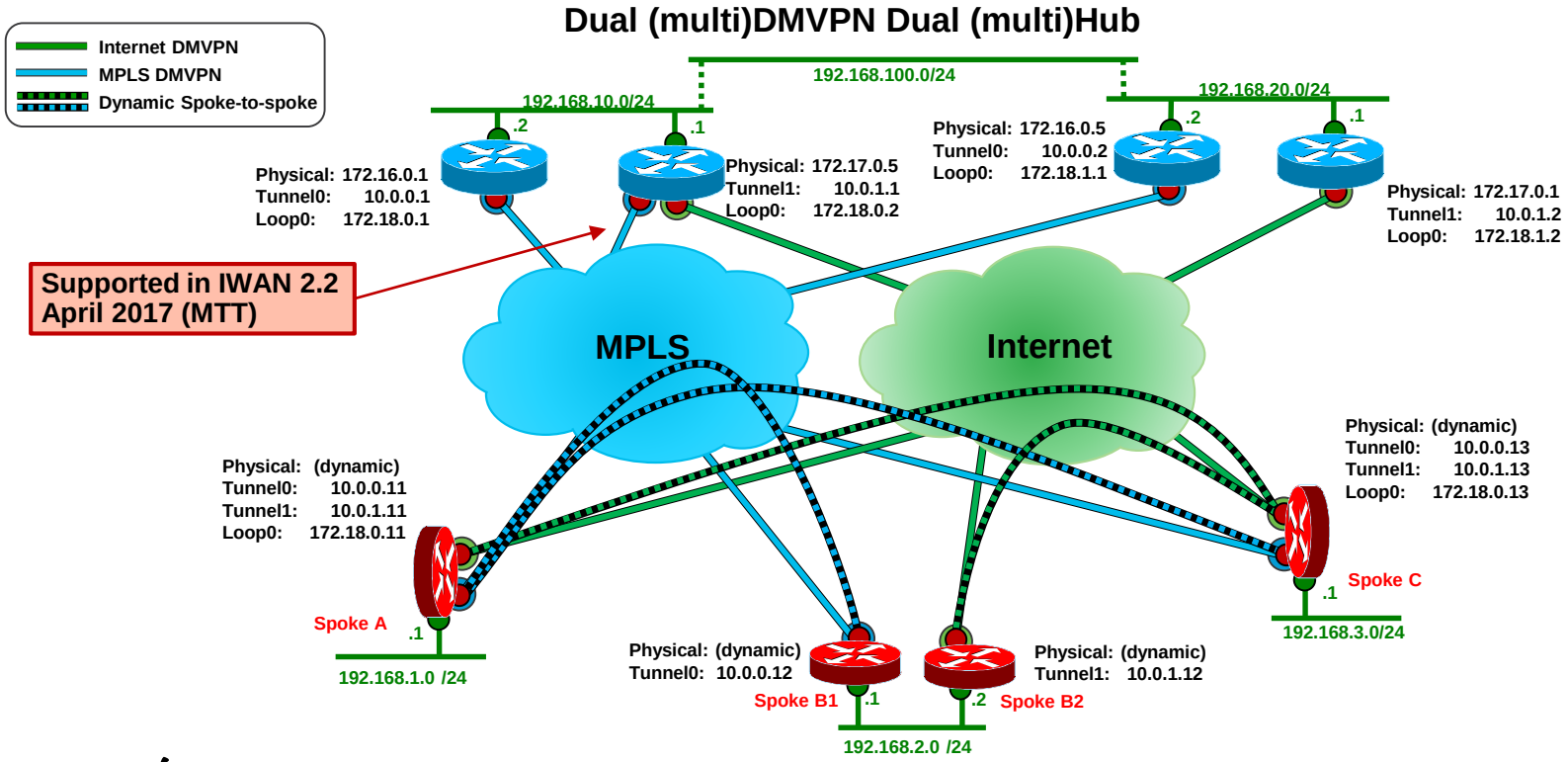


DMVPN designs for IWAN

- Multiple DMVPNs
  - One per physical transport network
  - Path diversity
  - Separate failure domains
- Each Phase 3 DMVPN
  - Single layer hub-and-spoke; hierarchical not currently supported
  - Physical WAN interface in f-VRF
  - Single Hub; Multi-Hub
    - PfRv3 Multi-NH and Multi-DC features
    - MTT (Multiple Tunnel Termination) feature
  - Spoke-Spoke dynamic tunnels
  - Per-Tunnel QOS

- PfRv3 interoperability
  - Dynamic path selection
    - Per application
    - Load Balancing
    - Brownout circumvention
  - Communicates with NHRP via RIB
    - Triggers secondary spoke-spoke tunnels
- Single Overlay Routing Domain
  - Simplified operations and support
  - Simple ECMP load-balancing and primary path provisioning
  - EIGRP or BGP
    - PfRv3 gets secondary path directly from RP

Basic DMVPN Design for IWAN



VPN Selection

Use Case/ Solution	DMVPN (mGRE, p-pGRE)	GETVPN (Tunnel-less)	FlexVPN (dVTI, IKEv2)	SSLVPN (TLS)	Easy VPN (IKEv1)	IPsec VPN (CM, sVTI, p-pGRE)
Remote Access	N-R	N-S	R	R	N-R	N-R
Hub-Spoke (HS)	R	N-S	R Non-Cisco Spoke	N-R	N-R	N-R
HS + Spoke-Spoke	R	R	N-R	N-S	N-S	N-S
IoT	R	N-R	R	R	N-R	N-R
IWAN	R	N-S	N-S	N-S	N-S	N-S
MPLS over xVPN	R MPLS-o-DMVPN	R MPLS-o-mGRE	N-R MPLS-o-Flex	N-S	N-S	N-R MPLS-o-GRE

R = Recommended

N-R = Not Recommended
N-S = Not Supported

DMVPN Details

Agenda

- DMVPN Design Overview
- DMVPN Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- DMVPN Network Segmentation
  - VRF-lite over DMVPN
  - MPLSoDMVPN

NHRP Message Types

- Registration
  - Build base hub-and-spoke network for control and data traffic (Phase 1 and 2 – single layer, Phase 3 – hierarchical)
- Resolution – Phase 2 and 3
  - Get mapping to build dynamic spoke-spoke tunnels
- Traffic Indication (Redirect) – Phase 3
  - Trigger resolution requests at previous GRE tunnel hop
- Purge
  - Clear out stale dynamic NHRP mappings
- Error
  - Signal error conditions

NHRP Main Functionality

- NHRP Registrations – Phase 1, 2 and 3
  - Static NHRP mappings on spokes for Hub (NHS)
  - Spoke (NHC) dynamically registers its VPN to NBMA address mapping with hub (NHS)
- NHRP Resolutions – Phase 2 and 3
  - Dynamically resolve spoke to spoke VPN to NBMA mapping for spoke-spoke tunnels
    - Phase 2 – NHC self triggers to send NHRP Resolution request
    - Phase 3 – NHC triggered by first hop NHS to send NHRP Resolution request
  - NHRP Resolution requests sent via hub-and-spoke or direct spoke-spoke path
  - NHRP Resolution replies sent via direct spoke-spoke path
- NHRP Redirects (Traffic Indication) – Phase 3
  - Data packets forwarded via NHS, which “hairpins” data packets back onto DMVPN
  - NHS sends redirect message to “trigger” NHC to resolve direct spoke-spoke path
  - Check for redirect configuration on egress, send redirect out ingress interface

NHRP Message Extension Types

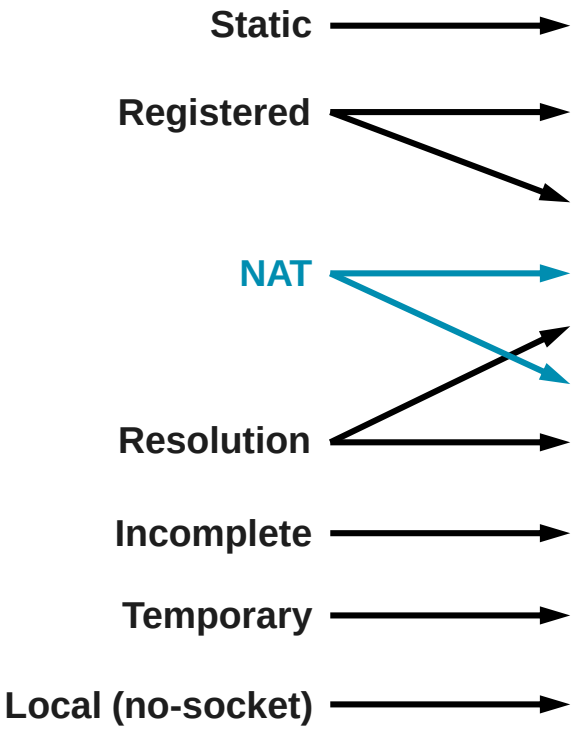
- Responder Address Extension
  - Address mapping for Responding node (Reply messages)
- Forward Transit NHS Record Extension
  - List of NHSs that NHRP request message traversed (loop detection) – copied to reply message
- Reverse Transit NHS Record Extension
  - List of NHSs that NHRP reply message traversed (usually empty – reply over direct tunnel)
- Authentication Extension
  - NHRP Authentication (clear-text)
- NAT Address Extension*
  - Address mapping: For peer (Registration request/reply); For self (Resolution request/reply)
- Cisco Vendor Extension*
  - NHRP Group name; Smart-spoke attributes (name; value);
    MPLS Transport Labels; CMD or NSH header negotiation (PfRv3, TrustSec, ...)

*** Added to NHRP by Cisco**

NHRP Mapping Entries

- **Static**
  - Both host (/32, /128) and network (/<x>) mappings
- **Dynamic**
  - Registered (/32, /128)
    - From NHRP Registration
    - NAT – record both inside and outside address
  - Learned (/32, /128 or /<x>)
    - From NHRP Resolution
    - NAT – record both inside and outside address
- **Incomplete (/32, /128)**
  - Rate-limit NHRP Resolution Requests
  - Data packets process-switched via NHS while building spoke-spoke tunnels. (Phase 2)
- **Temporary (/32) (12.4(22)T)**
  - Same as “Incomplete” mapping except that NBMA is set to Hub
  - Data packets CEF-switched via NHS while building spoke-spoke tunnels. (Phase 2)
- **Local (/32, /128 or /<x>)**
  - Mapping for local network sent in an NHRP Resolution Reply
  - Record which nodes were sent this mapping
- **(no socket)**
  - Not used to forward data packets
  - Do not trigger IPsec encryption
  - Set on Local entries

NHRP Mapping Entries



10.0.0.1/32 via 10.0.0.1, Tunnel0 created 01:20:10, never expire
Type: static, Flags: used
NBMA address: 172.17.0.9

10.0.0.19/32 via 10.0.0.19, Tunnel0 created 01:20:08, expire 00:05:51
Type: dynamic, Flags: unique registered used
NBMA address: 172.16.3.1

10.0.0.18/32 via 10.0.0.18, Tunnel0 created 00:16:09, expire 00:05:50
Type: dynamic, Flags: unique registered used
NBMA address: 172.18.0.2
(Claimed NBMA address: 172.16.2.1)

10.0.0.18/32 via 10.0.0.18, Tunnel0 created 00:09:04, expire 00:00:22
Type: dynamic, Flags: router implicit
NBMA address: 172.18.0.2
(Claimed NBMA address: 172.16.2.1)

192.168.23.0/24 via 10.0.0.19, Tunnel0 created 00:00:11, expire 00:05:48
Type: dynamic, Flags: router used
NBMA address: 172.16.3.1

10.0.0.45/32, Tunnel0 created 00:00:21, expire 00:02:43
Type: incomplete, Flags: negative
Cache hits: 2

10.0.0.17/32 via 10.0.2.17, Tunnel0 created 00:00:09, expire 00:02:55
Type: dynamic, Flags: used temporary
NBMA address: 172.17.0.9

192.168.15.0/24 via 10.0.0.11, Tunnel0 created 00:05:39, expire 00:05:50
Type: dynamic, Flags: router unique local
NBMA address: 172.16.1.1
(no-socket)

NHRP Mapping Flags

unique	**Mapping entry is unique, don't allow overwrite with new NBMA**
registered	**Mapping entry from an NHRP registration**
authoritative	**Mapping entry can be used to answer NHRP resolution requests**
used	**Mapping entry was used in last 60 seconds to forward data traffic**
router	**Mapping entry for remote router**
implicit	**Mapping entry from source information in NHRP resolution request packet**
local	**Mapping entry for a local network, record remote requester**
nat (added 12.4(6)T, hidden 12.4(15)T)	**Remote peer supports the NHRP NAT extension**
rib (12.2(33)XNE, 15.2(1)T)	**Routing Table entry created**
nho (12.2(33)XNE, 15.2(1)T)	**Next-Hop-Override Routing Table entry created**
nhop (15.3(2)S, 15.3(2)T)	**Explicit Next-Hop route out tunnel interface added to RIB/FIB**
nf	**Non-forwarding Entry (No Socket)**

Agenda

- DMVPN Design Overview
- DMVPN Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- DMVPN Network Segmentation
  - VRF-lite over DMVPN
  - MPLSoDMVPN

Phase 1: Hub-and-Spoke – Features

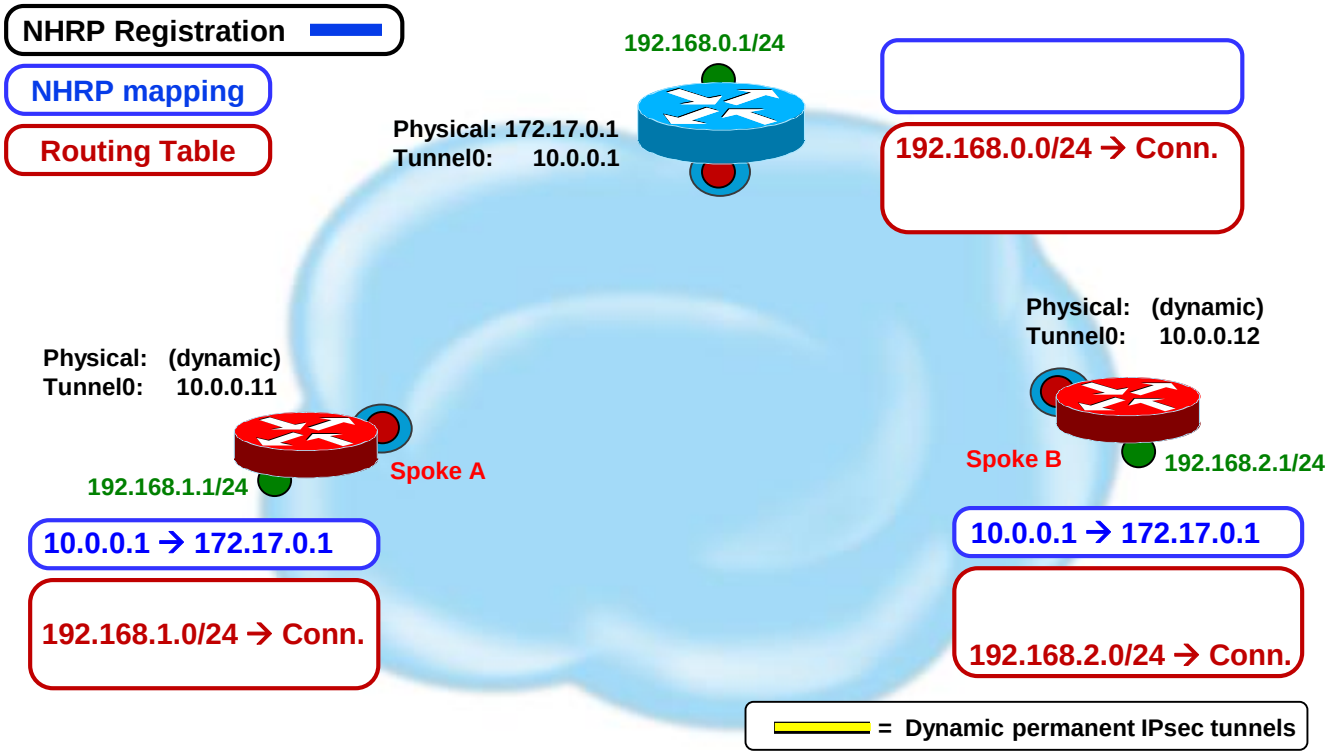
- GRE, NHRP and IPsec configuration
  - p-pGRE or mGRE on spokes; mGRE on hubs
  - ISAKMP/IKEv2 Authentication
    - Certificate (PKI), (Pairwise/Wildcard) Pre-shared Key (PSK)
- NHRP Registration
  - Spoke has static NHRP mapping for Hubs
  - Hub dynamically learns Spoke's NHRP mapping
    - Handles dynamically addressed spokes (DHCP, NAT , …)
  - NAT detection support
    - Check source protocol address in NHRP registration message with source IP on GRE/IP header
      - Same → No NAT; Different → NAT
    - Each spoke must get unique outside NAT IP address
    - Does **not** handle spokes using the same outside NAT IP address (no ports on GRE)
      - Can switch to IPsec tunnel mode, but then lose spoke-spoke tunnel capability

NHRP Registration

- Builds base hub-and-spoke network
  - Hub-and-spoke data traffic
  - Control traffic; NHRP, Routing protocol, IP multicast
  - Phase 2 – Single layer hub-and-spoke
  - Phase 3 – Hierarchical hub-and-spoke (tree).
- Next Hop Client (NHC) has static mapping for Next Hop Servers (NHSs)
- NHC dynamically registers own mapping with NHS
  - Supports spokes with dynamic NBMA addresses or NAT
  - Reports outside address of Hub (if Hub behind NAT)
  - NHRP-group for per-Tunnel QoS (H→S)
  - IPv6: Includes both Unicast-Global and Link-local spoke mappings
- NHS registration reply gives liveliness of NHS
  - Supplies outside NAT address of spoke (if spoke behind NAT)
  - NHRP-group for per-Tunnel QoS (H→S)
  - IPv6: Includes link-local address hub mapping (needed by EIGRP; OSPF)

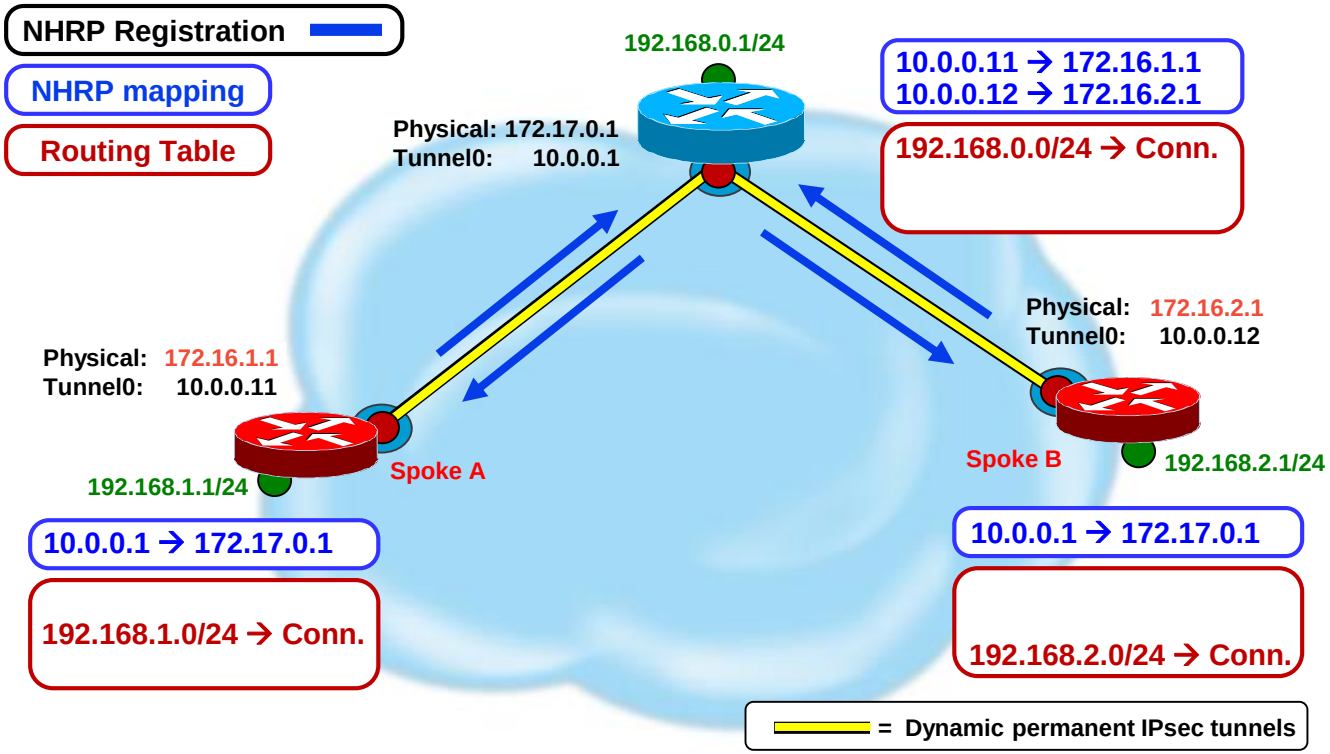
NHRP Registration

Before Building Spoke-Hub Tunnels



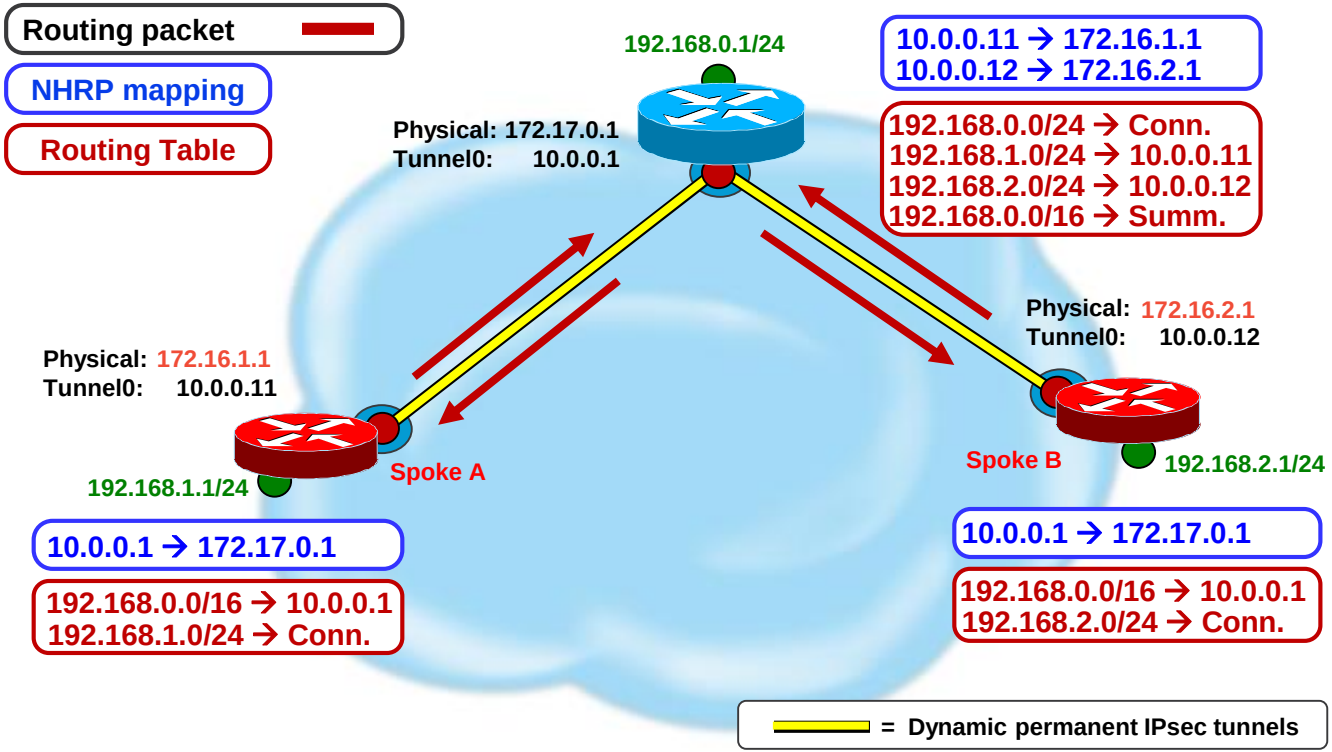
NHRP Registration

Building Spoke-Hub Tunnels



NHRP Registration

Routing Adjacency



Hub-and-Spoke
Data Packet Forwarding

- Process-switching
  - Routing table selects outgoing interface and IP next-hop
  - NHRP looks up packet IP destination to select IP next-hop, overriding IP next-hop from routing table.
    - Could attempt to trigger spoke-spoke tunnel
      - 'tunnel destination …' → Can only send to hub
      - 'ip nhrp server-only' → Don't send NHRP resolution request
  - If no matching NHRP mapping then send to NHS (hub)
- CEF switching
  - IP Next-hop from FIB table (Routing table)
    - IP Next-hop → Hub ➔ data packets send to Hub
  - Adjacency will be complete so CEF switch packet to hub
    - NHRP not involved

Agenda

- DMVPN Design Overview
- DMVPN Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- DMVPN Network Segmentation
  - VRF-lite over DMVPN
  - MPLSoDMVPN

Phase 2: Spoke-Spoke Features

- mGRE tunnel interface per DMVPN cloud
  - On Hubs and Spokes
  - Hubs must be inter-connected in a “Daisy chain” over same mGRE tunnel
  - IKE authentication information (Certificates, Wildcard Pre-shared Keys)
- Spoke-spoke data traffic direct
  - Reduced load on hub
  - Reduced latency
    - Single IPsec encrypt/decrypt
- Routing Protocol
  - Still hub-and-spoke
  - Hub cannot summarize spoke routes
  - Routes on spokes must have IP next-hop of remote spoke (preserve next-hop)

Phase 3: Spoke-Spoke Features

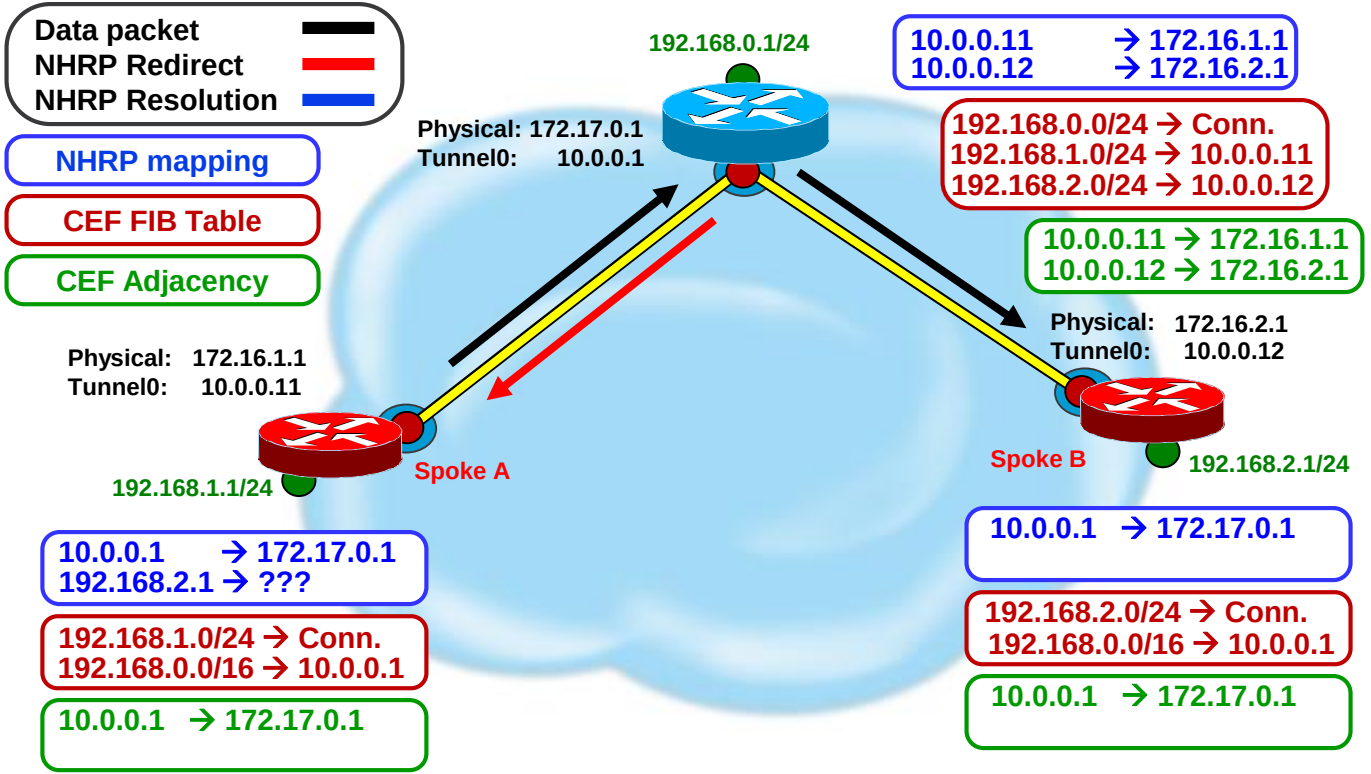
- Increase scale
  - Hierarchical network Layout
    - Increase total number of spokes; same spoke to hub ratio
    - Distribution hubs off load central hub
      - Manage local spoke-spoke tunnels
      - IP multicast and routing protocol
- No hub daisy-chain
  - NHS still interconnected; any pattern
    - Use RIB to forward NHRP packets
    - Reduces RP complexity and load
- OSPF not limited to 2 hubs
  - Network point-multipoint mode
  - Single OSPF area; No summarization

- Spokes don't need full routing tables
  - Can summarize routes at the hub
  - Reduces RIB space and RP load
  - Reduce RP load on hub
    - 1000 spokes, 1 route per spoke;
    - hub advertises 1 route to 1000 spokes → 1000 advertisements
- Phase 2 to Phase 3 migration
  - Build separate Phase 3 DMVPN (can be on same hub and spokes)
  - Migrate spokes one by one from Phase 2 to Phase 3 DMVPN
  - Remove Phase 2 DMVPN

Phase 3 – Building Spoke-spoke Tunnels

- Originating spoke
  - IP Data packet is forwarded out tunnel interface to destination via Hub (NHS)
- Hub (NHS)
  - Receives and forwards data packet on tunnel interfaces with same NHRP Network-id.
  - Check if 'ip nhrp redirect' configured on **outbound** tunnel interface
  - If yes, trigger to send NHRP Redirect message to originating spoke out **inbound** tunnel
- Originating spoke
  - Receives NHRP redirect message
  - Sends NHRP Resolution Request for Data IP packet destination
- Destination spoke
  - Receives NHRP Resolution Request
  - Builds spoke-spoke tunnel
  - Sends NHRP Resolution Reply over spoke-spoke tunnel

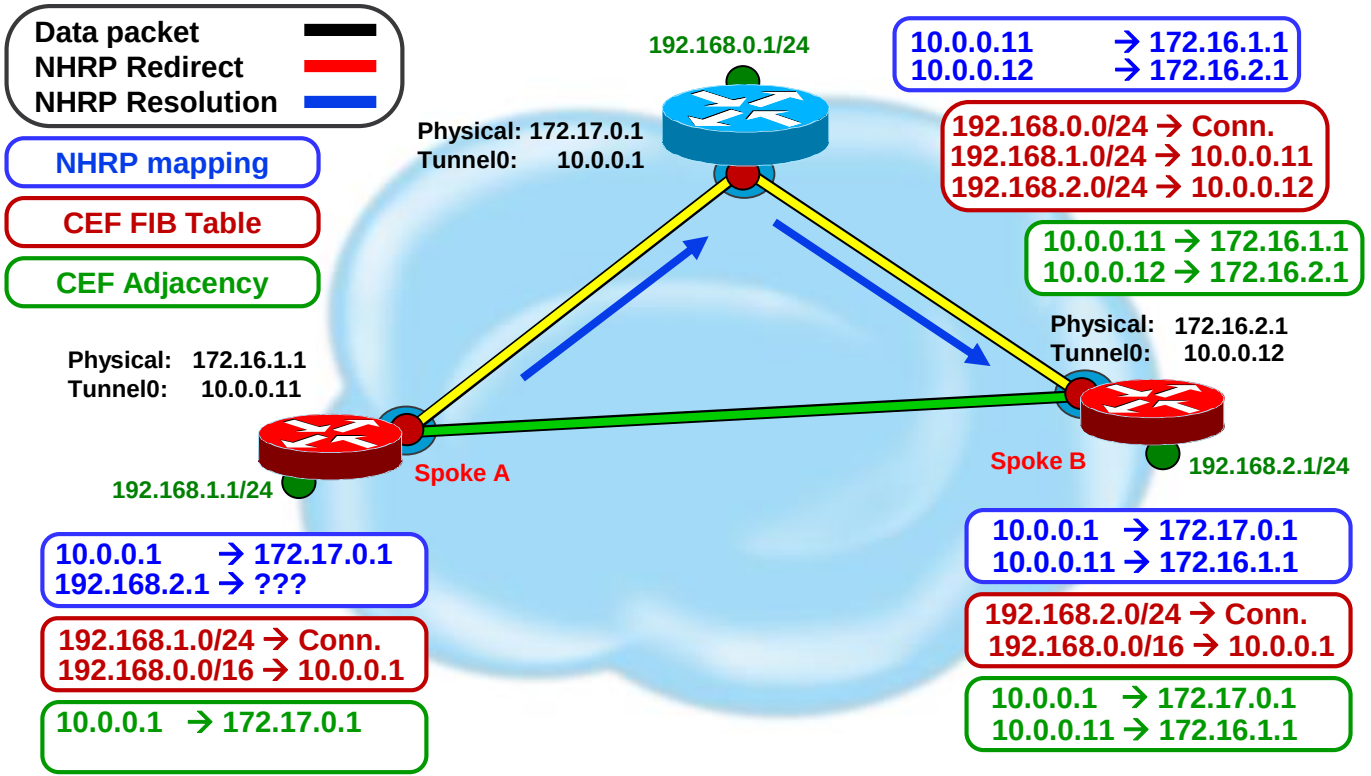
Phase 3 – NHRP Redirects



Phase 3 – NHRP Redirect Processing

- Sender
  - Insert (GRE IP header source, packet destination IP address) in NHRP redirect table – used to rate-limit NHRP redirect messages ‘show ip nhrp redirect’
  - Send NHRP redirect to GRE/IP header source (previous tunnel hop out inbound tunnel)
  - Time out rate-limit entries from the NHRP redirect table
- Receiver
  - Check data IP source address from data IP header in redirect
  - If routing to the IP source is out:
    - A GRE tunnel interface with the same NHRP Network-id
      - then drop redirect
    - Another interface, ‘ip nhrp shortcut’ is configured on inbound tunnel and the IP destination is permitted by ‘ip nhrp interest ACL’ (if configured)
      - then trigger an NHRP resolution request to data IP destination from data IP header in redirect
      - otherwise drop redirect

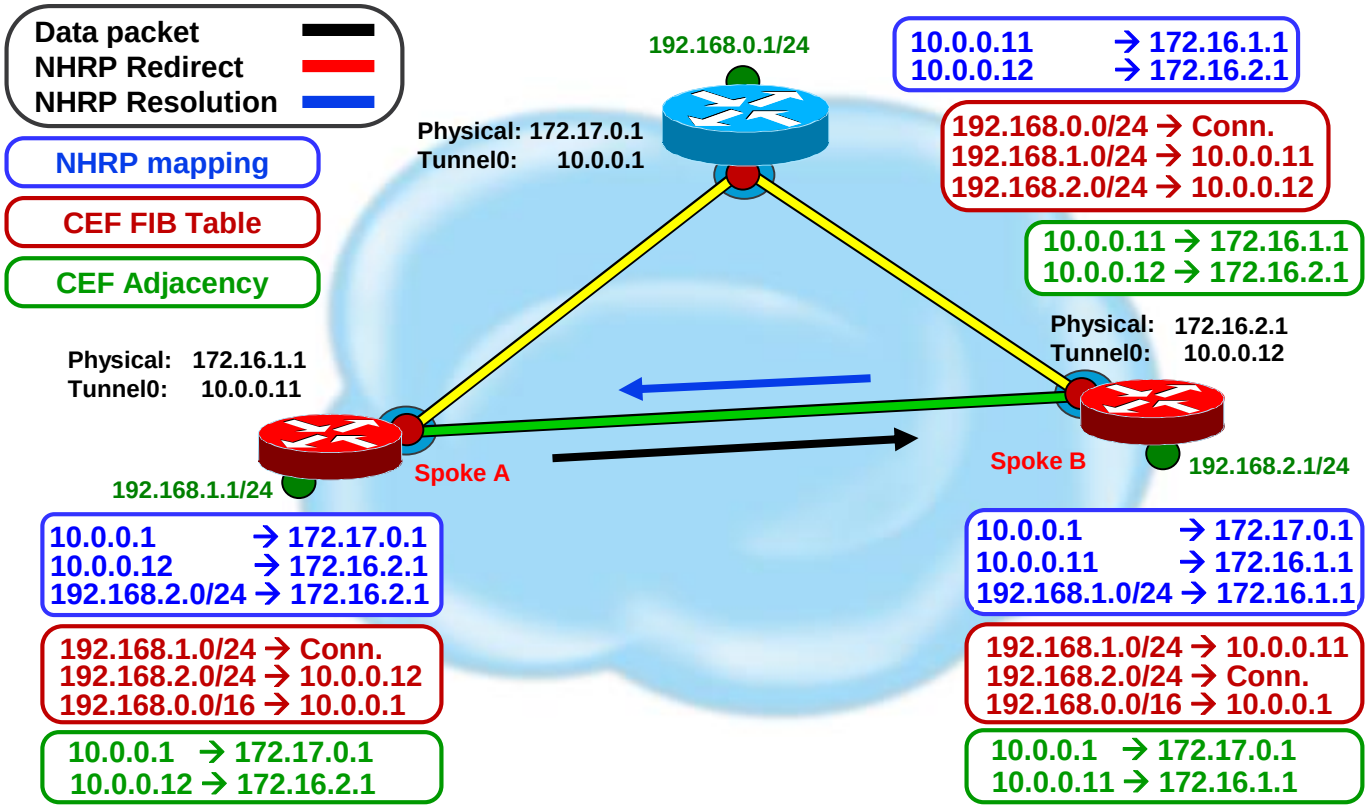
Phase 3 – NHRP Resolution Request



Phase 3 – NHRP Resolution Processing

- Spoke (NHC) routing table has Hub (NHS) as IP next-hop for networks behind remote Spoke
  - If routing table has IP next-hop of remote spoke then process as in Phase 2
- Data packets are forwarded (CEF-switched) via routed path
  - Redirect message sent by every tunnel hop on routed path
  - Redirect for data packet triggers resolution request only on source spoke
- Send resolution request for IP destination from data packet header in redirect
- Resolution requests forwarded via routed path
- Resolution replies forwarded over direct tunnel
  - Direct tunnel initiated from remote → local spoke
- Forward data packets over direct tunnel after receipt of resolution reply.

Phase 3 – NHRP Resolution Reply



Phase 3 – Refresh or Remove Dynamic Mappings

- Dynamic NHRP mapping entries have finite lifetime
  - Controlled by 'ip nhrp holdtime …' on source of mapping (remote spoke)
  - Two types of mapping entries
    - Master entry – Remote Spoke Tunnel IP address
    - Child entries – Remote Network address(es) behind remote-spoke
- Background process checks mapping entries every 60 seconds
  - Master entry: Timing out* → mark CEF adjacency stale
    - If CEF adjacency is then used
      - Refresh Master entry and for each child entry that is also timing out* → queue for immediate refresh
- Refreshing entries
  - Send another Resolution request and reply
    - Resolution request/reply sent over direct tunnel
- If entry expires it is removed
  - If using IPsec and last entry using this NBMA address
    - Trigger IPsec to remove IPsec and ISAKMP/IKEv2 SAs

*** Expire timer < 120 seconds**

NHRP Purge Messages

- Used to clear invalid NHRP mapping information from the network
- NHRP “local”, “(no socket)” mapping entries
  - Created when sending an NHRP resolution reply
  - Copy of mapping information sent in reply
  - Entry tied to corresponding entry in routing table
  - Keeps list of nodes where resolution reply was sent – ‘show ip nhrp detail’
- If routing table changes so that local mapping entry is no longer valid
  - Purge message is sent to each NHRP node in list
  - NHRP nodes clear that mapping from their table
  - Purge messages forwarded over direct tunnel if available, otherwise sent via routed path

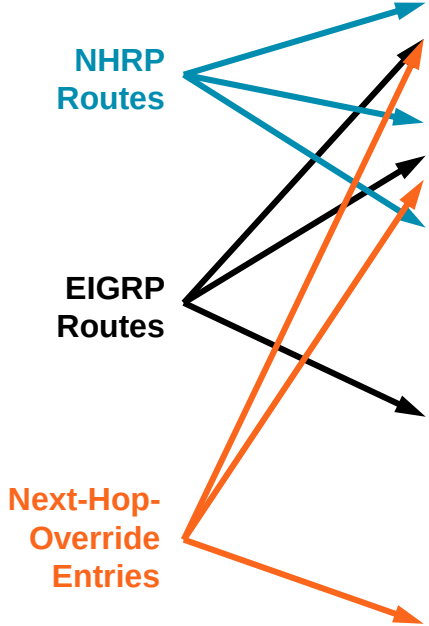
Phase 3 – NHRP and Routing Table

Data Packet Forwarding

- When NHRP resolution is received
  - Insert mapping information in mapping table replacing Incomplete/Temporary mapping
  - Insert NHRP routing entry in Routing Table (RT)
    - NHRP NET/Mask is longer (more specific) than RT Net/Mask
      - Add new route owned by NHRP **(Type = H)**
    - NHRP Net/Mask is equal to RT Net/Mask
      - Add Override Alternate Next-hop **(% flag)**
      - Route still owned by original owner
    - NHRP Net/Mask is shorter (less specific) than RT Net/Mask
      - Increase (make more specific) NHRP mask to = RT Mask
      - Add Override Alternate Next-hop **(% flag)**
      - Route still owned by original owner
  - Insert connected route for tunnel next-hop of NHRP parent mapping (nhop flag)

Phase 3 – NHRP and RT

Routing Table




```
#show ip route
H        192.168.11.0/24 [250/1] via 10.0.1.11, 00:01:02
D   %    192.168.128.0/24 [90/3200000] via 10.0.2.16, 00:50:56, Tunnel0
```

```
#show ip route next-hop-override | section H|%
H        192.168.11.0/24 [250/1] via 10.0.1.11, 00:01:02
D   %    192.168.128.0/24 [90/3200000] via 10.0.2.16, 00:50:56, Tunnel0
              [NHO][90/1] via 10.0.0.1, 00:00:40, Tunnel0
```

```
Routing entry for 192.168.11.0/24
     Known via "nhrp", distance 250, metric 1
     Last update from 10.0.1.11 00:05:29 ago
     Routing Descriptor Blocks:
     *   10.0.1.11, from 10.0.1.11, 00:05:29 ago
           Route metric is 1, traffic share count is 1

Routing entry for 192.168.128.0/24
     Known via "eigrp 1", distance 90, metric 3200000, type internal
     Redistributing via eigrp 1
     Last update from 10.0.2.16 on Tunnel0, 00:43:44 ago
     Routing Descriptor Blocks:
     *   10.0.2.16, from 10.0.2.16, 00:43:44 ago, via Tunnel0
           Route metric is 3200000, traffic share count is 1
     …
         [NHO]10.0.0.1, from 10.0.0.1, 00:05:57 ago, via Tunnel0
           Route metric is 1, traffic share count is 1
     …
```

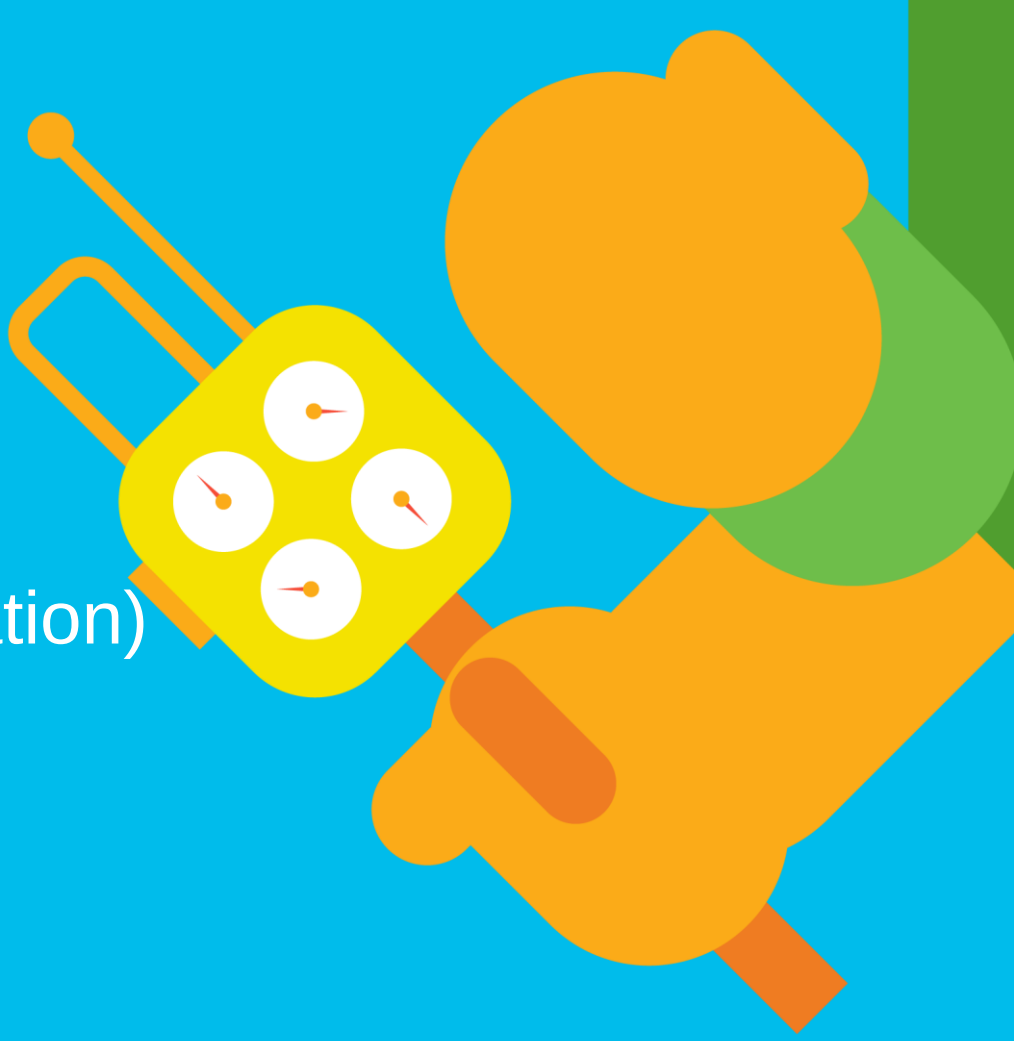
Phase 3 – NHRP and Routing Table

NHRP Parent Route Rules

- Insert NHRP routing entry in Routing Table (RIB)
  - NHRP follows the rules outlined above for inserting RIB routes

  BUT
  - NHRP also makes sure to not contradict routing protocol routes
- Check for “parent” route
  - Parent – next route with mask prefix less than or equal to NHRP route
  - If Parent route via:
    - same tunnel interface → add NHRP route
    - another interface → do not add NHRP route
- After adding NHRP route → Watch Parent route
  - If Parent route changed or removed (attach to next parent route)
  - If new Parent route now via:
    - same tunnel interface → leave NHRP route
    - another interface → remove NHRP route
- Override with ‘no nhrp route-watch’ – can misroute or black-hole traffic

DMVPN MTT
(Multiple Tunnel Termination)



Multiple Tunnel Termination (MTT)

- Issue
  - Multiple DMVPN clouds (IWAN Transports) terminating on the **same Hub**
  - Spoke-spoke tunnels don't always get built
    - Data packets CEF switched between DMVPNs
      - No NHRP Redirect sent → No Spoke-spoke tunnel
    - NHRP Resolution (NHRP) switched between DMVPNs
      - Hub answers NHRP resolution → No Spoke-spoke tunnel
  - Spoke-spoke traffic continues to traverse the hub
- Solution
  - Forward NHRP and Data traffic out the same DMVPN on which it arrived
    - Install regular and secondary routes into RIB
    - Part 1: NHRP traffic; controlled by NHRP control plane
    - Part 2: Data plane traffic forwarding; controlled by FIB/CEF (future)
    - Spoke-spoke tunnel

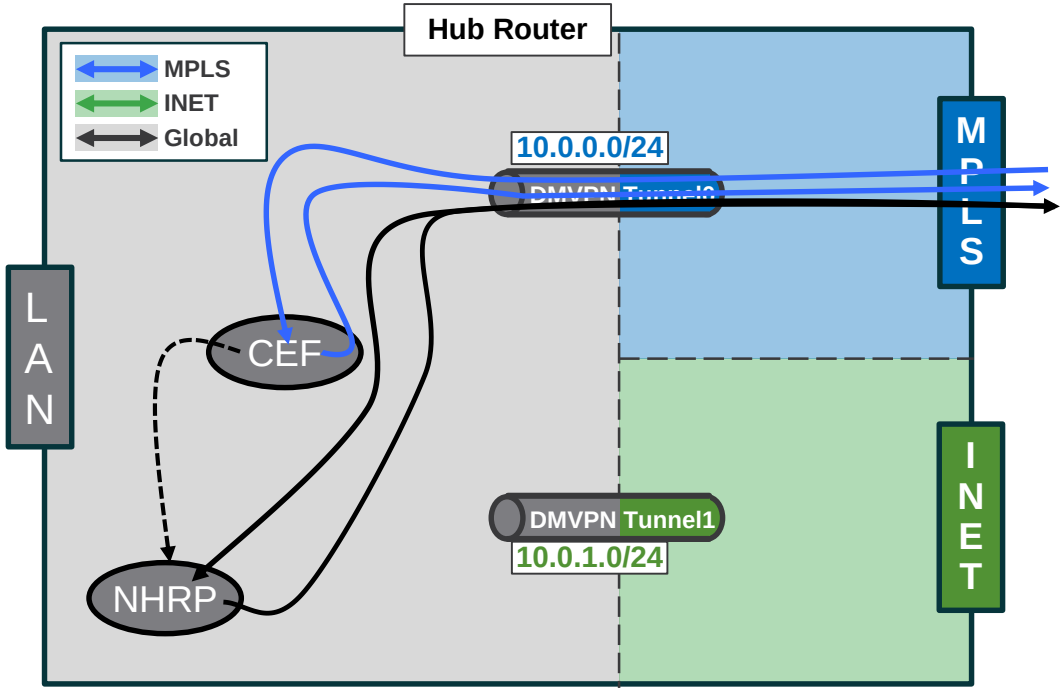
DMVPN without MTT

Routing preferred via MPLS

192.168.1.0/24 (10) → 10.0.0.11

192.168.2.0/24 (10) → 10.0.0.12

- CEF MPLS→MPLS
  - Send NHRP Redirect →MPLS
  - Forward NHRP Resolution Request MPLS→MPLS
  - Spoke-spoke



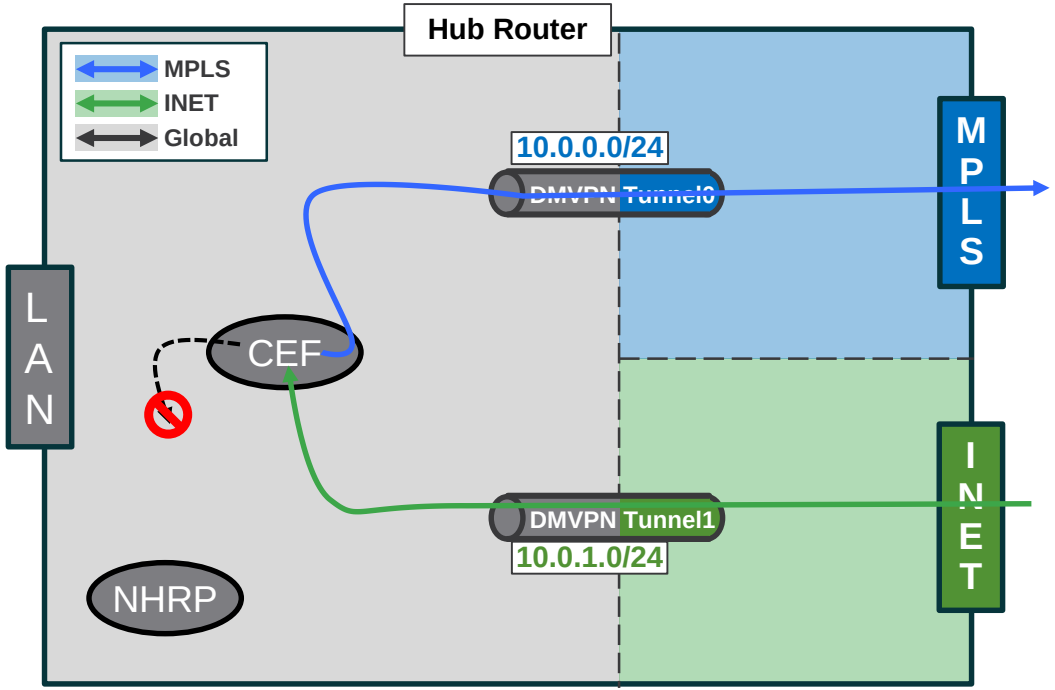
DMVPN without MTT

Routing preferred via MPLS (cont)

192.168.1.0/24 (10) → 10.0.0.11

192.168.2.0/24 (10) → 10.0.0.12

- CEF MPLS→MPLS
  - Send NHRP Redirect →MPLS
  - Forward NHRP Resolution Request MPLS→MPLS
  - Spoke-spoke
- CEF INET→MPLS
  - Don't send NHRP Redirect
  - **No** spoke-spoke

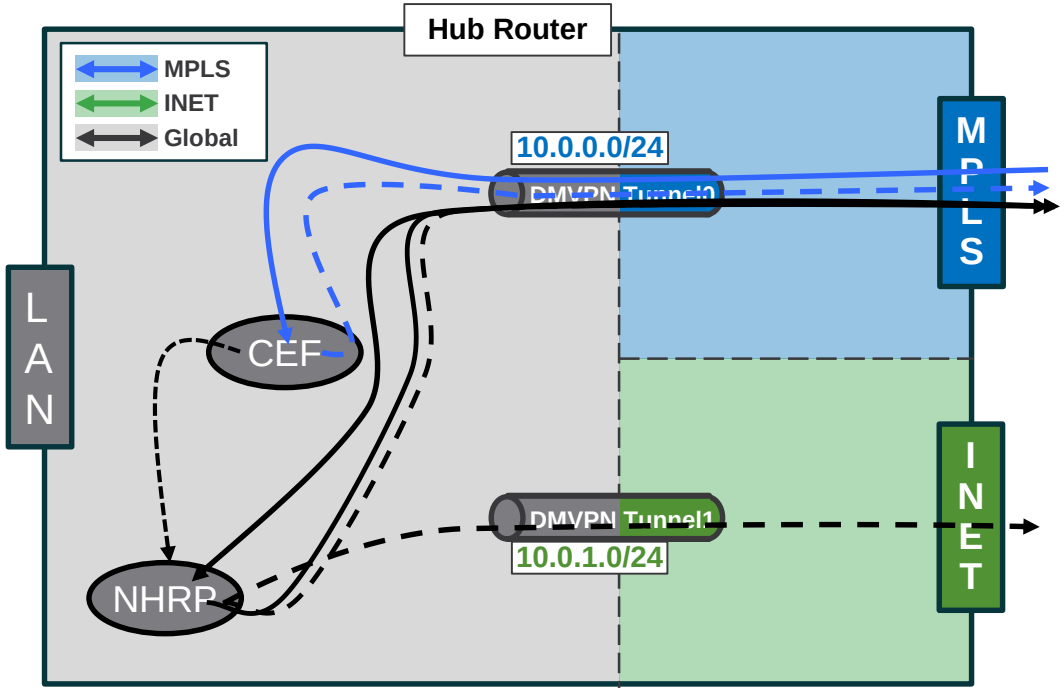


DMVPN without MTT

ECMP routing via MPLS and INET

192.168.1.0/24 (10) → 10.0.0.11
(10) → 10.0.1.11
192.168.2.0/24 (10) → 10.0.0.12
(10) → 10.0.1.12

- CEF inbound MPLS
  - Forward: 50% →MPLS
    - Send NHRP Redirect →MPLS
    - NHRP Resolution Request
      - 50% Forward →MPLS (spoke-spoke)
      - 50% Hub answers (**no** spoke-spoke)

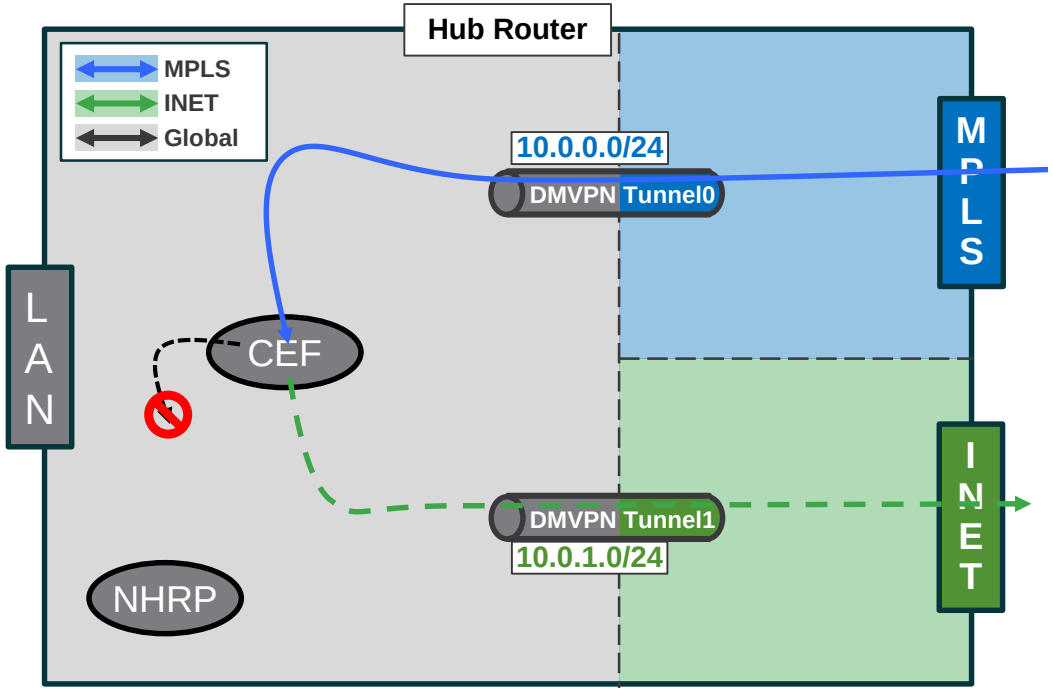


DMVPN without MTT

ECMP routing via MPLS and INET (cont)

192.168.1.0/24 (10) → 10.0.0.11
(10) → 10.0.1.11
192.168.2.0/24 (10) → 10.0.0.12
(10) → 10.0.1.12

- CEF inbound MPLS
  - Forward: 50% →MPLS
    - Send NHRP Redirect →MPLS
    - NHRP Resolution Request
      - 50% Forward →MPLS (spoke-spoke)
      - 50% Hub answers (**no** spoke-spoke)
  - Forward: 50% →INET
    - Don't send NHRP Redirect
    - **No** spoke-spoke
- CEF inbound INET (similar)



Kinds of RIB Paths

- Regular next-hops/paths
  - Most common kind of paths, often equal cost but could be unequal cost.
  - Governed by 'maximum-paths <n>' (up to 32)
  - Installed in the RIB and passed to FIB/CEF for immediate use
- Repair next-hop/path
  - Special paths that are used for IP FRR, BGP PIC, etc.
  - Only ONE repair path (per-prefix) for one or more regular paths.
  - Installed in the RIB and passed to FIB/CEF, but NOT USED as long as one or more regular next hops are active.
- Secondary next-hops/paths
  - Special loop free paths that are typically inferior to regular and repair paths.
  - Governed by 'maximum-secondary-paths <n>' (up to 32; default 0).
  - Installed in RIB but not passed to FIB

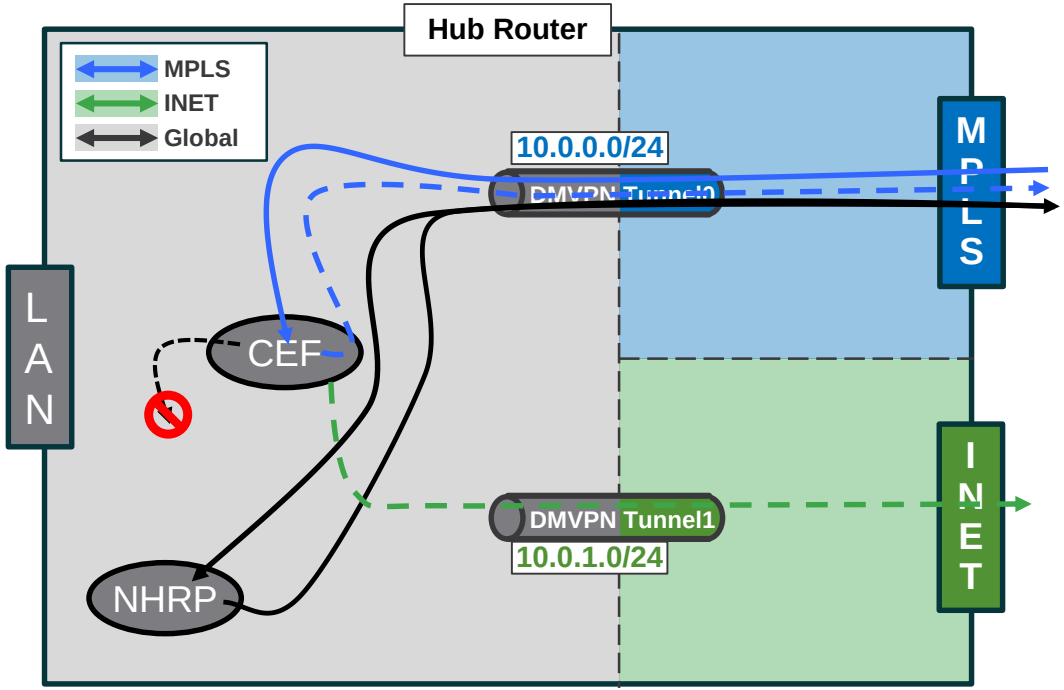
DMVPN with MTT (Part 1)

Routing via MPLS and INET (ECMP or secondary)

16.3.2, 16.4.1, 15.6(3)M2, 15.5(3)S5

192.168.1.0/24 (10) → 10.0.0.11
(10) → 10.0.1.11
192.168.2.0/24 (10) → 10.0.0.12
(10) → 10.0.1.12

- CEF inbound MPLS
  - Forward: 50% →MPLS
    - Send NHRP Redirect →MPLS
    - NHRP Resolution Request
      - 100% Forward →MPLS (spoke-spoke)
  - Forward: 50% →INET
    - Don't send NHRP Redirect
    - **No** spoke-spoke
- CEF inbound INET (similar)



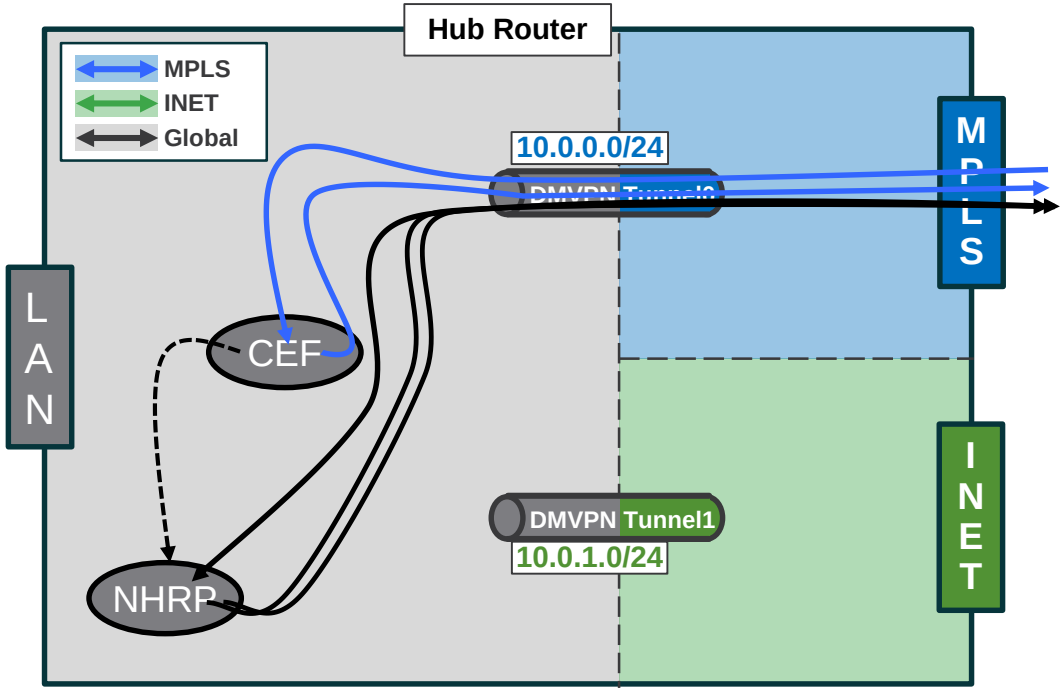
DMVPN with MTT (Part 2)

Routing via MPLS and INET (ECMP or secondary)

On Roadmap

192.168.1.0/24 (10) → 10.0.0.11
[SEC] (20) → 10.0.1.11
192.168.2.0/24 (10) → 10.0.0.12
[SEC] (20) → 10.0.1.12

- CEF inbound MPLS
  - CEF Forward: 100% →MPLS
    - Send NHRP Redirect →MPLS
    - NHRP Resolution Request
      - 100% Forward →MPLS (spoke-spoke)

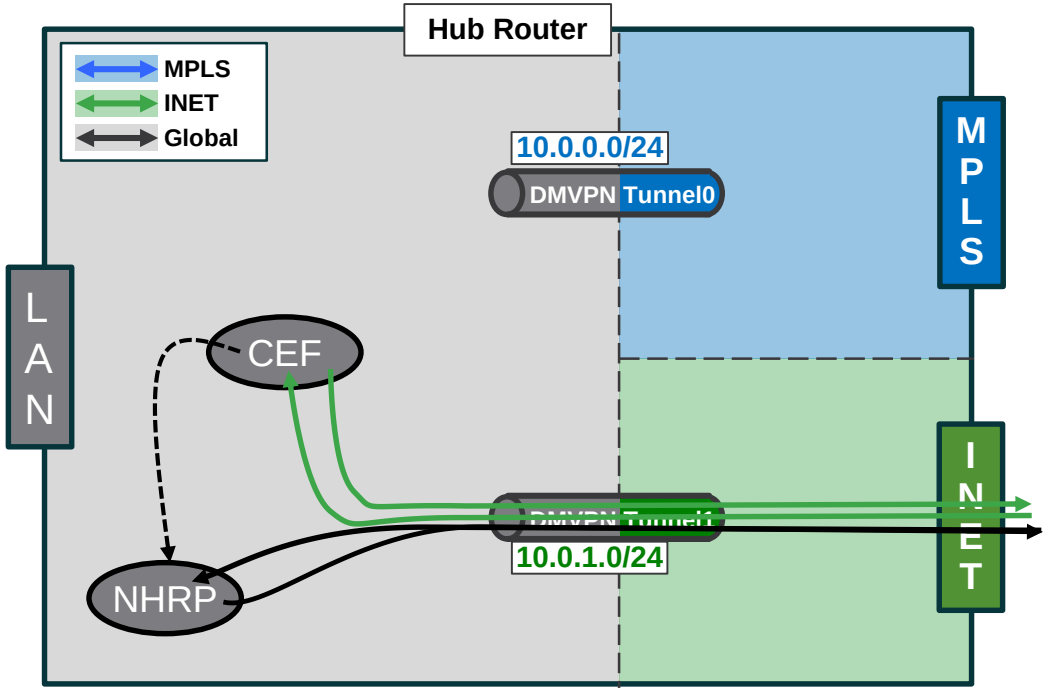


DMVPN with MTT (Part 2)

Routing via MPLS and INET (ECMP or secondary)

192.168.1.0/24 (10) → 10.0.0.11
[SEC] (20) → 10.0.1.11
192.168.2.0/24 (10) → 10.0.0.12
[SEC] (20) → 10.0.1.12

- CEF inbound MPLS
- CEF inbound INET
  - CEF Forward 100% →INET
    - Send NHRP Redirect →INET
  - NHRP Resolution Request
    - 100% Forward →INET (spoke-spoke)



DMVPN with Multiple Tunnel Termination (MTT)

- Configure Routing Protocol to insert secondary routes
  - Mandatory on Hub; Recommended on Spokes (IOS/XE*)
- Part 1:
  - DMVPN – ECMP routes over both tunnels
    - Statistical per flow (src-IP, dst-IP) whether spoke-spoke tunnel is triggered
    - All flows use spoke-spoke tunnel when built
  - IWAN 2.2
    - Data traffic over primary (preferred) route – spoke-spoke triggered by:
      - Data packets for primary tunnel; PfR probes for secondary tunnel
    - Data traffic over secondary (non-preferred) route – spoke-spoke triggered by:
      - PfR probes for both tunnels; up to 10 second delay for secondary tunnel
- Part 2 (future)
  - ECMP or Preferred route over one tunnel (all routes (regular, secondary) in RIB)
  - Data traffic and/or PfR Probes trigger spoke-spoke tunnels over both tunnels

(16.3.2, 16.4.1)

BGP:
maximum-secondary-paths [eibgp|ibgp] <x>

EIGRP:
topology base
maximum-secondary-paths <x>

DMVPN Network Segmentation

Agenda

- DMVPN Design Overview
- DMVPN Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- DMVPN Network Segmentation
  - VRF-lite over DMVPN
  - MPLSoDMVPN

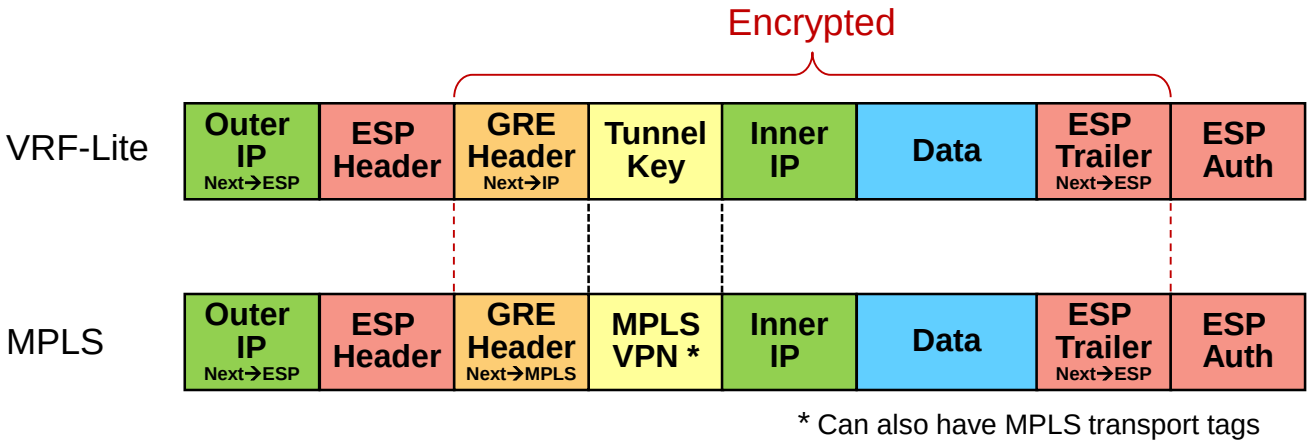
Network Segmentation over DMVPN

Two main techniques

- VRF-lite over DMVPN
  - Separate DMVPN cloud (mGRE tunnel) **per** VRF
  - **Single** IPsec session used by **all** GRE tunnels between same two peers
  - **Separate** Routing Protocol neighbor **per spoke and per VRF**
  - Tunnel Key used to separate packets over DMVPN and transport network
    - Must manually match tunnel keys to VRFs the same way on all nodes
- 2547oDMVPN (MPLS over DMVPN)
  - MPLS VPN running over **single** DMVPN cloud (mGRE tunnel)
  - **Single** IPsec session and **single** GRE tunnel between same two peers
  - **Single** MP-BGP routing neighbor **per spoke** regardless of number of VRFs
  - MPLS VPN tag used to separate packets over the DMVPN and transport network
    - MP-BGP automatically matches VPN tags to VRFs on all nodes

Network Segmentation over DMVPN

- On transport network both techniques look similar
  - Single IPsec session ➔ Cannot differentiate between style being used
  - Encapsulated tunnel packets are similar (effectively MPLS VPN tag == Tunnel Key)



Network Segmentation over DMVPN

- On DMVPN nodes the two techniques present to the router differently
  - VRF-lite
    - Separate mGRE tunnel interface **per VRF** – 'vrf forwarding <vrf>'
    - Tunnels match with per VRF LAN interfaces – 'vrf forwarding <vrf>'
    - **Separate** Routing Protocol neighbor **per spoke and per VRF**
    - Can use MP-BGP on hub to "leak" routes between VRFs for cross VRF forwarding
  - MPLS
    - Single mGRE tunnel interface **for all VRFs** – 'mpls nhrp'
    - MPLS maps per VRF LAN interfaces ('vrf forwarding <vrf>') to/from single tunnel
    - **Single** MP-BGP routing neighbor **per spoke** regardless of number of VRFs
    - MPLS just on the DMVPN or part of a larger MPLS network
      - Hub is an MPLS P/PE
      - Spokes can be an MPLS PE or P/PE
      - VRF RD and RT tags must match on all DMVPN PE routers

Network Segmentation over DMVPN

VRFs on the DMVPN

- VRF definition on the DMVPN nodes
  - Define VRFs used on that node
    - VRF-lite:
      - Defined VRFs will match the configured VRFs on mGRE tunnels and LAN interfaces
    - MPLS:
      - Hubs: Must define all VRFs, even if no local interface uses that VRF*
      - Spokes: Must define all VRFs used at spoke site, even if no local interface uses that VRF*
- Add a new VRF to the network
  - VRF-lite:
    - On Hubs and Spokes – if already defined on Hub only do this on the Spoke
      - Add new mGRE tunnel for VRF with matching Tunnel Key
      - Add Routing Protocol address-family for new VRF
  - MPLS:
    - On Hubs and Spokes – if already defined on Hub only need this on the Spoke
      - Redistribute LAN VRF routes in/out of MP-BGP VRF address family

* **Extending an existing MPLS over DMVPN**

Configuration

Crypto and Physical Interfaces

```
crypto ikev2 keyring DMVPN
   peer DMVPN
      address 0.0.0.0 0.0.0.0
      pre-shared-key cisco123
!
crypto ikev2 profile DMVPN
   match fvrf Outside
   match identity remote address 0.0.0.0
   authentication local pre-share
   authentication remote pre-share
   keyring local DMVPN
!
crypto ipsec transform-set DMVPN esp-aes esp-sha-hmac
   mode transport
!
crypto ipsec profile DMVPN
   set transform-set DMVPN
   set ikev2-profile DMVPN
!
interface Ethernet<y>/0
   vrf forwarding <vrf-name>
   ip address 192.168.<x><y>.1 255.255.255.0
!
interface Ethernet3/0
   vrf forwarding Internet
   ip address 192.168.254.1 255.255.255.0
!
interface Serial<#>/0
   vrf forwarding Outside
   ip address 172.<z>.1 255.255.255.252
!
ip route vrf Outside 0.0.0.0 0.0.0.0 172.<z>.2
```

vrf-name = Yellow, Red, Green

x = Hub (0), Spoke (1,2,3)

y = BU# (Yellow = 0, Red = 2, Green = 4)

z = Hub (17.0), Spoke (16.(1,2,3))

! <inside LAN/VRF interface(s)>

! <Internet access, on hub only>

! <outside (public) interface>

Configuration Block repeated per VRF

Support Internet access for all VRFs

fVRF to separate out transport routing

Configuration

Basic VRF and Tunnel

```
vrf definition <vrf-name>                                  ! <spokes and hub>
   rd <x>:<x>                                              ! <repeated per VRF>
   route-target export <x>:<x>
   route-target import <x>:<x>
!
vrf definition Outside
!
vrf definition Internet                                    ! <hub only>
   rd 10:10
   route-target export 10:10
   route-target import 10:10
   route-target import 1:1
   route-target import 2:2                                 ! <import routes into Internet VRF>
   route-target import 3:3
!
interface Tunnel<y>                                        ! <spokes and hubs>
   bandwidth 1000                                          ! <repeated per VRF for VRF-lite>
   ip address 10.0.<y>.1 255.255.255.0
   ip mtu 1400                                             ! <VRF-lite solution only>
   ip nhrp authentication <vrf-name> or MPLS               ! <MPLS (single instance) for MPLS
   ip nhrp map multicast dynamic                           ! <hub only>
   ip nhrp network-id 10<y>
   ip nhrp holdtime 600
   ip nhrp redirect
   ip tcp adjust-mss 1360                                  ! <VRF-lite solution only>
   tunnel source Serial<#>/0
   tunnel mode gre multipoint
   tunnel key 10<y>
   tunnel vrf Outside
   tunnel protection ipsec profile DMVPN shared
```

x = rd# (Yellow = 1, Red = 2, Green = 3)

vrf-name = Yellow, Red, Green

y = BU# (Yellow = 0, Red = 2, Green = 4)

Configuration
EIGRP and BGP

```
router eigrp 1                                          vrf-name = Yellow,
    no auto-summary                                                Red,
    !                                                              Green
    address-family ipv4 vrf <vrf-name>
        redistribute bgp 1
        network <LAN-network>
        default-metric 1000 100 255 1 1500
        no auto-summary
        autonomous-system 1
    exit-address-family
    !
    ...
!
router bgp 1                                  ! <VRF-lite solution on hub only>
    no synchronization
    bgp log-neighbor-changes
    no auto-summary
    !
    address-family ipv4 vrf [<vrf-name>, Internet]
        redistribute connected
        redistribute eigrp 1
        no synchronization
    exit-address-family
    !
```

Configuration Block repeated per VRF

Configuration Block repeated per VRF

VRF-lite over DMVPN

VRF-lite – Separate DMVPNs

n = number of spokes
v = average number of VRFs per spoke
m = number of hubs for spoke-hub tunnels

- Separate mGRE tunnel per VRF (BU)
  - Dynamic spoke-spoke tunnels per DMVPN (VRF)
  - Hub routers have all VRF (BU) DMVPNs; Spokes only those they need
- Multiple Hub routers for load and redundancy
  - Load:
    - (**m**) **Hubs to support n×v hub-spoke tunnels** (each hub limited by number of RP neighbors)
  - Redundancy:
    - Manually map (2×(**m**) Hubs) or dynamically map ((IOS SLB) (**m**)+1 hubs)
- Routing
  - EIGRP, BGP or OSPF over DMVPN **and** LAN
- Forwarding between VRFs (optional)
  - MP-BGP can be used on hub to leak routes between VRFs
  - Use a router or firewall behind the hub

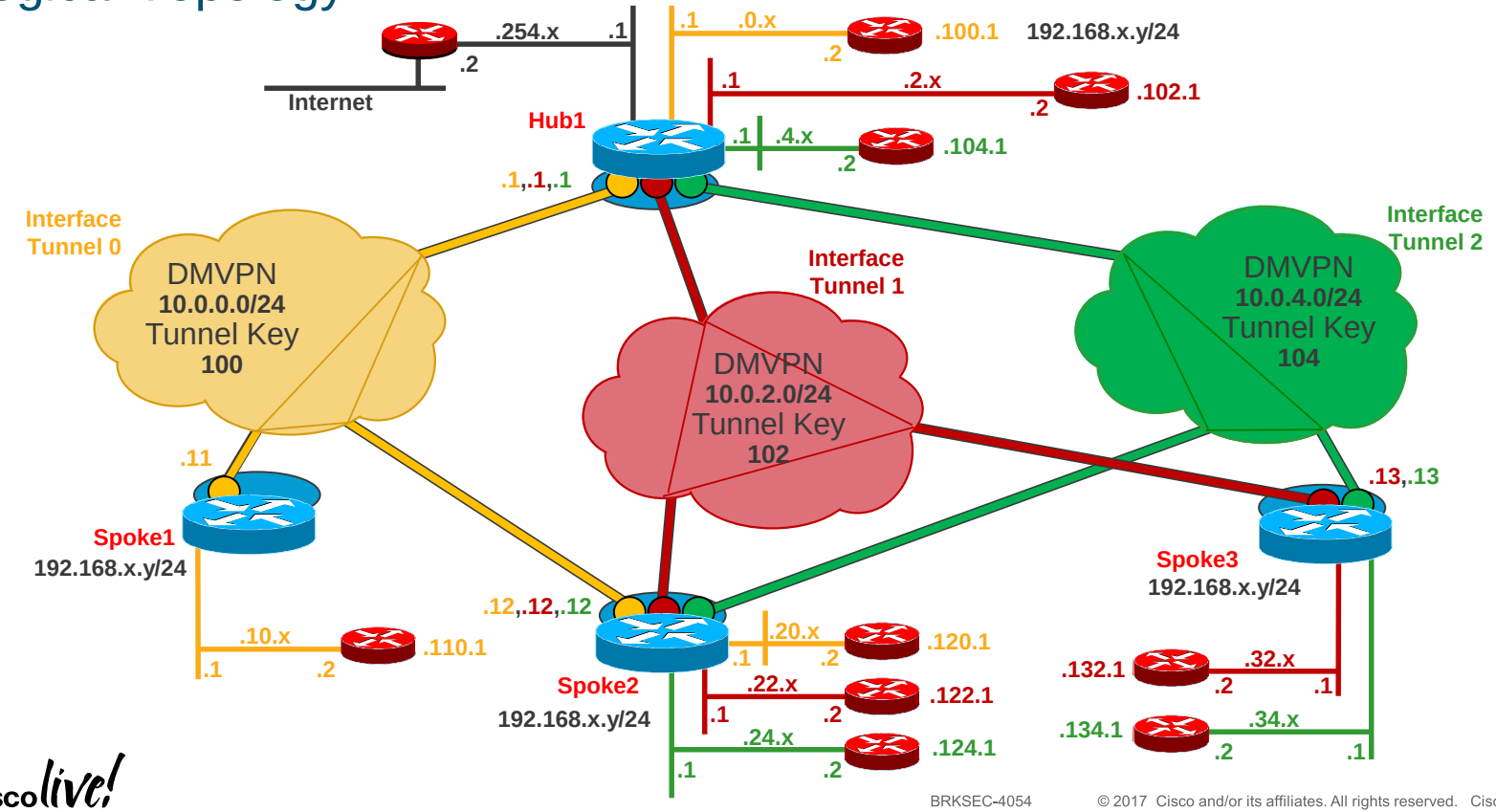
VRF-lite – Separate DMVPNs

Example uses:

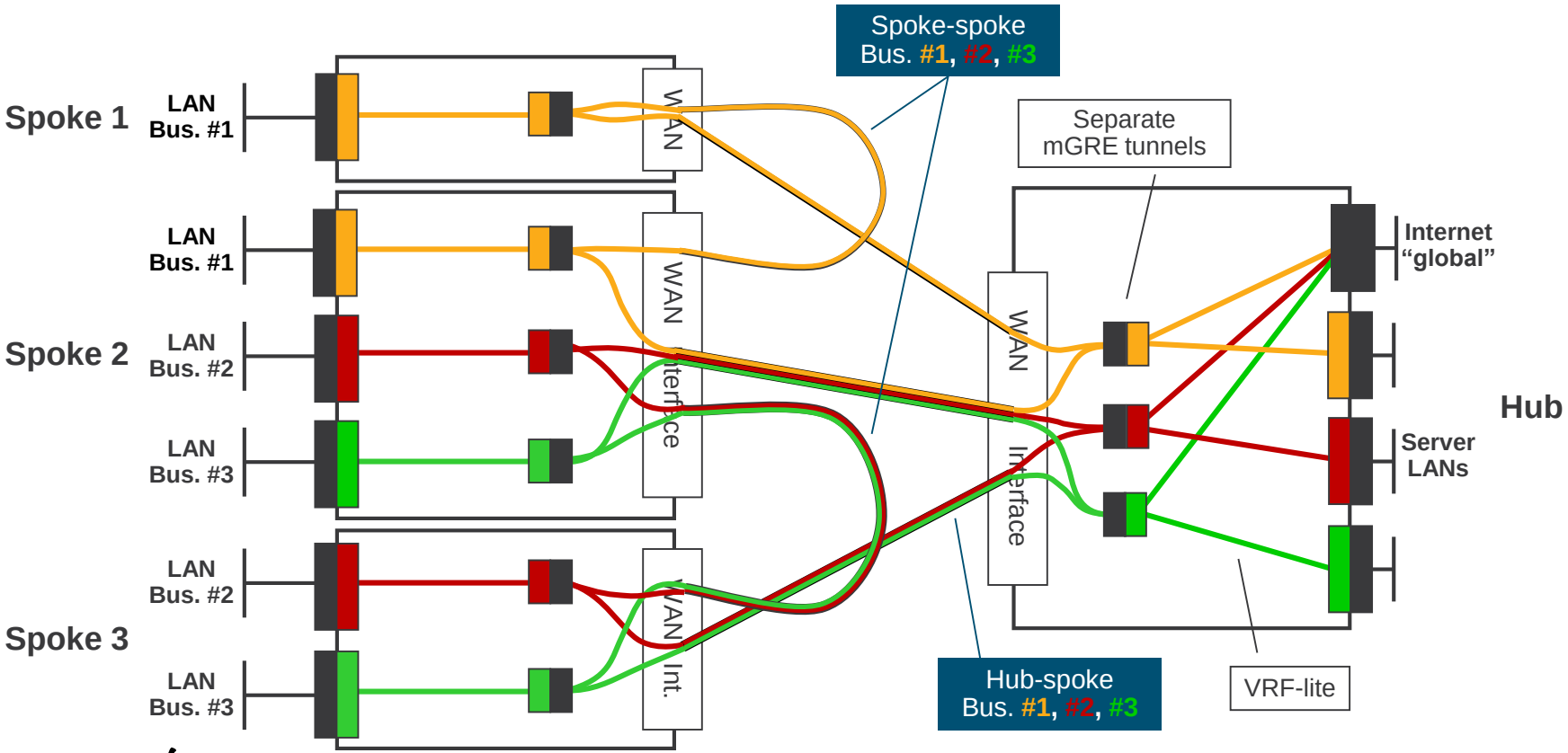
- EIGRP for routing protocol outside of and over DMVPNs
  - An ipv4 address-family per VRF
- MP-BGP only on the hub
  - Import/export routes between VRFs and “Internet” VRF
  - An **ipv4** address-family per VRF
- Internet access via Hub router for all VRFs – no split-tunneling
  - IP addresses spaces unique across all VRFs
  - Routing used to forward return packets from Internet back to correct VRF
- “Outside” f-VRF used for forwarding tunnel packets on WAN (transport)
  - Separation of Transport address space from Overlay (VRF) address space

Separate DMVPNs VRF-lite

Logical Topology

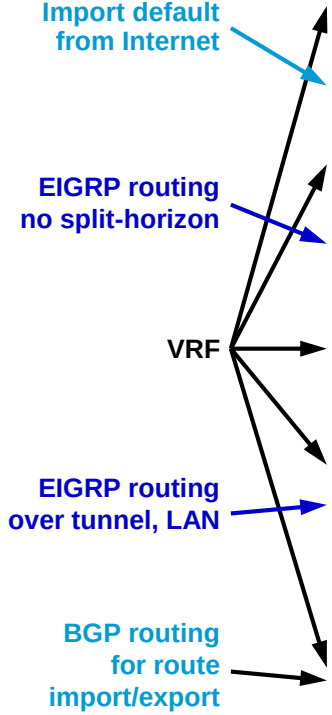


Separate DMVPNs – VRF-lite



Separate DMVPNs – VRF-lite

Hub Configuration – BU VRFs



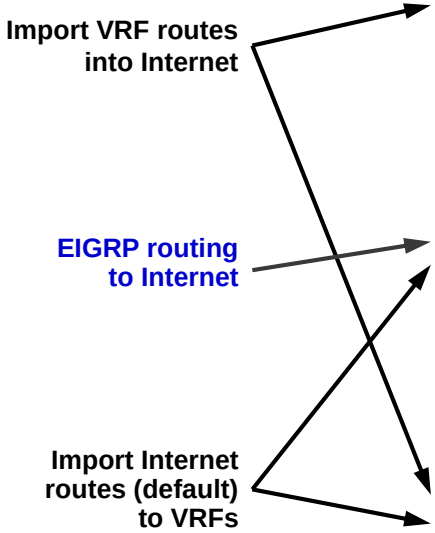

```
vrf definition Yellow
   rd 1:1
   route-target export 1:1
   route-target import 1:1
   route-target import 10:10
!
interface Tunnel0
   vrf forwarding Yellow
   ip address 10.0.0.1 255.255.255.0
   ip nhrp network-id 100
   ip nhrp authentication Yellow
   no ip split-horizon eigrp 1
   tunnel key 100
!
interface Ethernet0/0
   vrf forwarding Yellow
   ip address 192.168.0.1 255.255.255.0
!
router eigrp 1
   !
   address-family ipv4 vrf Yellow
      redistribute bgp 1
      network 10.0.0.0 0.0.0.255
      network 192.168.0.0
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Yellow
      redistribute connected
      redistribute eigrp 1
   exit-address-family
```

```
vrf definition Red
   rd 2:2
   route-target export 2:2
   route-target import 2:2
   route-target import 10:10
!
interface Tunnel2
   vrf forwarding Red
   ip address 10.0.2.1 255.255.255.0
   ip nhrp network-id 102
   ip nhrp authentication Red
   no ip split-horizon eigrp 1
   tunnel key 102
!
interface Ethernet1/0
   vrf forwarding Red
   ip address 192.168.2.1 255.255.255.0
!
router eigrp 1
   !
   address-family ipv4 vrf Red
      redistribute bgp 1
      network 10.0.2.0 0.0.0.255
      network 192.168.2.0
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Red
      redistribute connected
      redistribute eigrp 1
   exit-address-family
```

```
vrf definition Green
   rd 3:3
   route-target export 3:3
   route-target import 3:3
   route-target import 10:10
!
interface Tunnel4
   vrf forwarding Green
   ip address 10.0.4.1 255.255.255.0
   ip nhrp network-id 104
   ip nhrp authentication Green
   no ip split-horizon eigrp 1
   tunnel key 104
!
interface Ethernet2/0
   vrf forwarding Green
   ip address 192.168.4.1 255.255.255.0
!
router eigrp 1
   !
   address-family ipv4 vrf Green
      redistribute bgp 1
      network 10.0.4.0 0.0.0.255
      network 192.168.4.0
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Green
      redistribute connected
      redistribute eigrp 1
   exit-address-family
```

Separate DMVPNs – VRF-lite

Hub Configuration – Internet VRF




```
vrf definition Internet
   rd 10:10
   route-target export 10:10
   route-target import 10:10
   route-target import 1:1
   route-target import 2:2
   route-target import 3:3
!
interface Ethernet3/0
   vrf forwarding Internet
   ip address 192.168.254.1 255.255.255.0
!
router eigrp 1
   !
   address-family ipv4 vrf Internet
      redistribute bgp 1
      network 192.168.254.0
      autonomous-system 1
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Internet
      redistribute connected
      redistribute eigrp 1
      default-information originate
   exit-address-family
```

Separate DMVPNs – VRF-lite

Spoke 2 – Configuration

VRF config

EIGRP routing over DMVPN

No BGP config

```
vrf definition Yellow
   rd 1:1
   route-target export 1:1
   route-target import 1:1
!
interface Tunnel0
   vrf forwarding Yellow
   ip address 10.0.0.12 255.255.255
   ip nhrp authentication Yellow
   ip nhrp network-id 100
   ip nhrp nhs 10.0.0.1 nbma 172.17
   tunnel key 100
!
router eigrp 1
   no auto-summary
   !
   address-family ipv4 vrf Yellow
      network 10.0.0.0 0.0.0.255
      network 192.168.20.0
      autonomous-system 1
   exit-address-family
!
interface Ethernet0/0
   vrf forwarding Yellow
   ip address 192.168.20.1 255.255.
```

```
vrf definition Red
   rd 2:2
   route-target export 2:2
   route-target import 2:2
!
interface Tunnel1
   vrf forwarding Red
   ip address 10.0.2.12 255.255.255
   ip nhrp authentication Red
   ip nhrp network-id 102
   ip nhrp nhs 10.0.2.1 nbma 172.17
   tunnel key 102
!
router eigrp 1
   no auto-summary
   !
   address-family ipv4 vrf Red
      network 10.0.2.0 0.0.0.255
      network 192.168.22.0
      autonomous-system 1
   exit-address-family
!
interface Ethernet1/0
   vrf forwarding Red
   ip address 192.168.22.1 255.255.
```

```
vrf definition Green
   rd 3:3
   route-target export 3:3
   route-target import 3:3
!
interface Tunnel2
   vrf forwarding Green
   ip address 10.0.4.12 255.255.255.0
   ip nhrp authentication Green
   ip nhrp network-id 104
   ip nhrp nhs 10.0.4.1 nbma 172.17.0.1 multicast
   tunnel key 104
!
router eigrp 1
   no auto-summary
   !
   address-family ipv4 vrf Green
      network 10.0.4.0 0.0.0.255
      network 192.168.24.0
      autonomous-system 1
   exit-address-family
!
interface Ethernet2/0
   vrf forwarding Green
   ip address 192.168.24.1 255.255.255.0
```

Separate DMVPNs – VRF-lite

Routing Tables – Hub

vrf Outside

```
S*    0.0.0.0/0 [1/0] via 172.17.0.2
C       172.17.0.0/30 is directly connected, Serial4/0
```

vrf Internet

```
D*EX  0.0.0.0/0 [170/281600] via 192.168.254.2, 3w6d, Ethernet3/0
B       10.0.0.0/24 is directly connected, 3w6d, Tunnel0
B       192.168.0.0/24 is directly connected, 3w6d, Ethernet0/0
B     192.168.10.0/24 [20/3865600] via 10.0.0.11 (Yellow), 3w6d, Tunnel0
B     192.168.20.0/24 [20/3865600] via 10.0.0.12 (Yellow), 3w6d, Tunnel0
B     192.168.100.0/24 [20/307200] via 192.168.0.2 (Yellow), 3w6d, Ethernet0/0
B     192.168.110.0/24 [20/3891200] via 10.0.0.11 (Yellow), 3w6d, Tunnel0
B     192.168.120.0/24 [20/3891200] via 10.0.0.12 (Yellow), 3w6d, Tunnel0
B       10.0.2.0/24 is directly connected, 00:01:22, Tunnel2
B       192.168.2.0/24 is directly connected, 3w6d, Ethernet1/0
B     192.168.22.0/24 [20/3865600] via 10.0.2.12 (Red), 00:01:25, Tunnel2
B     192.168.32.0/24 [20/3865600] via 10.0.2.13 (Red), 00:01:13, Tunnel2
B     192.168.102.0/24 [20/307200] via 192.168.2.2 (Red), 3w6d, Ethernet1/0
B     192.168.122.0/24 [20/3891200] via 10.0.2.12 (Red), 00:01:25, Tunnel2
B     192.168.132.0/24 [20/3891200] via 10.0.2.13 (Red), 00:01:13, Tunnel2
B       10.0.4.0/24 is directly connected, 00:02:05, Tunnel4
B       192.168.4.0/24 is directly connected, 3w6d, Ethernet2/0
B     192.168.24.0/24 [20/3865600] via 10.0.4.12 (Green), 00:01:57, Tunnel4
B     192.168.34.0/24 [20/3865600] via 10.0.4.13 (Green), 00:01:48, Tunnel4
B     192.168.104.0/24 [20/307200] via 192.168.4.2 (Green), 3w6d, Ethernet2/0
B     192.168.124.0/24 [20/3891200] via 10.0.4.12 (Green), 00:01:57, Tunnel4
B     192.168.134.0/24 [20/3891200] via 10.0.4.13 (Green), 00:01:48, Tunnel4
C       192.168.254.0/24 is directly connected, Ethernet3/0
```

Separate DMVPNs – VRF-lite

Routing Tables – Hub (VRFs)

vrf Yellow

```
C      10.0.0.0/24 is directly connected, Tunnel0
C      192.168.0.0/24 is directly connected, Ethernet0/0
D    192.168.100.0/24 [90/307200] via 192.168.0.2, 4w0d, Ethernet0/0
D    192.168.10.0/24 [90/3865600] via 10.0.0.11, 4w0d, Tunnel0
D    192.168.20.0/24 [90/3865600] via 10.0.0.12, 4w0d, Tunnel0
D    192.168.110.0/24 [90/3891200] via 10.0.0.11, 4w0d, Tunnel0
D    192.168.120.0/24 [90/3891200] via 10.0.0.12, 4w0d, Tunnel0
B      192.168.254.0/24 is directly connected, 4w0d, Ethernet3/0
B*   0.0.0.0/0 [20/281600] via 192.168.254.2 (Internet), 4w0d, Ethernet3/0
```

vrf Red

```
C      10.0.2.0/24 is directly connected, Tunnel2
C      192.168.2.0/24 is directly connected, Ethernet1/0
D    192.168.102.0/24 [90/307200] via 192.168.2.2, 00:12:10, Ethernet1/0
D    192.168.22.0/24 [90/3865600] via 10.0.2.12, 00:09:19, Tunnel2
D    192.168.32.0/24 [90/3865600] via 10.0.2.13, 00:09:07, Tunnel2
D    192.168.122.0/24 [90/3891200] via 10.0.2.12, 00:09:19, Tunnel2
D    192.168.132.0/24 [90/3891200] via 10.0.2.13, 00:09:07, Tunnel2
B      192.168.254.0/24 is directly connected, 4w0d, Ethernet3/0
B*   0.0.0.0/0 [20/281600] via 192.168.254.2 (Internet), 4w0d, Ethernet3/0
```

vrf Green

```
C      10.0.4.0/24 is directly connected, Tunnel4
C      192.168.4.0/24 is directly connected, Ethernet2/0
D    192.168.104.0/24 [90/307200] via 192.168.4.2, 00:14:09, Ethernet2/0
D    192.168.24.0/24 [90/3865600] via 10.0.4.12, 00:13:51, Tunnel4
D    192.168.34.0/24 [90/3865600] via 10.0.4.13, 00:13:42, Tunnel4
D    192.168.124.0/24 [90/3891200] via 10.0.4.12, 00:13:51, Tunnel4
D    192.168.134.0/24 [90/3891200] via 10.0.4.13, 00:13:42, Tunnel4
B      192.168.254.0/24 is directly connected, 4w0d, Ethernet3/0
B*   0.0.0.0/0 [20/281600] via 192.168.254.2 (Internet), 4w0d, Ethernet3/0
```

Separate DMVPNs – VRF-lite

Routing Tables – Spoke2

Spoke2: vrf Yellow

```
C       10.0.0.0/24 is directly connected, Tunnel0
C       192.168.20.0/24 is directly connected, Ethernet0/0
D    192.168.120.0/24 [90/307200] via 192.168.20.2, 4w0d, Ethernet0/0
D    192.168.0.0/24 [90/2841600] via 10.0.0.1, 4w0d, Tunnel0
D    192.168.10.0/24 [90/4121600] via 10.0.0.1, 4w0d, Tunnel0
D    192.168.100.0/24 [90/2867200] via 10.0.0.1, 4w0d, Tunnel0
D    192.168.110.0/24 [90/4147200] via 10.0.0.1, 4w0d, Tunnel0
D    192.168.254.0/24 [90/2841600] via 10.0.0.1, 4w0d, Tunnel0
D*EX  0.0.0.0/0 [170/2841600] via 10.0.0.1, 4w0d, Tunnel0
```

vrf Red

```
C       10.0.2.0/24 is directly connected, Tunnel2
C       192.168.22.0/24 is directly connected, Ethernet1/0
D    192.168.122.0/24 [90/307200] via 192.168.22.2, 00:05:22, Ethernet1/0
D    192.168.2.0/24 [90/2841600] via 10.0.2.1, 00:05:22, Tunnel2
D    192.168.32.0/24 [90/4121600] via 10.0.2.1, 00:05:22, Tunnel2
D    192.168.102.0/24 [90/2867200] via 10.0.2.1, 00:05:22, Tunnel2
D    192.168.132.0/24 [90/4147200] via 10.0.2.1, 00:05:22, Tunnel2
D    192.168.254.0/24 [90/2841600] via 10.0.2.1, 00:05:22, Tunnel2
D*EX  0.0.0.0/0 [170/2841600] via 10.0.2.1, 00:05:22, Tunnel2
```

vrf Green

```
C       10.0.4.0/24 is directly connected, Tunnel4
C       192.168.24.0/24 is directly connected, Ethernet2/0
D    192.168.124.0/24 [90/307200] via 192.168.24.2, 00:01:41, Ethernet2/0
D    192.168.4.0/24 [90/2841600] via 10.0.4.1, 00:01:41, Tunnel4
D    192.168.34.0/24 [90/4121600] via 10.0.4.1, 00:01:41, Tunnel4
D    192.168.104.0/24 [90/2867200] via 10.0.4.1, 00:01:41, Tunnel4
D    192.168.134.0/24 [90/4147200] via 10.0.4.1, 00:01:41, Tunnel4
D    192.168.254.0/24 [90/2841600] via 10.0.4.1, 00:01:41, Tunnel4
D*EX  0.0.0.0/0 [170/2841600] via 10.0.4.1, 00:01:41, Tunnel4
```

Separate DMVPNs – VRF-lite

Routing Tables – Spoke 1 and 3

Spoke1: vrf Yellow

```
C       10.0.0.0/24 is directly connected, Tunnel0
D    192.168.0.0/24 [90/2841600] via 10.0.0.1, 4w0d, Tunnel0
C       192.168.10.0/24 is directly connected, Ethernet0/0
D    192.168.20.0/24 [90/4121600] via 10.0.0.1, 4w0d, Tunnel0
D    192.168.100.0/24 [90/2867200] via 10.0.0.1, 4w0d, Tunnel0
D    192.168.110.0/24 [90/307200] via 192.168.10.2, 4w0d, Ethernet0/0
D    192.168.120.0/24 [90/4147200] via 10.0.0.1, 4w0d, Tunnel0
D    192.168.254.0/24 [90/2841600] via 10.0.0.1, 4w0d, Tunnel0
D*EX  0.0.0.0/0 [170/2841600] via 10.0.0.1, 4w0d, Tunnel0
```

Spoke3: vrf Red

```
C       10.0.2.0/24 is directly connected, Tunnel2
D    192.168.2.0/24 [90/2841600] via 10.0.2.1, 00:01:33, Tunnel2
D    192.168.22.0/24 [90/4121600] via 10.0.2.1, 00:01:33, Tunnel2
C       192.168.32.0/24 is directly connected, Ethernet1/0
D    192.168.102.0/24 [90/2867200] via 10.0.2.1, 00:01:33, Tunnel2
D    192.168.122.0/24 [90/4147200] via 10.0.2.1, 00:01:33, Tunnel2
D    192.168.132.0/24 [90/307200] via 192.168.32.2, 00:01:33, Ethernet1/0
D    192.168.254.0/24 [90/2841600] via 10.0.2.1, 00:01:33, Tunnel2
D*EX  0.0.0.0/0 [170/2841600] via 10.0.2.1, 00:01:33, Tunnel2
```

vrf Green

```
C       10.0.4.0/24 is directly connected, Tunnel4
D    192.168.4.0/24 [90/2841600] via 10.0.4.1, 00:02:15, Tunnel4
D    192.168.24.0/24 [90/4121600] via 10.0.4.1, 00:02:15, Tunnel4
C       192.168.34.0/24 is directly connected, Ethernet2/0
D    192.168.104.0/24 [90/2867200] via 10.0.4.1, 00:02:15, Tunnel4
D    192.168.124.0/24 [90/4147200] via 10.0.4.1, 00:02:15, Tunnel4
D    192.168.134.0/24 [90/307200] via 192.168.34.2, 00:02:15, Ethernet2/0
D    192.168.254.0/24 [90/2841600] via 10.0.4.1, 00:02:15, Tunnel4
D*EX  0.0.0.0/0 [170/2841600] via 10.0.4.1, 00:02:15, Tunnel4
```

Separate DMVPNs – VRF-lite

Summary

- Separate DMVPN mGRE tunnel per BU VRF
- Hub routers handle all DMVPNs
  - Multiple Hub routers for redundancy and load
- EIGRP used for routing protocol outside of and over DMVPNs on Spokes and Hubs
  - Address family per VRF
- BGP used only on the hub
  - Redistribute between EIGRP and BGP for import/export of routes between VRFs
  - “Internet” VRF for Internet access and routing between VRFs
- “Outside” VRF for routing DMVPN tunnel packets

MPLS over DMVPN

MPLS over DMVPN

n = number of spokes
m = number of hubs for spoke-hub tunnels

- Single DMVPN mGRE tunnel on all routers
  - Dynamic spoke-spoke tunnels support all common VRFs between spokes
  - Hub routers support all VRFs (BU); Spokes only those they need
- Multiple Hub routers for load and redundancy
  - Load:
    - (**m**) **Hubs to support n hub-spoke tunnels** (each hub limited by number of MP-BGP neighbors)
  - Redundancy:
    - Manually map (2×(**m**) Hubs) or dynamically ((IOS SLB) map (**m**)+1 hubs)
- Routing
  - MP-BGP over DMVPN; EIGRP, BGP or OSPF on LANs
- Forwarding between VRFs (optional)
  - MP-BGP can be used on hub to leak routes between VRFs
  - Use a router or firewall behind the hub

MPLS over DMVPN

Example uses:

- EIGRP for routing protocol on LANs (outside of DMVPN)
  - An ipv4 address-family per VRF
- MP-BGP over DMVPN
  - Transport all VRF routes over a single MP-BGP neighborship
  - Import/export routes between VRFs and “Internet” VRF
  - An **ipv4** address-family per VRF plus one **vpnv4** address-family
- Internet access via Hub router for all VRFs – no split-tunneling
  - IP address spaces are unique across all VRFs
  - Routing used to forward return packets from Internet back to correct VRF
- “Outside” f-VRF used for forwarding tunnel packets on WAN (transport)
  - Separation of Transport address space from Overlay (VRF) address space

MPLS over DMVPN

- DMVPN Phase 1 – hub-and-spoke only
  - LDP (mpls ip) is used for MPLS tag distribution only on hub-spoke tunnels
  - Hub is configured as MPLS P router
  - Spoke to spoke packets are MPLS tag-switched via Hub
- DMVPN Phase 2 – spoke-spoke only after shortcut tunnel is up
  - LDP (mpls ip) is used for MPLS tag distribution only on hub-spoke tunnels
  - Hub is configured as MPLS PE router
  - Spoke to spoke packets:
    - **Cannot be** MPLS tag-switched via Hub → dropped
    - **Can be** MPLS tag-switched through dynamic spoke-spoke tunnel once it is up
- DMVPN Phase 3 – full spoke-spoke support (15.4(1)S, 15.4(2)T)
  - NHRP (mpls nhrp) is used for MPLS tag distribution on hub-spoke and spoke-spoke tunnels
  - Hub is configured as MPLS P/PE router
  - Spoke to spoke packets are MPLS tag-switched via Hub **then** through spoke-spoke tunnel once it is up

MPLS over DMVPN Phase 3

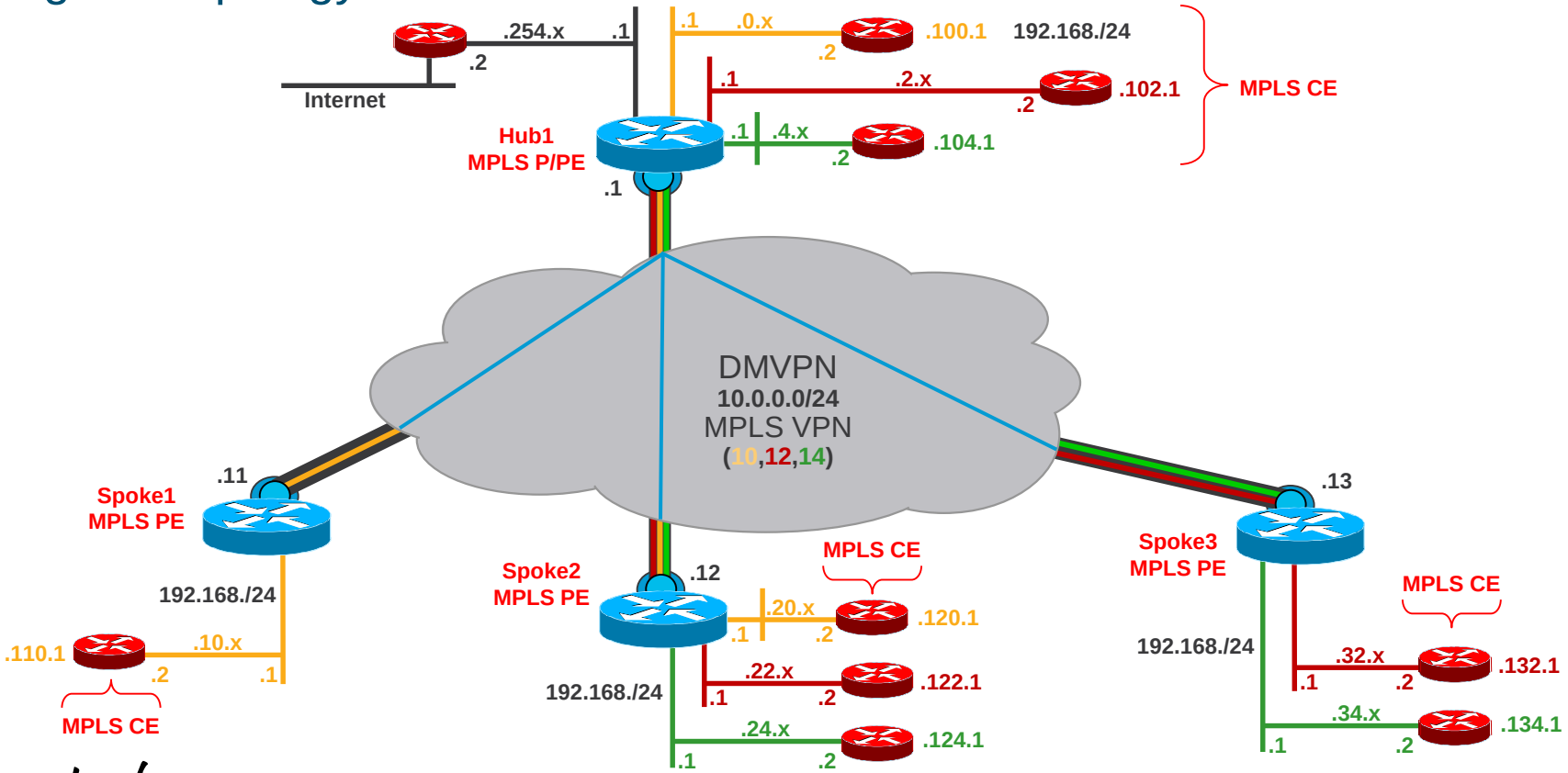
- New support in NHRP to
  - Keep track of NHRP mapping table entries per VRF
  - Transport MPLS forwarding labels
    - MPLS LDP **not** used over DMVPN
    - MP-BGP propagates VPN labels
    - VRF RD and RT tags must be the same on all nodes.
- New CLI
  - 'mpls nhrp' replaces 'mpls ip' on the tunnel interface
    - Tag switching on spoke-hub-spoke and spoke-spoke path
    - Hub router is MPLS P/PE
  - 'mpls mtu ...' applied before MPLS "encapsulation"
    - Apply 'ip tcp adjust-mss ...' on any IP interface in path
  - Per-tunnel QoS
    - MPLS experimental bits (15.5(3)M,S)

```
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.1 255.255.255.0
 no ip redirects
 ip nhrp authentication test
 ip nhrp map multicast dynamic
 ip nhrp network-id 100000
 ip nhrp holdtime 360
 ip nhrp redirect
 mpls mtu 1400
 mpls nhrp
 tunnel source Serial2/0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
```

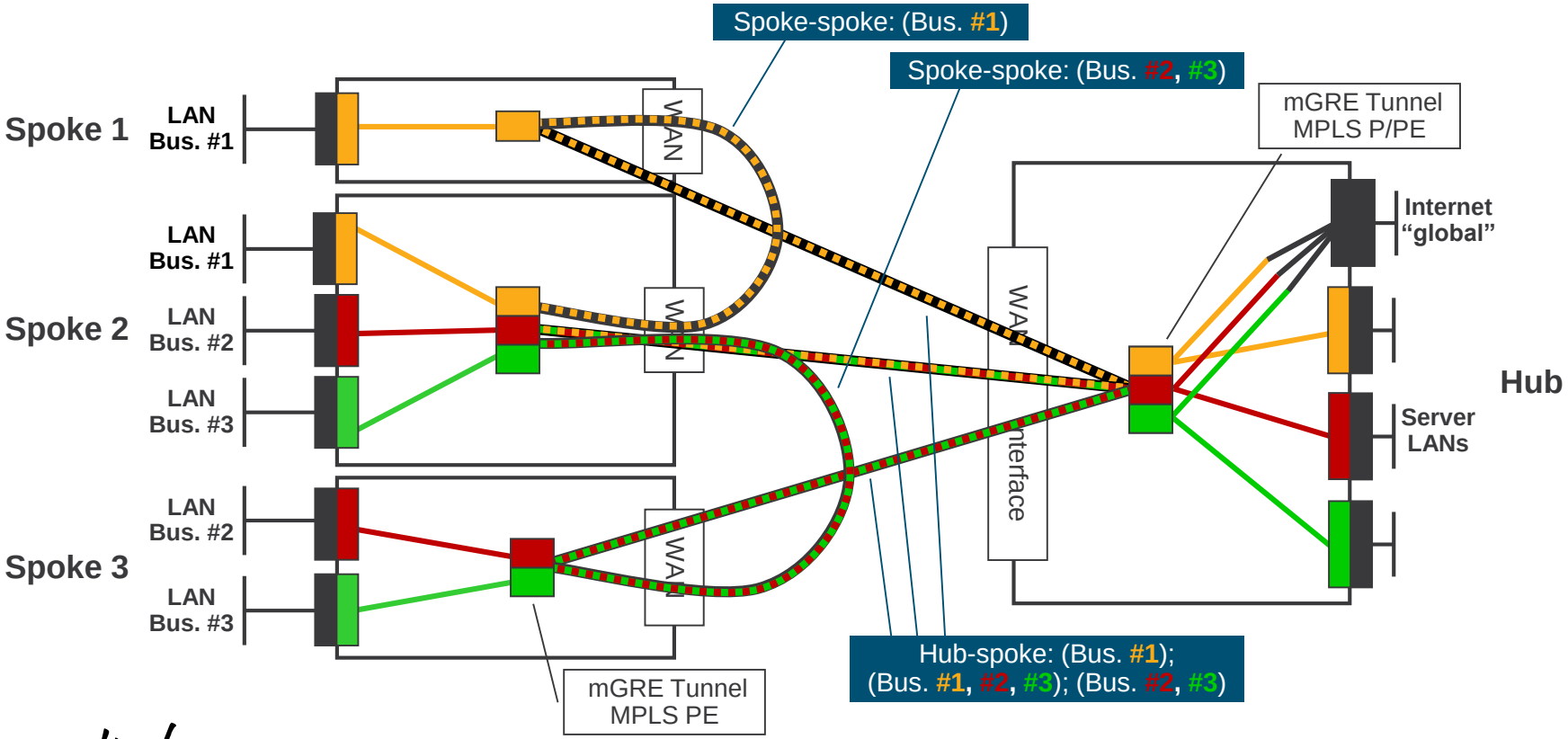
15.5(3)M,S

MPLS over DMVPN – 2547oDMVPN

Logical Topology

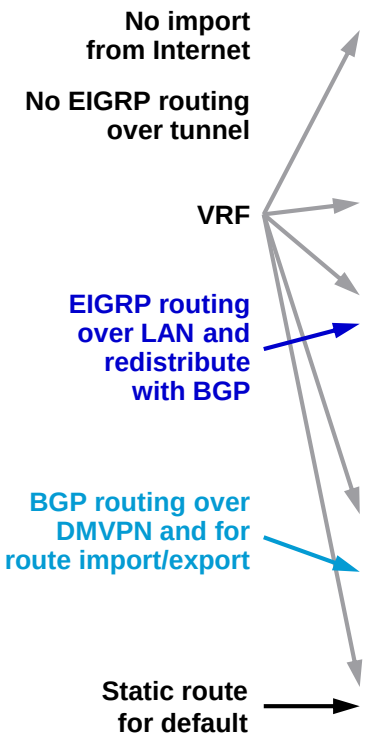


MPLS over DMVPN – 2547oDMVPN



MPLS over DMVPN – 2547oDMVPN

Hub Configuration – BU VRFs



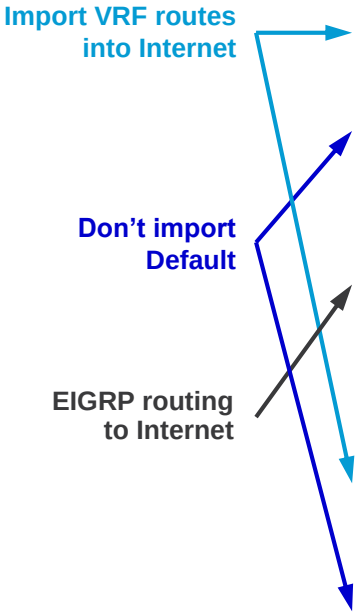

```
vrf definition Yellow
   rd 1:1
   route-target export 1:1
   route-target import 1:1
!
interface Ethernet0/0
   vrf forwarding Yellow
   ip address 192.168.0.1 255.255.255.0
   ip tcp adjust-mss 1360
!
router eigrp 1
   !
   address-family ipv4 vrf Yellow
      default-metric 1000 100 255 1 1500
      redistribute bgp 1
      network 192.168.0.0
      autonomous-system 1
   exit-address-family
   !
router bgp 1
   !
   address-family ipv4 vrf Yellow
      redistribute connected
      redistribute static
      redistribute eigrp 1
      default-information originate
   exit-address-family
!
ip route vrf Yellow 0.0.0.0 0.0.0.0 –
                    Ethernet3/0 192.168.254.2
```

```
vrf definition Red
   rd 2:2
   route-target export 2:2
   route-target import 2:2
!
interface Ethernet1/0
   vrf forwarding Red
   ip address 192.168.2.1 255.255.255.0
   ip tcp adjust-mss 1360
!
router eigrp 1
   !
   address-family ipv4 vrf Red
      default-metric 1000 100 255 1 1500
      redistribute bgp 1
      network 192.168.2.0
      autonomous-system 1
   exit-address-family
   !
router bgp 1
   !
   address-family ipv4 vrf Red
      redistribute connected
      redistribute static
      redistribute eigrp 1
      default-information originate
   exit-address-family
!
ip route vrf Red 0.0.0.0 0.0.0.0 –
                 Ethernet3/0 192.168.254.2
```

```
vrf definition Green
   rd 3:3
   route-target export 3:3
   route-target import 3:3
!
interface Ethernet2/0
   vrf forwarding Green
   ip address 192.168.4.1 255.255.255.0
   ip tcp adjust-mss 1360
!
router eigrp 1
   !
   address-family ipv4 vrf Green
      default-metric 1000 100 255 1 1500
      redistribute bgp 1
      network 192.168.4.0
      autonomous-system 1
   exit-address-family
   !
router bgp 1
   !
   address-family ipv4 vrf Green
      redistribute connected
      redistribute static
      redistribute eigrp 1
      default-information originate
   exit-address-family
!
ip route vrf Green 0.0.0.0 0.0.0.0 –
                   Ethernet3/0 192.168.254.2
```

MPLS over DMVPN – 2547oDMVPN

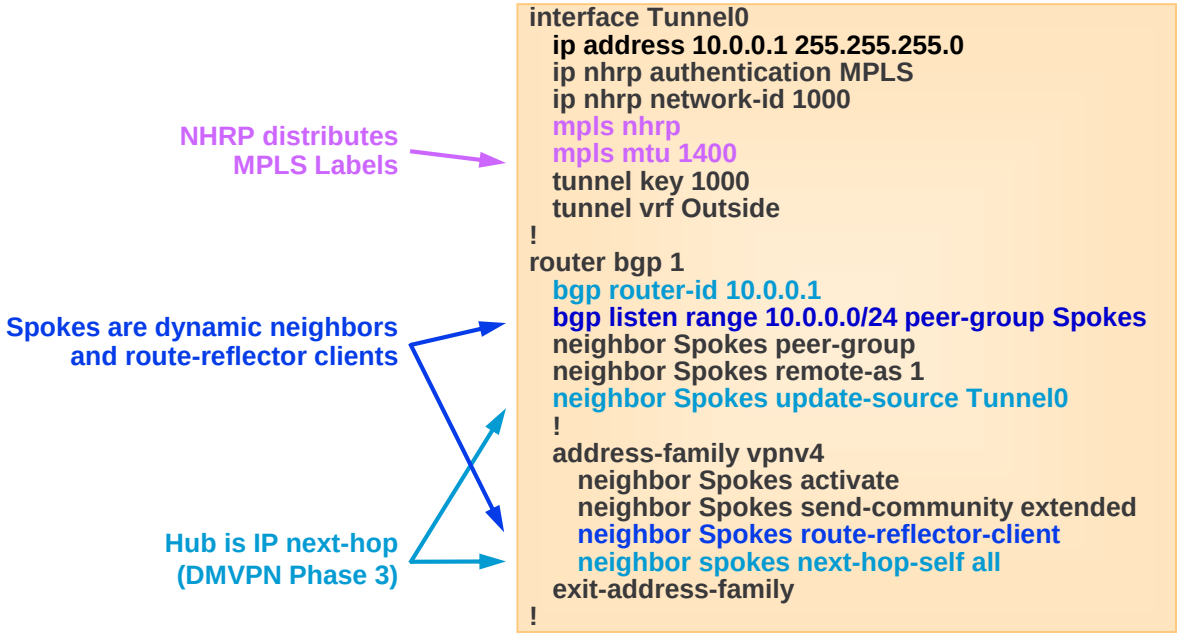
Hub Configuration – Internet VRF




```
vrf definition Internet
   rd 10:10
   route-target export 10:10
   route-target import 10:10
   route-target import 1:1
   route-target import 2:2
   route-target import 3:3
   !
   address-family ipv4
      import map No-Default
   exit address-family
!
interface Ethernet3/0
   vrf forwarding Internet
   ip address 192.168.254.1 255.255.255.0
!
router eigrp 1
   !
   address-family ipv4 vrf Internet
      default-metric 1000 100 255 1 1500
      redistribute bgp 1
      network 192.168.254.0
      autonomous-system 1
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Internet
      network 192.168.254.0
      redistribute eigrp 1
   exit-address-family
!
access-list 20 deny host 0.0.0.0
access-list 20 permit any
!
route-map No-Default permit 10
   match ip address 20
```

MPLS over DMVPN – 2547oDMVPN

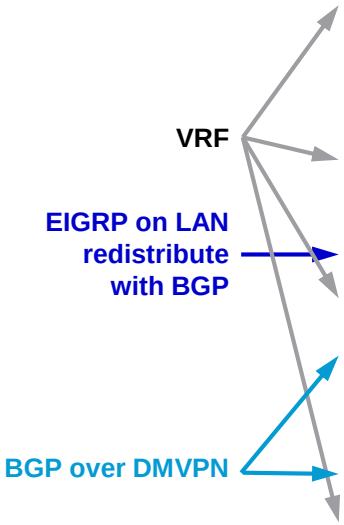
Hub Configuration – MP-BGP over DMVPN




```
interface Tunnel0
   ip address 10.0.0.1 255.255.255.0
   ip nhrp authentication MPLS
   ip nhrp network-id 1000
   mpls nhrp
   mpls mtu 1400
   tunnel key 1000
   tunnel vrf Outside
!
router bgp 1
   bgp router-id 10.0.0.1
   bgp listen range 10.0.0.0/24 peer-group Spokes
   neighbor Spokes peer-group
   neighbor Spokes remote-as 1
   neighbor Spokes update-source Tunnel0
   !
   address-family vpnv4
      neighbor Spokes activate
      neighbor Spokes send-community extended
      neighbor Spokes route-reflector-client
      neighbor spokes next-hop-self all
   exit-address-family
!
```

MPLS over DMVPN – 2547oDMVPN

Spoke2 Configuration – BU VRFs




```
vrf definition Yellow
   rd 1:1
   route-target export 1:1
   route-target import 1:1
!
interface Ethernet0/0
   vrf forwarding Yellow
   ip address 192.168.20.1 255.255.25
   ip tcp adjust-mss 1360
!
router eigrp 1
   !
   address-family ipv4 vrf Yellow
      default-metric 1000 100 255 1 15(
      redistribute bgp 1
      network 192.168.20.0
      autonomous-system 1
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Yellow
      redistribute eigrp 1
   exit-address-family
```

```
vrf definition Red
   rd 2:2
   route-target export 2:2
   route-target import 2:2
!
interface Ethernet1/0
   vrf forwarding Red
   ip address 192.168.22.1 255.255.25
   ip tcp adjust-mss 1360
!
router eigrp 1
   !
   address-family ipv4 vrf Red
      default-metric 1000 100 255 1 15(
      redistribute bgp 1
      network 192.168.22.0
      autonomous-system 1
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Red
      redistribute eigrp 1
   exit-address-family
```

```
vrf definition Green
   rd 3:3
   route-target export 3:3
   route-target import 3:3
!
interface Ethernet2/0
   vrf forwarding Green
   ip address 192.168.24.1 255.255.255.0
   ip tcp adjust-mss 1360
!
router eigrp 1
   !
   address-family ipv4 vrf Green
      default-metric 1000 100 255 1 1500
      redistribute bgp 1
      network 192.168.24.0
      autonomous-system 1
   exit-address-family
!
router bgp 1
   !
   address-family ipv4 vrf Green
      redistribute eigrp 1
   exit-address-family
```

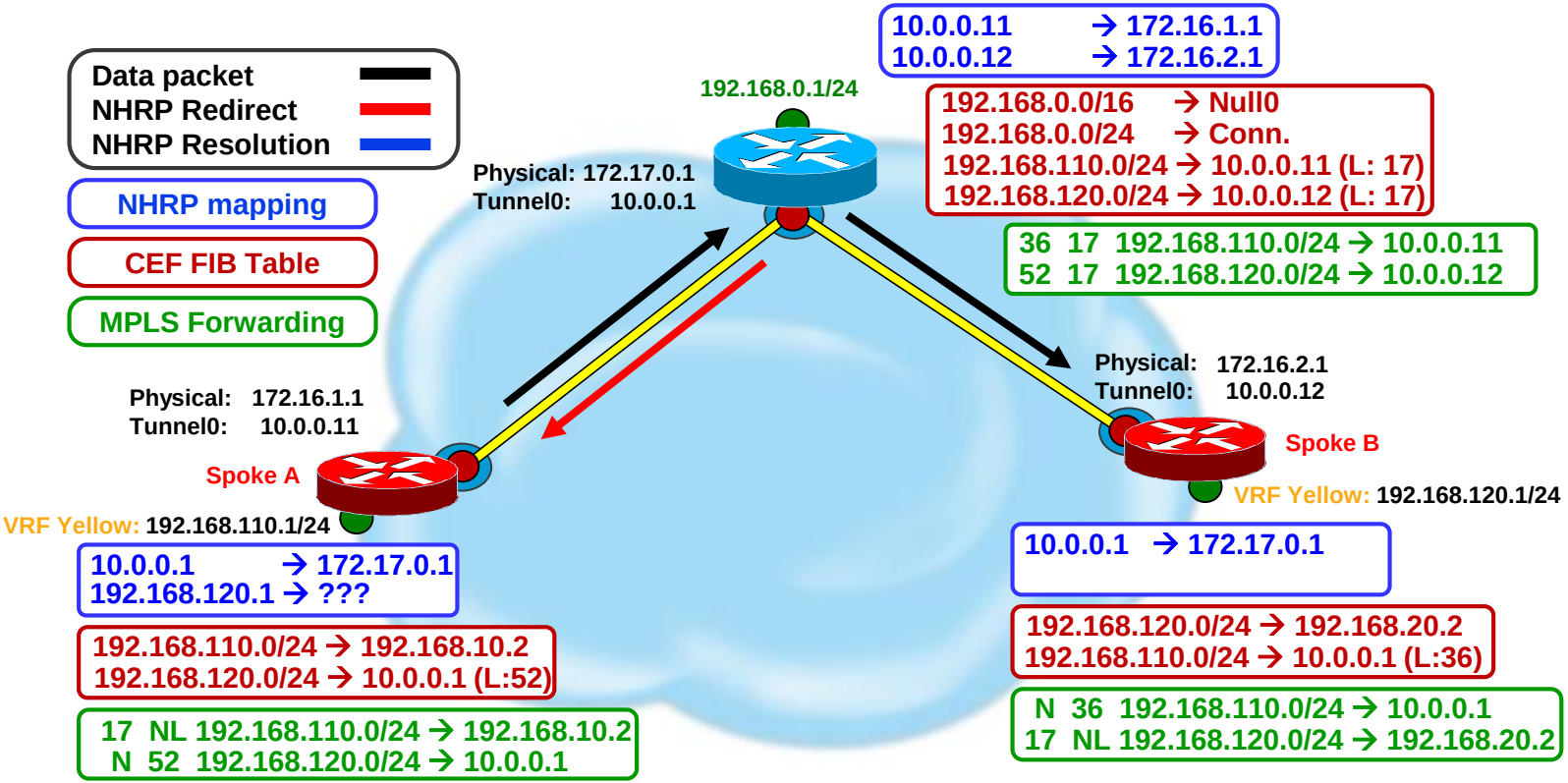
MPLS over DMVPN – 2547oDMVPN

Spoke2 Configuration – MP-BGP over DMVPN


```
interface Tunnel0
  ip address 10.0.0.12 255.255.255.0
  ip nhrp authentication MPLS
  ip nhrp network-id 1000
  ip nhrp holdtime 600
  ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
  ip nhrp shortcut
  mpls nhrp
  mpls mtu 1400
  tunnel source Serial4/0
  tunnel mode gre multipoint
  tunnel key 1000
  tunnel vrf Outside
!
!
router bgp 1
  bgp router-id 10.0.0.12
  neighbor 10.0.0.1 remote-as 1
  neighbor 10.0.0.1 update-source Tunnel0
  !
  address-family vpnv4
    neighbor 10.0.0.1 activate
    neighbor 10.0.0.1 send-community extended
  exit-address-family
!
```

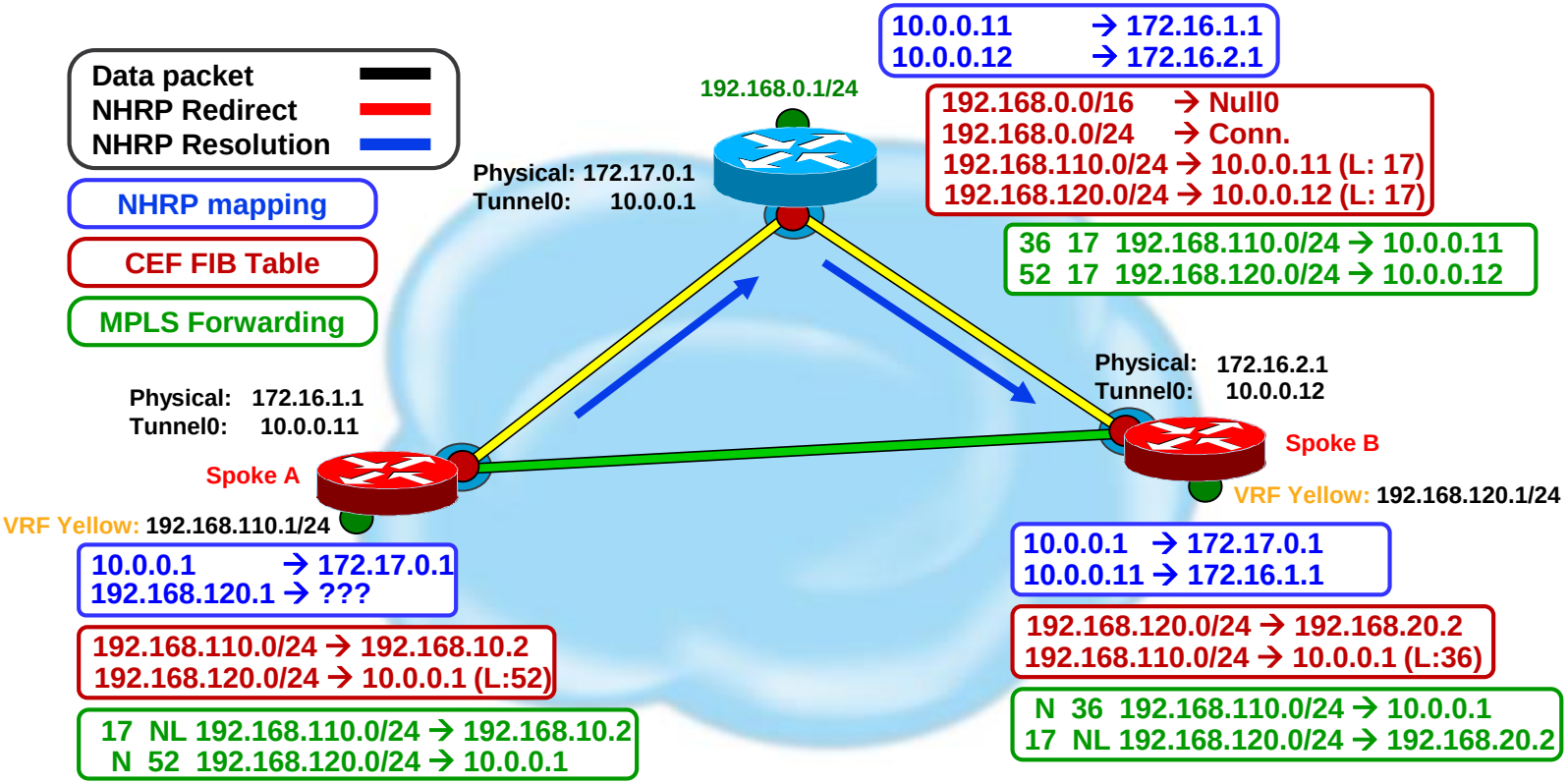
MPLS over DMVPN

NHRP Redirects



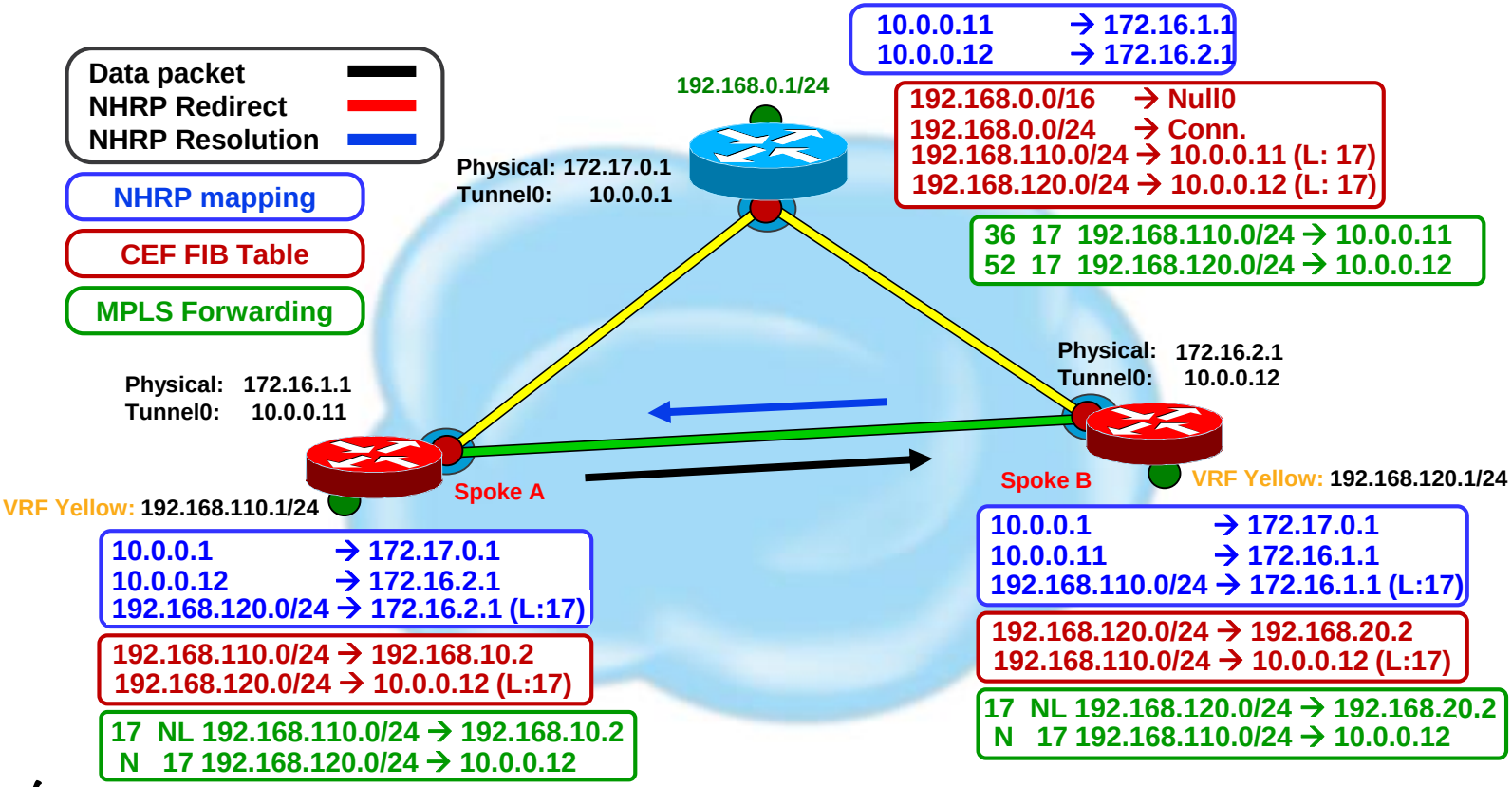
MPLS over DMVPN

NHRP Resolution Request



MPLS over DMVPN

NHRP Resolution Reply



MPLS over DMVPN – 2547oDMVPN

Routing Tables – Hub

Global

```
C       10.0.0.0/24 is directly connected, Tunnel0
```

vrf Outside

```
S*   0.0.0.0/0 [1/0] via 172.17.0.2
C      172.17.0.0/30 is directly connected, Serial4/0
```

vrf Yellow

```
S*   0.0.0.0/0 [1/0] via 192.168.254.2, Ethernet3/0
C    192.168.0.0/24 is directly connected, Ethernet0/0
D    192.168.100.0/24 [90/307200] via 192.168.0.2, 03:27:24, Ethernet0/0

B    192.168.10.0/24 [200/0] via 10.0.0.11, 03:26:56
B    192.168.20.0/24 [200/0] via 10.0.0.12, 03:26:56
B    192.168.110.0/24 [200/307200] via 10.0.0.11, 03:26:56
B    192.168.120.0/24 [200/307200] via 10.0.0.12, 03:26:56
```

vrf Red

```
S*   0.0.0.0/0 [1/0] via 192.168.254.2, Ethernet3/0
C    192.168.2.0/24 is directly connected, Ethernet1/0
D    192.168.102.0/24 [90/307200] via 192.168.1.2, 03:27:22, Ethernet1/0

B    192.168.22.0/24 [200/0] via 10.0.0.12, 03:26:54
B    192.168.32.0/24 [200/0] via 10.0.0.13, 03:26:54
B    192.168.122.0/24 [200/307200] via 10.0.0.12, 03:26:54
B    192.168.132.0/24 [200/307200] via 10.0.0.13, 03:26:54
```

vrf Green

```
S*   0.0.0.0/0 [1/0] via 192.168.254.2, Ethernet3/0
C    192.168.4.0/24 is directly connected, Ethernet2/0
D    192.168.104.0/24 [90/307200] via 192.168.2.2, 03:27:18, Ethernet2/0

B    192.168.24.0/24 [200/0] via 10.0.0.12, 03:26:53
B    192.168.34.0/24 [200/0] via 10.0.0.13, 03:26:53
B    192.168.134.0/24 [200/307200] via 10.0.0.13, 03:26:53
B    192.168.124.0/24 [200/307200] via 10.0.0.12, 03:26:53
```

MPLS over DMVPN – 2547oDMVPN

MPLS Tables – Hub

Hub1#show mpls forwarding

Local label	Outgoing label	Prefix or Tunnel Id	Bytes label Switched	Outgoing interface	Next Hop
25	No Label	192.168.254.0/24[V]	0	aggregate/Internet	
16	No Label	0.0.0.0/0[V]	0	Et3/0	192.168.254.2
17	No Label	192.168.0.0/24[V]	0	aggregate/Yellow	
18	No Label	192.168.100.0/24[V]	0	Et0/0	192.168.0.2
36	17	192.168.110.0/24[V]	0	Tu0	10.0.0.11
50	16	192.168.10.0/24[V]	0	Tu0	10.0.0.11
51	16	192.168.20.0/24[V]	0	Tu0	10.0.0.12
52	17	192.168.120.0/24[V]	0	Tu0	10.0.0.12
19	No Label	0.0.0.0/0[V]	0	Et3/0	192.168.254.2
21	No Label	192.168.102.0/24[V]	0	Et1/0	192.168.2.2
30	24	192.168.122.0/24[V]	0	Tu0	10.0.0.12
35	19	192.168.32.0/24[V]	0	Tu0	10.0.0.13
38	No Label	192.168.2.0/24[V]	0	aggregate/Red	
44	21	192.168.132.0/24[V]	0	Tu0	10.0.0.13
53	23	192.168.22.0/24[V]	0	Tu0	10.0.0.12
20	No Label	192.168.104.0/24[V]	0	Et2/0	192.168.4.2
22	No Label	0.0.0.0/0[V]	0	Et3/0	192.168.254.2
24	No Label	192.168.4.0/24[V]	0	aggregate/Green	
31	25	192.168.124.0/24[V]	0	Tu0	10.0.0.12
32	20	192.168.134.0/24[V]	0	Tu0	10.0.0.13
34	22	192.168.24.0/24[V]	0	Tu0	10.0.0.12
45	17	192.168.34.0/24[V]	0	Tu0	10.0.0.13

MPLS over DMVPN – 2547oDMVPN

MPLS Tables – Spoke 2

Spoke2#show mpls forwarding

Local label	Outgoing label	Prefix or Tunnel Id	Bytes label Switched	Outgoing interface	Next Hop
16	No Label	192.168.20.0/24[V]	0	aggregate/Yellow	
17	No Label	192.168.120.0/24[V]	23496	Et0/0	192.168.20.2
None	16	0.0.0.0/0[V]	0	Tu0	10.0.0.1
None	17	192.168.0.0/24[V]	0	Tu0	10.0.0.1
None	50	192.168.10.0/24[V]	0	Tu0	10.0.0.1
None	18	192.168.100.0/24[V]	0	Tu0	10.0.0.1
None	36	192.168.110.0/24[V]	0	Tu0	10.0.0.1
23	No Label	192.168.22.0/24[V]	0	aggregate/Red	
24	No Label	192.168.122.0/24[V]	0	Et1/0	192.168.22.2
None	19	0.0.0.0/0[V]	0	Tu0	10.0.0.1
None	38	192.168.2.0/24[V]	0	Tu0	10.0.0.1
None	35	192.168.32.0/24[V]	0	Tu0	10.0.0.1
None	21	192.168.102.0/24[V]	0	Tu0	10.0.0.1
None	44	192.168.132.0/24[V]	0	Tu0	10.0.0.1
22	No Label	192.168.24.0/24[V]	0	aggregate/Green	
25	No Label	192.168.124.0/24[V]	0	Et2/0	192.168.24.2
None	22	0.0.0.0/0[V]	0	Tu0	10.0.0.1
None	24	192.168.4.0/24[V]	0	Tu0	10.0.0.1
None	45	192.168.34.0/24[V]	0	Tu0	10.0.0.1
None	20	192.168.104.0/24[V]	0	Tu0	10.0.0.1
None	32	192.168.134.0/24[V]	0	Tu0	10.0.0.1

MPLS over DMVPN – 2547oDMVPN

Routing Tables – Spoke2

Spoke2: Yellow

```
B*   0.0.0.0/0 [200/0] via 10.0.0.1, 00:39:20
B    192.168.0.0/24 [200/0] via 10.0.0.1, 00:39:20
B    192.168.10.0/24 [200/0] via 10.0.0.1, 00:39:20
C       192.168.20.0/24 is directly connected, Ethernet0/0
B    192.168.100.0/24 [200/307200] via 10.0.0.1, 00:39:20
B    192.168.110.0/24 [200/307200] via 10.0.0.1, 00:39:20
D    192.168.120.0/24 [90/307200] via 192.168.20.2, 5w0d, Ethernet0/0
```

Red

```
B*   0.0.0.0/0 [200/0] via 10.0.0.1, 00:40:37
B    192.168.2.0/24 [200/0] via 10.0.0.1, 00:40:37
C       192.168.22.0/24 is directly connected, Ethernet1/0
B    192.168.32.0/24 [200/0] via 10.0.0.1, 00:40:33
B    192.168.102.0/24 [200/307200] via 10.0.0.1, 00:40:37
D    192.168.122.0/24 [90/307200] via 192.168.22.2, 4w1d, Ethernet1/0
B    192.168.132.0/24 [200/307200] via 10.0.0.1, 00:40:33
```

Green

```
B*   0.0.0.0/0 [200/0] via 10.0.0.1, 00:41:07
B    192.168.4.0/24 [200/0] via 10.0.0.1, 00:41:07
C       192.168.24.0/24 is directly connected, Ethernet2/0
B    192.168.34.0/24 [200/0] via 10.0.0.1, 00:41:03
B    192.168.104.0/24 [200/307200] via 10.0.0.1, 00:41:07
D    192.168.124.0/24 [90/307200] via 192.168.24.2, 4w1d, Ethernet2/0
B    192.168.134.0/24 [200/307200] via 10.0.0.1, 00:41:03
```

MPLS over DMVPN – 2547oDMVPN

MPLS Tables – Spoke 1 and 3

Spoke1: Yellow

Local label	Outgoing label	Prefix or Tunnel Id	Bytes label Switched	Outgoing interface	Next Hop
16	No Label	192.168.10.0/24[V]	0	aggregate/Yellow	
17	No Label	192.168.110.0/24[V]	23496	Et0/0	192.168.20.2
None	16	0.0.0.0/0[V]	0	Tu0	10.0.0.1
None	17	192.168.0.0/24[V]	0	Tu0	10.0.0.1
None	51	192.168.20.0/24[V]	0	Tu0	10.0.0.1
None	18	192.168.100.0/24[V]	0	Tu0	10.0.0.1
None	52	192.168.120.0/24[V]	0	Tu0	10.0.0.1

Spoke3: Red

Local label	Outgoing label	Prefix or Tunnel Id	Bytes label Switched	Outgoing interface	Next Hop
19	No Label	192.168.32.0/24[V]	0	aggregate/Red	
21	No Label	192.168.132.0/24[V]	0	Et1/0	192.168.22.2
None	19	0.0.0.0/0[V]	0	Tu0	10.0.0.1
None	38	192.168.2.0/24[V]	0	Tu0	10.0.0.1
None	53	192.168.22.0/24[V]	0	Tu0	10.0.0.1
None	21	192.168.102.0/24[V]	0	Tu0	10.0.0.1
None	30	192.168.122.0/24[V]	0	Tu0	10.0.0.1

Green

Local label	Outgoing label	Prefix or Tunnel Id	Bytes label Switched	Outgoing interface	Next Hop
17	No Label	192.168.34.0/24[V]	0	aggregate/Green	
20	No Label	192.168.134.0/24[V]	0	Et2/0	192.168.24.2
None	22	0.0.0.0/0[V]	0	Tu0	10.0.0.1
None	24	192.168.4.0/24[V]	0	Tu0	10.0.0.1
None	34	192.168.24.0/24[V]	0	Tu0	10.0.0.1
None	20	192.168.104.0/24[V]	0	Tu0	10.0.0.1
None	31	192.168.124.0/24[V]	0	Tu0	10.0.0.1

MPLS over DMVPN – 2547oDMVPN

Routing Tables – Spoke 1 and 3

Spoke1: Yellow

```
B*   0.0.0.0/0 [200/0] via 10.0.0.1, 00:43:03
B    192.168.0.0/24 [200/0] via 10.0.0.1, 00:43:03
C       192.168.10.0/24 is directly connected, Ethernet0/0
B    192.168.20.0/24 [200/0] via 10.0.0.1, 00:43:03
B    192.168.100.0/24 [200/307200] via 10.0.0.1, 00:43:03
D    192.168.110.0/24 [90/307200] via 192.168.10.2, 5w0d, Ethernet0/0
B    192.168.120.0/24 [200/307200] via 10.0.0.1, 00:43:03
```

Spoke3: Red

```
B*   0.0.0.0/0 [200/0] via 10.0.0.1, 00:41:59
B    192.168.2.0/24 [200/0] via 10.0.0.1, 00:41:59
B    192.168.22.0/24 [200/0] via 10.0.0.1, 00:41:59
C       192.168.32.0/24 is directly connected, Ethernet1/0
B    192.168.102.0/24 [200/307200] via 10.0.0.1, 00:41:59
B    192.168.122.0/24 [200/307200] via 10.0.0.1, 00:41:59
D    192.168.132.0/24 [90/307200] via 192.168.32.2, 4w1d, Ethernet1/0
```

Green

```
B*   0.0.0.0/0 [200/0] via 10.0.0.1, 00:42:32
B    192.168.4.0/24 [200/0] via 10.0.0.1, 00:42:32
B    192.168.24.0/24 [200/0] via 10.0.0.1, 00:42:32
C       192.168.34.0/24 is directly connected, Ethernet2/0
B    192.168.104.0/24 [200/307200] via 10.0.0.1, 00:42:32
B    192.168.124.0/24 [200/307200] via 10.0.0.1, 00:42:32
D    192.168.134.0/24 [90/307200] via 192.168.34.2, 4w1d, Ethernet2/0
```

MPLS over DMVPN – 2547oDMVPN

Summary

- Single DMVPN mGRE tunnel on all routers
- MPLS
  - NHRP (mpls nhrp) is used for MPLS tag distribution on hub-spoke and spoke-spoke tunnels
  - Hub is configured as MPLS P/PE and Spoke as MPLS PE
  - Spoke to spoke packets are MPLS tag-switched via Hub then through spoke-spoke tunnel once it is up
- Routing
  - EIGRP, OSPF or BGP is used for routing outside of DMVPN (on “LAN”)
  - MP-BGP used for routing over DMVPN
    - Redistribute between RP on “LAN” and MP-BGP for transport over DMVPN
- “Outside” f-VRF used for forwarding tunnel packets on WAN (transport)
- Tunnel Interface is in “Global”

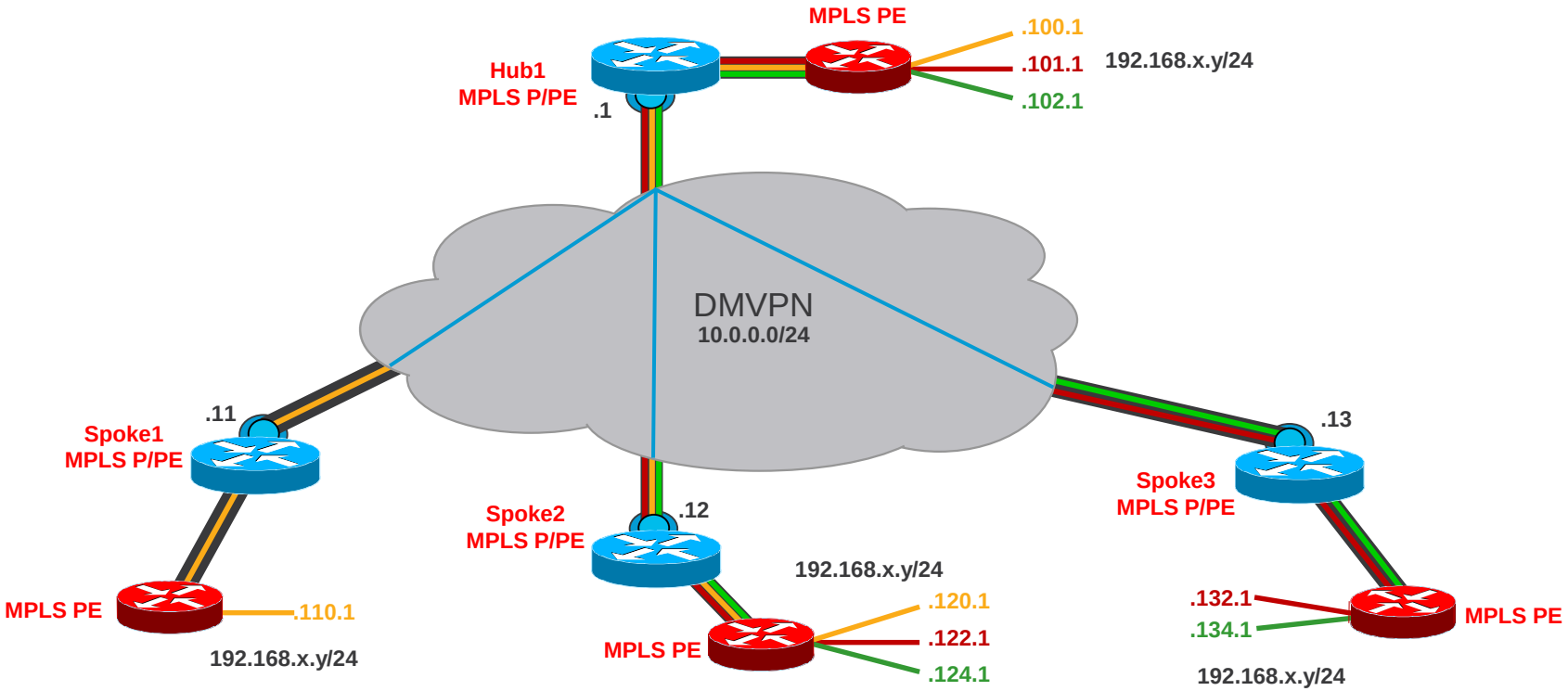
Extending MPLS over DMVPN

Extending MPLS over DMVPN

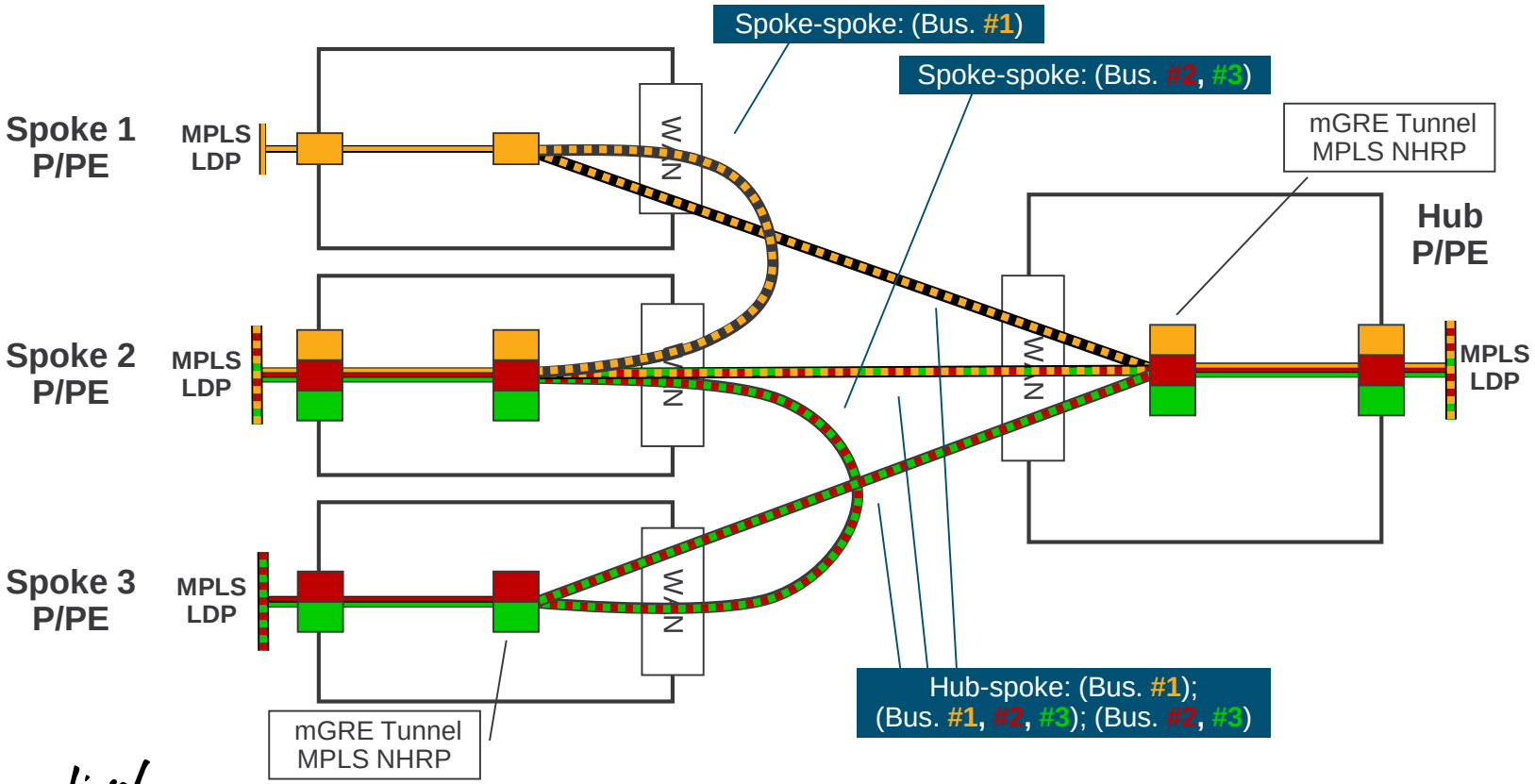
- Mostly the same as above MPLS over DMVPN
  - Single DMVPN mGRE tunnel on Hub and Spoke routers
- MPLS:
  - NHRP (mpls nhrp) on DMVPN and LDP (mpls ip) on LAN, for MPLS label distribution
  - Hubs and Spokes – Must be MPLS P/PE
    - Must define VRFs, even if not used on any local interface
  - Nodes behind Hubs and Spokes can be MPLS P, PE or P/PE
- Routing
  - MP-BGP over DMVPN and LANs
    - Hubs are BGP Route-reflectors for Spokes
    - Hubs and Spokes may be BGP route-reflectors for MPLS PE nodes behind them

MPLS over DMVPN – Extending an MPLS

Logical Topology



MPLS over DMVPN – Extending an MPLS



Extending MPLS over DMVPN

Current Issues with spoke-spoke tunnels

- NHRP redirects
  - Issue:
    - Hubs send redirects to IP data packet source → injected into MPLS on tunnel
    - MPLS tag-switches NHRP redirect through to MPLS PE behind spoke → dropped
    - Spoke doesn't get NHRP redirect → no spoke-spoke tunnel
  - Workaround:
    - Configure Spoke to summarize IP subnets behind it
    - NHRP redirect packet will be MPLS tag-switched to spoke → spoke-spoke tunnel
    - Problem:
      - May not be able to summarize IP subnets without covering subnets behind other spokes
      - If subnets change, may need to manually modify summarization on spoke
  - Solution:
    - Use regular IP forwarding over hub-spoke tunnel for NHRP redirect

Extending MPLS over DMVPN

Current Issues with spoke-spoke tunnels

- NHRP spoke-spoke routes
  - Issue:
    - NHRP on spoke injects route to use spoke-spoke tunnel with MPLS tag into RIB/FIB
    - This route is not redistributed into MP-BGP to be advertised to MPLS PE behind spoke
    - MPLS PE still uses original tag that tag-switches the packet via the Hub
    - Spoke-spoke tunnel is not used
  - Workaround:
    - Configure MP-BGP with network statements for data subnets behind other spokes
      - MP-BGP will pick up the spoke-spoke route and tag when NHRP inserts it into the RIB
    - Not Scalable: would need to add MP-BGP statement for all subnets behind other spokes
  - Solution:
    - Enable ‘redistribute nhrp’ under BGP configuration (also enable for EIGRP and OSPF)
    - Then MP-BGP will automatically pick up NHRP routes
    - May also want to use a route-map to filter which routes are picked up by MP-BGP

Recent and New Features

DMVPN Other Recent and Future Features

- Recently Available
  - Metadata (CMD, NSH) over DMVPN
    - PfR; TrustSec (SGT)
  - Multiple Tunnel Termination (MTT)
    - NHRP fixes (done); CEF fixes (next)
- Coming Next
  - Extending MPLS over DMVPN fixes
    - MPLS on DMVPN part of larger MPLS
  - VXLAN-GPE encapsulation for DMVPN
    - Support for multiple spokes behind NPAT (**16.4.1)
- Future
  - More MPLS over DMVPN
    - mVPN and VPLS support

- Future (cont)
  - DMVPN extended authentication
    - Strong NHRP authentication using HMAC
    - NHRP spoke authentication using Radius
    - Dynamic Tunnel Key on spokes
  - Increased Multicast Support
    - Limited spoke-spoke multicast support
      - Large spoke to many small spokes
    - Native Multicast over DMVPN
      - Tunnel packets with multicast destination
      - ISP network does replication
      - ESON KS (Group and pair-wise keys)
  - NHRP route advertisement
  - GRE tunnel sub-interfaces
    - EVN WAN using DMVPN

Thank you

Complete Your Online Session Evaluation

- Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 gift card.
- Complete your session surveys through the Cisco Live mobile app or on www.CiscoLive.com/us.

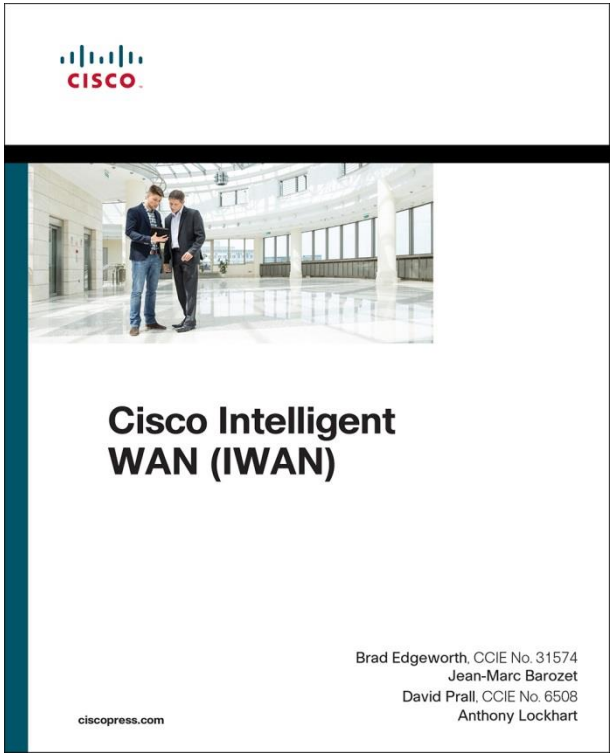
Don't forget: Cisco Live sessions will be available for viewing on demand after the event at www.CiscoLive.com/Online.

Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
  - LABIOT-2012 Implementing Dynamic Multipoint VPN
- Lunch & Learn
- Meet the Engineer 1:1 meetings
- Related sessions

Recommended Reading

- Explains all key IWAN technologies and components
- VIRL labs are available so that you can practice these concepts as you read them in the book



- Copies are available at the CLUS Cisco Press bookstore

Come Meet The Authors

- Anthony, Brad, David, and Jean-Marc are signing books at Cisco Press bookstore on Weds. 1:30 – 2 PM

Thank you


cisco
You're
Cisco live!

Extras

Agenda

- DMVPN Design Overview
  - DMVPN General
  - IWAN Specific
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects

Phase 2 – Features

- Single mGRE interface with ‘tunnel protection …’
  - On Hubs and Spokes
  - Hubs must be inter-connected in a “Daisy chain” over same mGRE tunnel
  - IKE authentication information (Certificates, Wildcard Pre-shared Keys)
- Spoke-spoke data traffic direct
  - Reduced load on hub
  - Reduced latency
    - Single IPsec encrypt/decrypt
- Routing Protocol
  - Still hub-and-spoke
  - Cannot summarize spoke routes on hub
  - Routes on spokes must have IP next-hop of remote spoke (preserve next-hop)

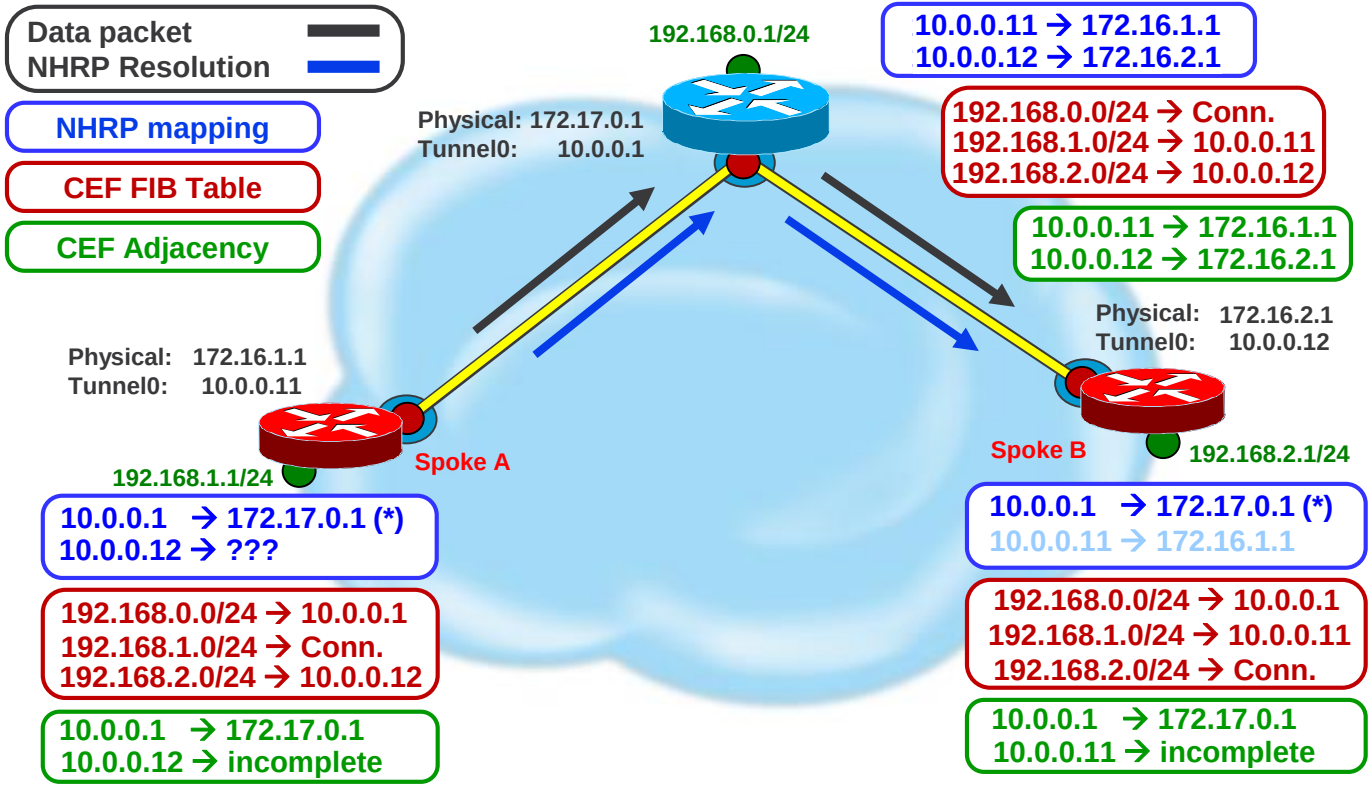
Phase 2 – Process switching

- IP Data packet is forwarded out tunnel interface to IP next-hop from routing table
- NHRP looks in mapping table for IP destination
  - If Entry Found
    - Forward to NBMA from mapping table – overriding IP next-hop
  - If No Entry Found
    - Forward to IP next-hop (if in NHRP table) otherwise to NHS
    - If arriving interface was not tunnel interface
      - Initiate NHRP Resolution Request for IP next-hop and send via NHS path (first up NHS)
  - If (no socket) Entry Found
    - If arriving interface is not tunnel interface – convert entry to (socket)
    - Trigger IPsec to bring up crypto socket
    - Forward to IP next-hop (if in NHRP table) otherwise to NHS

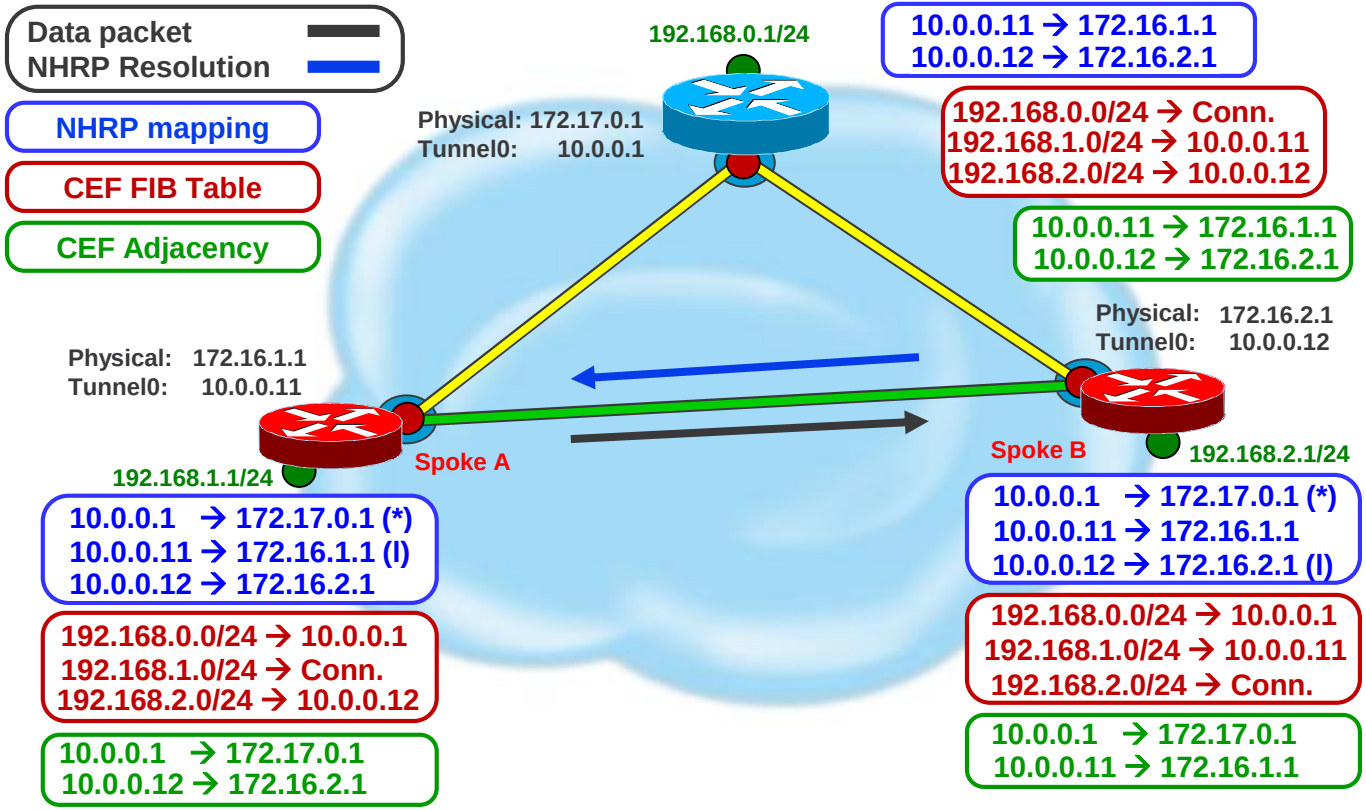
Phase 2 – CEF Switching

- IP Data packet is forwarded out tunnel interface to IP next-hop from FIB table
- If adjacency is of type Valid
  - Packet is encapsulated and forwarded by CEF out tunnel interface
  - NHRP is not involved
- If adjacency is of type Glean or Incomplete
  - Punt packet to process switching
  - If original arriving interface was not this tunnel interface
  - Initiate NHRP Resolution Request for IP next-hop
    - Send resolution request for IP next-hop (tunnel IP address) of remote Spoke
    - Resolution request forwarded via NHS path (first up NHS)
    - Resolution reply is used to create NHRP mapping and to complete the Adjacency

Phase 2 – NHRP Resolution Request



Phase 2 – NHRP Resolution Reply



Phase 2 – NHRP Resolution Response Processing

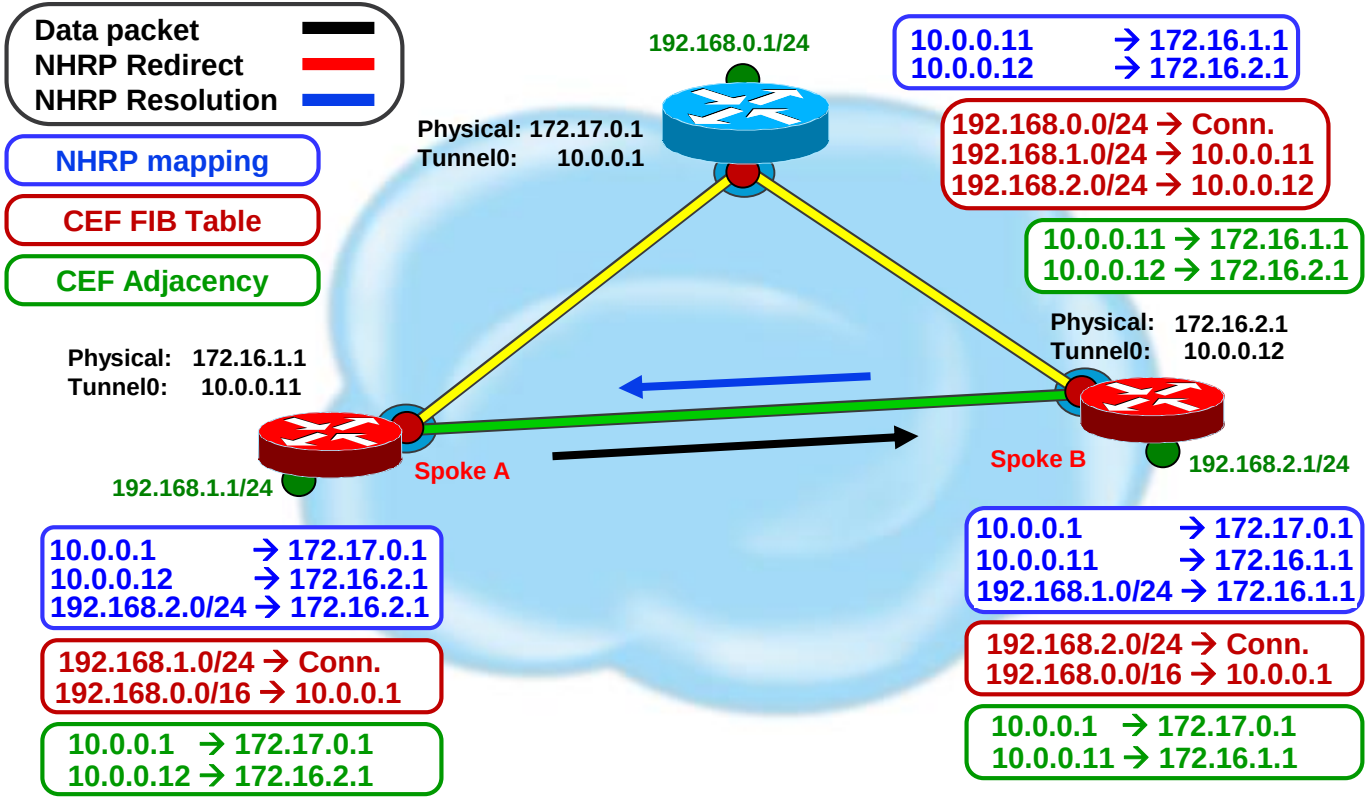
- Receive NHRP Resolution reply
  - If using IPsec (tunnel protection …) then
    - Trigger IPsec to setup ISAKMP and IPsec SAs for tunnel
    - Data packets still forwarded via spoke-hub-…-hub-spoke path
    - IPsec triggers back to NHRP when done
- Install new mapping in NHRP mapping table
- Send trigger to CEF to complete corresponding CEF adjacency
  - Data packets now forwarded via direct spoke-spoke tunnel by CEF
  - NHRP no longer involved

Phase 2 – Refresh or Remove Dynamic mappings

- Dynamic NHRP mapping entries have finite lifetime
  - Controlled by 'ip nhrp holdtime …' on source of mapping (spoke)
- Background process checks mapping entry every 60 seconds
  - Process-switching
    - Used flag set each time mapping entry is used
    - If used flag is set and expire time < 120 seconds then refresh entry, otherwise clear used flag
  - CEF-switching
    - If expire time < 120 seconds, CEF Adjacency entry marked "stale"
    - If "stale" CEF Adjacency entry is then used, signal to NHRP to refresh entry
- Another resolution request is sent to refresh entry
  - Resolution request via NHS path; reply via direct tunnel
- If entry expires it is removed
  - If using IPsec → Trigger IPsec to remove IPsec/ISAKMP SAs

Phase 3 – NHRP Resolution Reply

(Prior to 15.2(1)T – ISR, 7200)



Phase 3 – CEF Switching

Data Packet Forwarding

(Prior to 15.2(1)T – ISR, 7200)

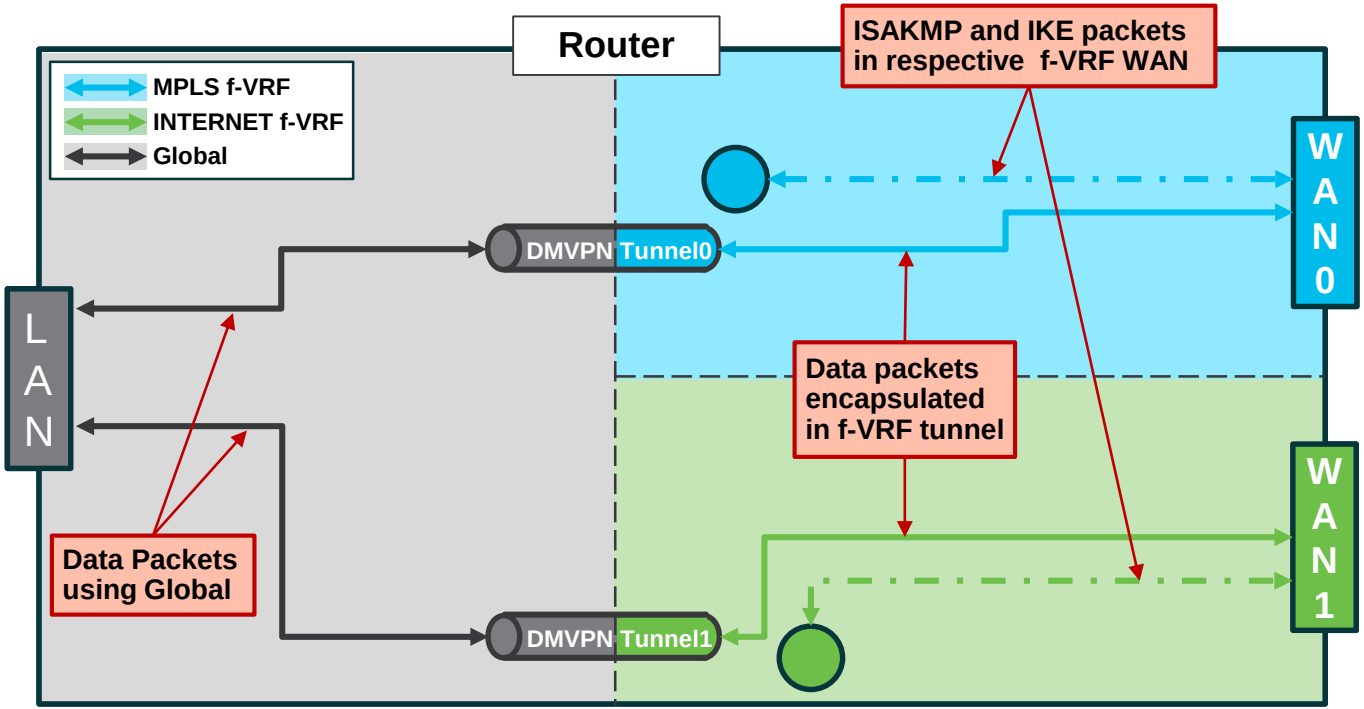
- IP Data packet is forwarded out tunnel interface
  1. IP next-hop from CEF FIB mapped to Adjacency
     If adjacency is:
     - Glean or Incomplete → Punt to process switching
     - Valid → Select adjacency for the packet
  2. NHRP in Outbound CEF Feature path
     Look up packet IP destination in NHRP mapping table
     - Matching entry: Reselect adjacency → use direct spoke-spoke tunnel
     - No matching entry: Leave CEF adjacency → packet goes to hub
- If packet arrived on and is forwarded out the same tunnel interface
  - Forward data packet
  - If 'ip nhrp redirect' is on inbound tunnel then send NHRP redirect
- Packet is encapsulated, encrypted and forwarded

Interaction with IWAN

Agenda

- DMVPN Design Overview
  - General and IWAN Specific
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- Interaction with IWAN
  - f-VRFs
  - NHRP the RIB and PfR
- Recent and New Features

DMVPN with IWAN f-VRFs



DMVPN with IWAN f-VRFs

- Create VRF for each transport WAN interface (Ex: INTERNET, MPLS)
  - vrf definition <fvrf-name>
- “Outside” of tunnel is in front-door VRF (f-VRF)
  - interface tunnel<x>; tunnel vrf <fvrf-name>
- WAN (transport) interface is in f-VRF
  - interface <wan-interface>; vrf forwarding <fvrf-name>
- Crypto – ISAKMP/IKEv2 are also in f-VRFs
  - ISAKMP – need keyring for each f-VRF
  - IKEv2 – need keyring, IKEv2 profile and IPsec profile
    - Separate one for each f-VRF

    Or
    - Single one for all fVRFs by using ‘match fvrf any’ in IKEv2 profile

DMVPN with IWAN f-VRFs

f-VRF Configuration

```
vrf definition INTERNET
  ...
vrf definition MPLS
  ...
!
crypto ikev2 keyring DMVPN
  peer ANY
  address 0.0.0.0 0.0.0.0
  pre-shared-key cisco123
!
crypto ikev2 profile DMVPN
  match fvrf any
  match identity remote address 0.0.0.0
  authentication remote pre-share
  authentication local pre-share
  keyring local DMVPN
  dpd 20 5 on-demand                    ! Spokes only
!
crypto ipsec transform-set DMVPN esp-aes 256 esp-sha256-hmac
  mode transport
!
crypto ipsec profile DMVPN
  set transform-set DMVPN
  set ikev2-profile DMVPN
```

```
interface Tunnel0
   ip address 10.0.0.11 255.255.255.0
   ...
   tunnel source FastEthernet0
   tunnel key 100000
   tunnel vrf INTERNET
   tunnel protection ipsec profile DMVPN
interface Tunnel1
   ip address 10.0.1.11 255.255.255.0
   ...
   tunnel source FastEthernet1
   tunnel key 100001
   tunnel vrf MPLS
   tunnel protection ipsec profile DMVPN
!
interface FastEthernet0
   vrf forwarding INTERNET
   ip address 172.16.1.1 255.255.255.240
!
interface FastEthernet1
   vrf forwarding MPLS
   ip address 172.17.1.1 255.255.255.240
!
ip route vrf MPLS 0.0.0.0 0.0.0.0 172.17.1.2
ip route vrf INTERNET 0.0.0.0 0.0.0.0 172.16.1.2
```

DMVPN with IWAN f-VRFs

Routing

Spoke1#show ip route vrf *

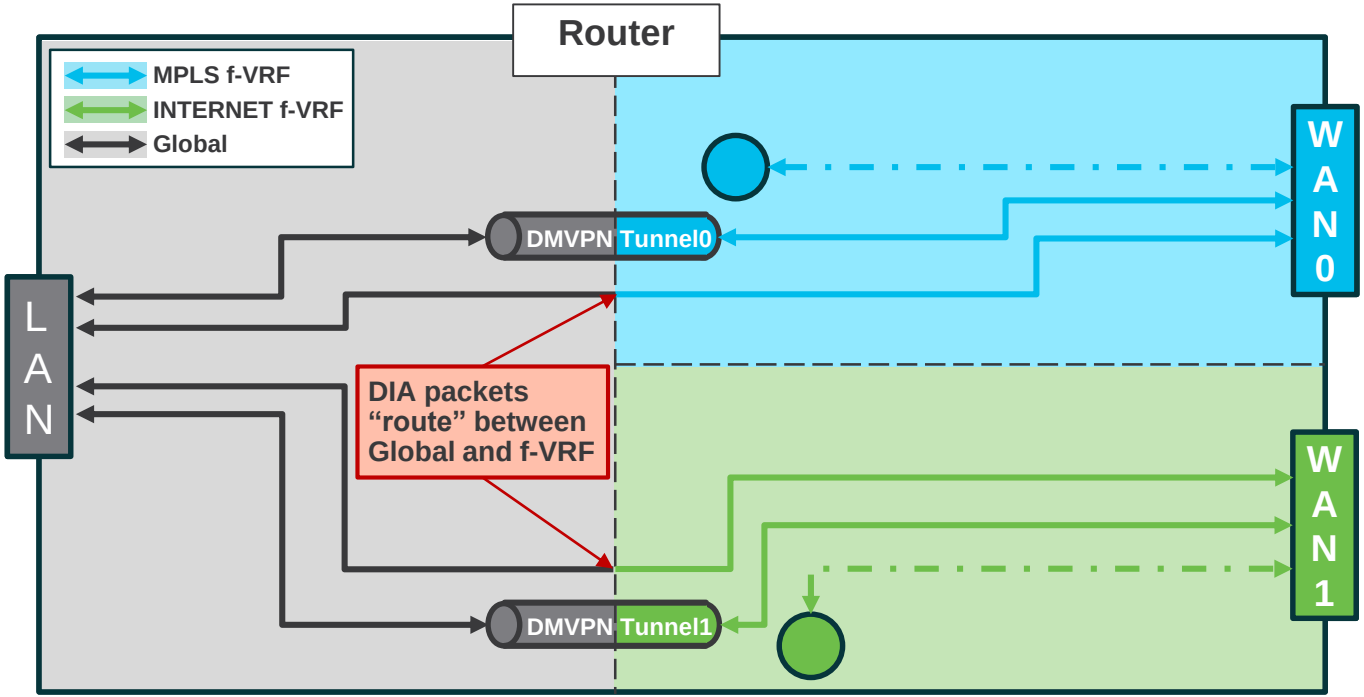
```
D*EX  0.0.0.0/0 [170/2918400] via 10.0.1.2, 00:00:04, Tunnel1
      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C        10.0.0.0/24 is directly connected, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
D     192.168.0.0/21 [90/2892800] via 10.0.1.2, 00:20:27, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
D     192.168.10.0/24 [90/2918400] via 10.0.1.2, 00:32:39, Tunnel1

Routing Table: INTERNET
Gateway of last resort is 172.16.1.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.1.2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/28 is directly connected, FastEthernet0

Routing Table: MPLS
Gateway of last resort is 172.17.1.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.17.1.2
      172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.17.1.0/28 is directly connected, FastEthernet1
```

Crypto

Spoke1#show crypto ikev2 session

```
Session-id:1845, Status:UP-ACTIVE, IKE count:1, CHILD count:1

T-id  Local            Remote           fvrf/ivrf        Status
2     172.16.1.1/500   172.16.0.1/500   INTERNET/none    READY

      Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512,
      DH Grp:5, Auth sign: PSK, Auth verify: PSK
      Life/Active Time: 86400/1263 sec
Child sa:  local selector     172.16.1.1/0 - 172.16.1.1/65535
           remote selector    172.16.0.1/0 - 172.16.0.1/65535
           ESP spi in/out:    0x86D2651B/0x1B72FEB6

Session-id:1844, Status:UP-ACTIVE, IKE count:1, CHILD count:1

T-id  Local            Remote           fvrf/ivrf        Status
1     172.17.1.1/500   172.17.0.5/500   MPLS/none        READY

      Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512,
      DH Grp:5, Auth sign: PSK, Auth verify: PSK
      Life/Active Time: 86400/1290 sec
Child sa:  local selector     172.17.1.1/0 - 172.17.1.1/65535
           remote selector    172.17.0.5/0 - 172.17.0.5/65535
           ESP spi in/out:    0xF8C63D42/0x66DEA87D
```

DMVPN with IWAN DIA



DMVPN with IWAN DIA

- Outbound
  - Block learning default through tunnel
    - Access-list: deny default; match everything else
    - Route-map: if match “learn” route
    - Apply route-map in Routing Protocol
      - EIGRP: use “distribute-list ... in <tunnel-interface>
      - BGP: use “neighbor ... in”
  - Static default route in global table forwarding out Internet WAN interface
    - ip route 0.0.0.0 0.0.0.0 <Internet-WAN> <next-hop>|dhcp <admin-distance>
- Inbound
  - Policy-based routing (PBR)
    - access-list: match internal networks
    - route-map: if match use global routing table

DMVPN with IWAN DIA

Inbound

```
interface FastEthernet0
  description INTERNET
  vrf forwarding INTERNET
  ip address 172.16.1.1 255.255.255.240
  ip policy route-map INET-INTERNAL
!
ip access-list extended INTERNAL-NETS
  permit ip any 10.0.0.0 0.0.1.255
  permit ip any 192.168.0.0 0.0.255.255
  permit ip any 172.20.0.0 0.0.255.255

route-map INET-INTERNAL permit 10
  match ip address INTERNAL-NETS
  set global
!
```

Outbound

```
router eigrp 1
  distribute-list route-map BLOCK-DEFAULT in Tunnel0
  [distribute-list route-map BLOCK-DEFAULT in Tunnel1]
  network 10.0.0.0 0.0.1.255
  network 192.168.1.0
!
ip access-list standard ALL-EXCEPT-DEFAULT
  deny   0.0.0.0
  permit any
!
route-map BLOCK-DEFAULT permit 10
  match ip address ALL-EXCEPT-DEFAULT
!
ip route 0.0.0.0 0.0.0.0 FastEthernet0 172.16.1.2 10
!
```

DMVPN with IWAN DIA

Before

```
Spoke1#show ip eigrp  topology
P 192.168.10.0/24, 1 successors, FD is 2918400
        via 10.0.1.2 (2918400/332800), Tunnel1
        via 10.0.0.1 (3020800/332800), Tunnel0
P 172.20.1.0/24, 1 successors, FD is 409600
        via 192.168.1.2 (409600/128256), Ethernet0/0
P 192.168.0.0/21, 1 successors, FD is 2892800
        via 10.0.1.2 (2892800/307200), Tunnel1
        via 10.0.0.1 (2995200/307200), Tunnel0
P 192.168.1.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
P 0.0.0.0/0, 1 successors, FD is 2918400
        via 10.0.1.2 (2918400/2636800), Tunnel1
        via 10.0.0.1 (3020800/2636800), Tunnel0

Spoke1#show ip route
D*EX  0.0.0.0/0 [170/2918400] via 10.0.1.2, 00:00:04, Tunnel1
      ...
D     172.20.1.0 [90/409600] via 192.168.1.2, 01:47:00, Ethernet0/0
D     192.168.0.0/21 [90/2892800] via 10.0.1.2, 00:20:27, Tunnel1
C     192.168.1.0/24 is directly connected, Ethernet0/0
D     192.168.10.0/24 [90/2918400] via 10.0.1.2, 00:32:39, Tunnel1
```

After

```
Spoke1#sho ip eigrp topology
P 192.168.10.0/24, 1 successors, FD is 2918400
        via 10.0.1.2 (2918400/332800), Tunnel1
        via 10.0.0.1 (3020800/332800), Tunnel0
P 172.20.1.0/24, 1 successors, FD is 409600
        via 192.168.1.2 (409600/128256), Ethernet0/0
P 192.168.0.0/21, 1 successors, FD is 2892800
        via 10.0.1.2 (2892800/307200), Tunnel1
        via 10.0.0.1 (2995200/307200), Tunnel0
P 192.168.1.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
P 0.0.0.0/0, 0 successors, FD is Infinity
        via 10.0.1.2 (2918400/2636800), Tunnel1

Spoke1#show ip route
S*    0.0.0.0/0 [10/0] via 172.16.1.2, Fastethernet0
      ...
D     172.20.1.0 [90/409600] via 192.168.1.2, 01:47:00, Ethernet0/0
D     192.168.0.0/21 [90/2892800] via 10.0.1.2, 01:46:28, Tunnel1
C     192.168.1.0/24 is directly connected, Ethernet0/0
D     192.168.10.0/24 [90/2918400] via 10.0.1.2, 01:46:28, Tunnel1
```

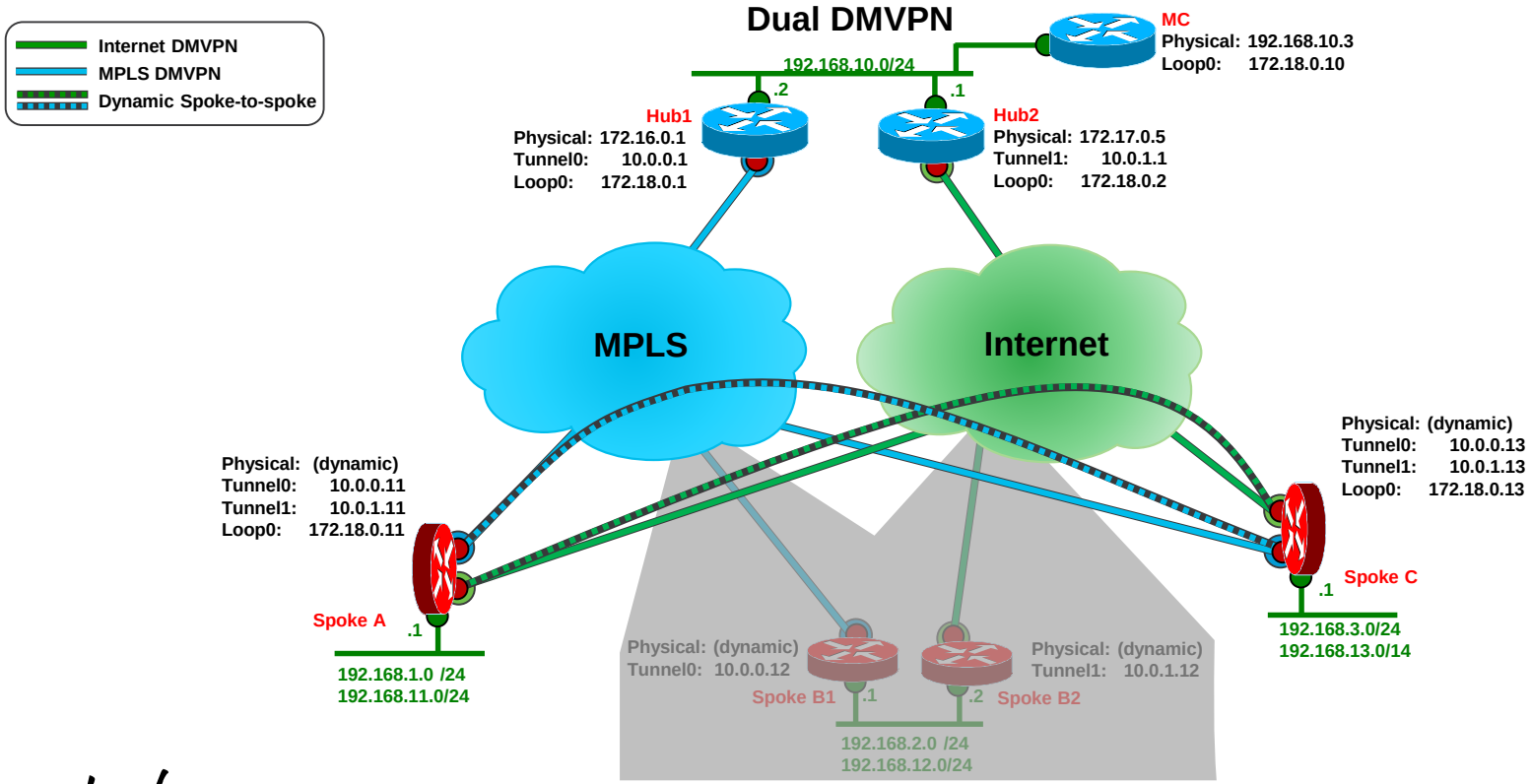
Agenda

- DMVPN Design Overview
  - General and IWAN Specific
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- Interaction with IWAN
  - f-VRFs
  - NHRP the RIB and PfRv3
- Recent and New Features

Routing Protocol (RP), NHRP and PfRv3

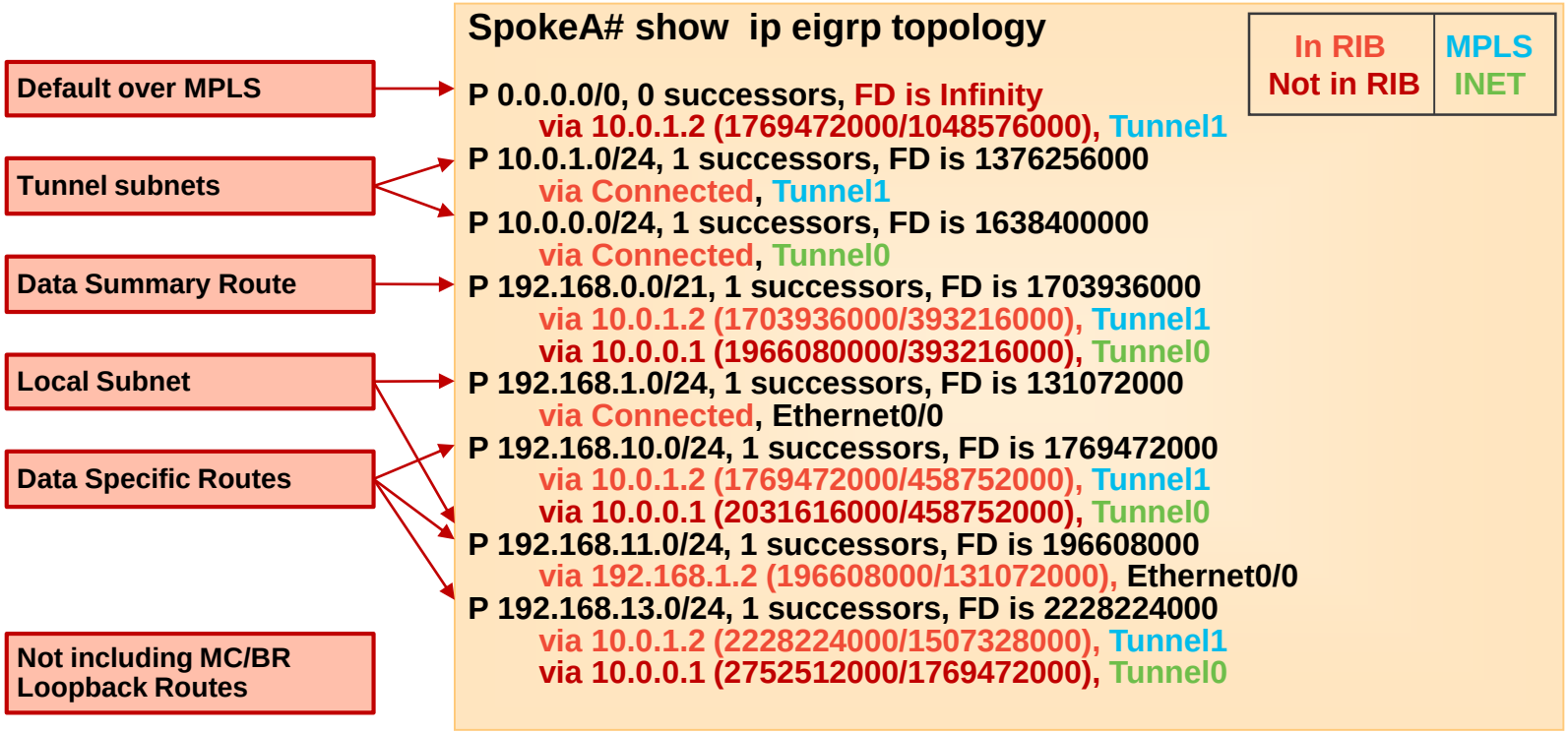
- Routing protocol (RP) – destinations outside of the DMVPN
  - Advertises reachability of these destinations over any/all DMVPNs
  - Sets base forwarding within DMVPNs via the RIB
- PfRv3 – optimize forwarding of flows over different DMVPN paths
  - PfR RIB used to control forwarding of flows
  - Lookup alternate paths directly in RP database (except OSPF)
  - Bring up alternate paths, with probe traffic
- NHRP – optimizes forwarding within a single DMVPN
  - Shortcut (spoke-spoke) tunnels
    - Triggered by data traffic, including PfRv3 probe traffic
    - Changes forwarding by making changes in the RIB
    - Tracks RIB RP entries to control adding/removing shortcut tunnel

Basic DMVPN Design for IWAN



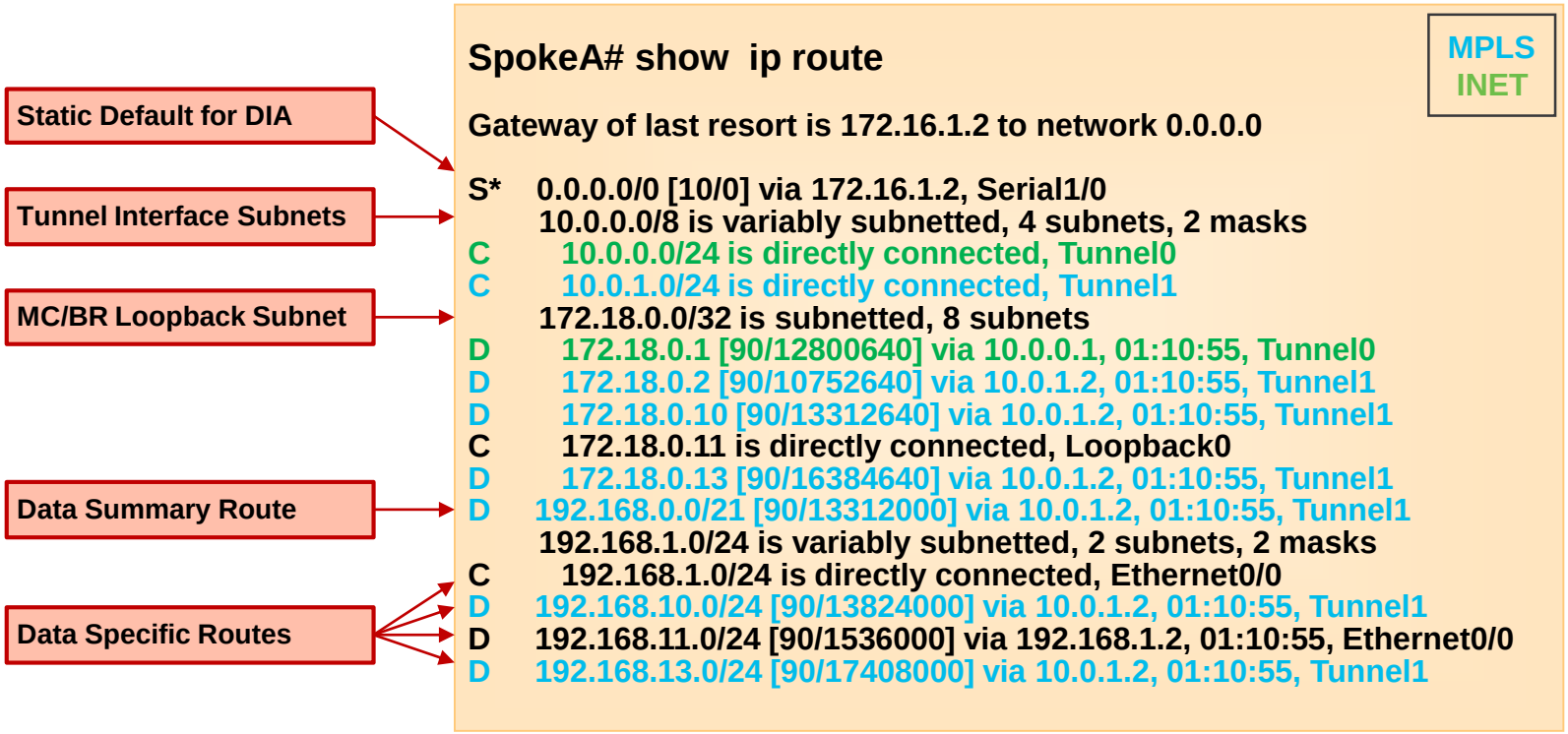
DMVPN with Routing Protocol

Routing Protocol – Both paths

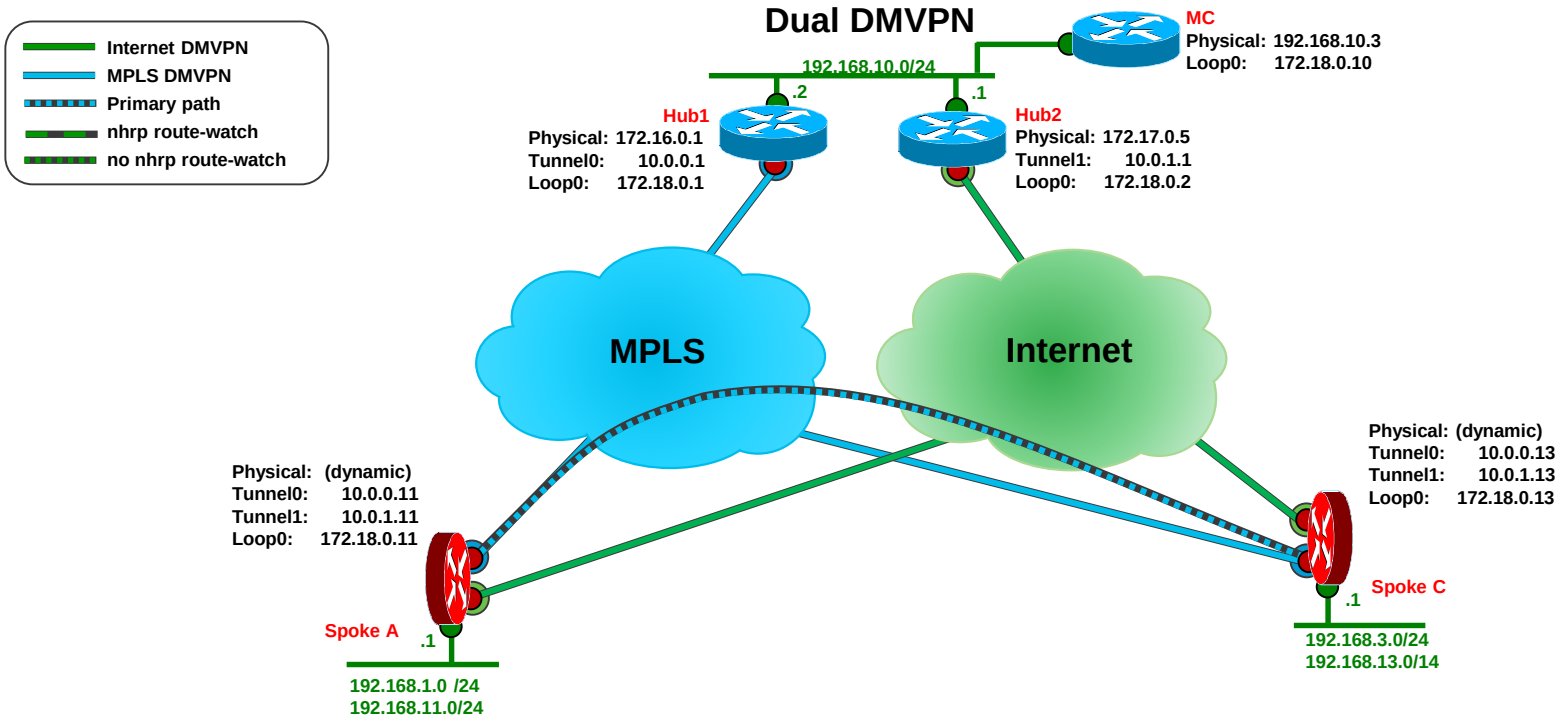


DMVPN with Routing Protocol

RIB – Path via MPLS



Forwarding over Primary DMVPN



Forwarding over Primary DMVPN

NHRP

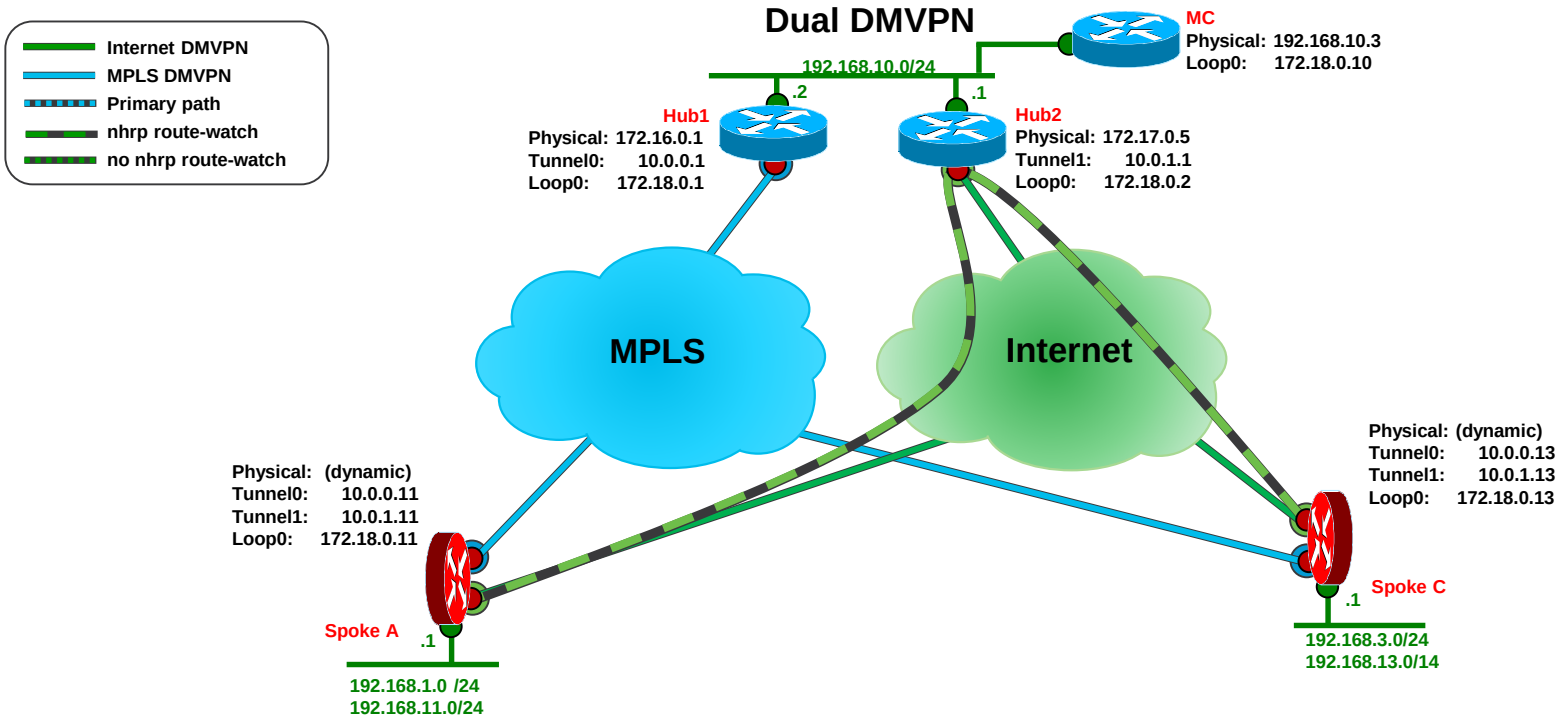
```
SpokeA# show ip nhrp

10.0.1.13/32 via 10.0.1.13
   Tunnel1 created 00:04:23, expire 00:04:19
   Type: dynamic, Flags: router nhop rib
   NBMA address: 172.17.3.1
192.168.1.0/24 via 10.0.1.11
   Tunnel1 created 00:04:25, expire 00:01:36
   Type: dynamic, Flags: router unique local
   NBMA address: 172.17.1.1
    (no-socket)
192.168.3.0/24 via 10.0.1.13
   Tunnel1 created 00:01:40, expire 00:04:19
   Type: dynamic, Flags: router rib
   NBMA address: 172.17.3.1
192.168.11.0/24 via 10.0.1.11
   Tunnel1 created 00:04:02, expire 00:01:57
   Type: dynamic, Flags: router unique local
   NBMA address: 172.17.1.1
    (no-socket)
192.168.13.0/24 via 10.0.1.13
   Tunnel1 created 00:04:02, expire 00:01:57
   Type: dynamic, Flags: router rib nho
   NBMA address: 172.17.3.1
```

RIB

```
SpokeA# show ip route

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.11/32 is directly connected, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
L        10.0.1.11/32 is directly connected, Tunnel1
H        10.0.1.13/32 is directly connected, 00:05:28, Tunnel1
D     192.168.0.0/21 [90/13312000] via 10.0.1.2, 00:11:02, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.1/32 is directly connected, Ethernet0/0
H     192.168.3.0/24 [250/1] via 10.0.1.13, 00:03:06, Tunnel1
D     192.168.10.0/24 [90/13824000] via 10.0.1.2, 00:11:02, Tunnel1
D     192.168.11.0/24 [90/1536000] via 192.168.1.2, 00:11:02, Ethernet0/0
D %   192.168.13.0/24 [90/17408000] via 10.0.1.2, 00:11:02, Tunnel1
            [NHO][90/1] via 10.0.1.13, 00:05:28, Tunnel1
```

Parent Routes

Forwarding over Secondary DMVPN (nhrp route-watch)



Forwarding over Secondary DMVPN (nhrp route-watch)

NHRP

SpokeA# show ip nhrp

```
10.0.0.13/32 via 10.0.0.13
   Tunnel0 created 00:01:01, expire 00:05:07
   Type: dynamic, Flags: router nhop
   NBMA address: 172.16.3.1
192.168.1.0/24 via 10.0.0.11
   Tunnel0 created 00:01:01, expire 00:04:58
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.1.1
    (no-socket)
192.168.3.0/24 via 10.0.0.13
   Tunnel0 created 00:01:00, expire 00:04:59
   Type: dynamic, Flags: router
   NBMA address: 172.16.3.1
192.168.11.0/24 via 10.0.0.11
   Tunnel0 created 00:00:52, expire 00:05:07
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.1.1
    (no-socket)
192.168.13.0/24 via 10.0.0.13
   Tunnel0 created 00:00:52, expire 00:05:07
   Type: dynamic, Flags: router
   NBMA address: 172.16.3.1
```

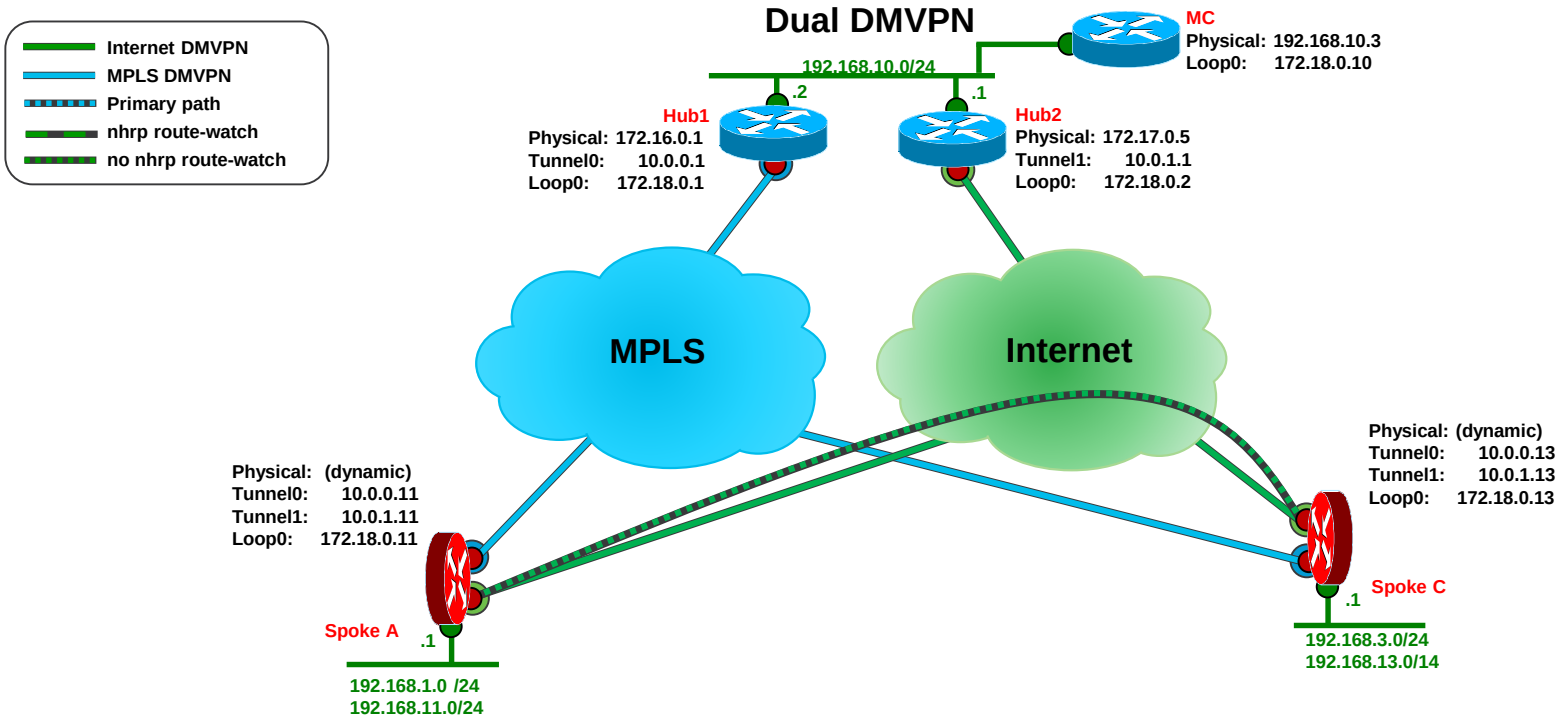
RIB

SpokeA# show ip route

```
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.11/32 is directly connected, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
L        10.0.1.11/32 is directly connected, Tunnel1
D     192.168.0.0/21 [90/13312000] via 10.0.1.2, 00:04:38, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.1/32 is directly connected, Ethernet0/0
D     192.168.10.0/24 [90/13824000] via 10.0.1.2, 00:04:38, Tunnel1
D     192.168.11.0/24 [90/1536000] via 192.168.1.2, 00:04:38, Ethernet0/0
D     192.168.13.0/24 [90/17408000] via 10.0.1.2, 00:04:38, Tunnel1
```

NHRP mapping entries not in RIB
No matching Parent Route

Forwarding over Secondary DMVPN (no nhrp route-watch)



Forwarding over Secondary DMVPN (no nhrp route-watch)

NHRP

SpokeA# show ip nhrp

```
10.0.0.13/32 via 10.0.0.13
   Tunnel0 created 00:00:36, expire 00:05:25
   Type: dynamic, Flags: router nhop rib
   NBMA address: 172.16.3.1
192.168.1.0/24 via 10.0.0.11
   Tunnel0 created 00:00:35, expire 00:05:24
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.1.1
    (no-socket)
192.168.3.0/24 via 10.0.0.13
   Tunnel0 created 00:00:34, expire 00:05:25
   Type: dynamic, Flags: router rib
   NBMA address: 172.16.3.1
192.168.11.0/24 via 10.0.0.11
   Tunnel0 created 00:00:24, expire 00:05:35
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.1.1
    (no-socket)
192.168.13.0/24 via 10.0.0.13
   Tunnel0 created 00:00:24, expire 00:05:35
   Type: dynamic, Flags: router rib nho
   NBMA address: 172.16.3.1
```

RIB

SpokeA# show ip route

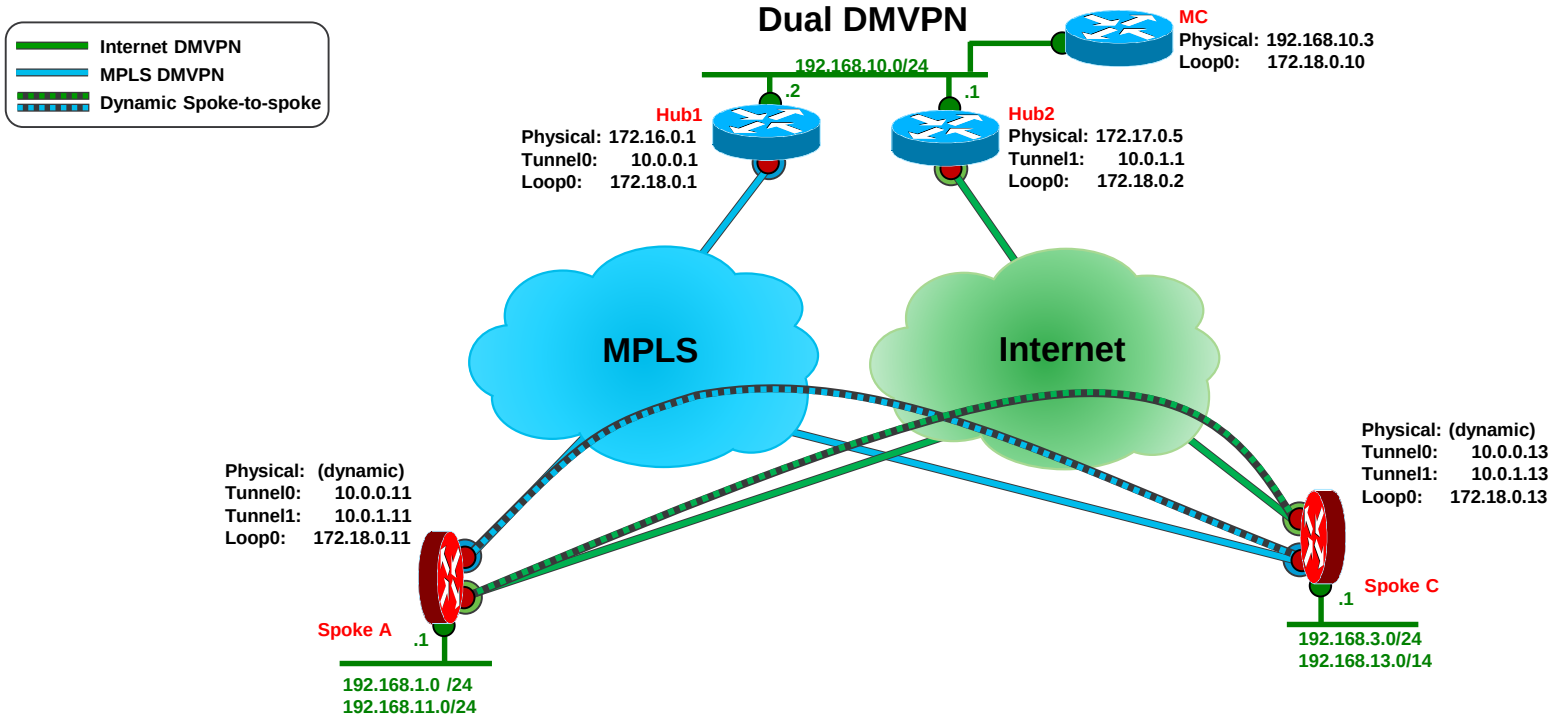
```
      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.11/32 is directly connected, Tunnel0
H        10.0.0.13/32 is directly connected, 00:00:34, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
L        10.0.1.11/32 is directly connected, Tunnel1
D     192.168.0.0/21 [90/13312000] via 10.0.1.2, 00:11:02, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.1/32 is directly connected, Ethernet0/0
H     192.168.3.0/24 [250/1] via 10.0.0.13, 00:00:34, Tunnel0
D     192.168.10.0/24 [90/13824000] via 10.0.1.2, 00:11:02, Tunnel1
D     192.168.11.0/24 [90/1536000] via 192.168.1.2, 00:11:02, Ethernet0/0
D %   192.168.13.0/24 [90/17408000] via 10.0.1.2, 00:11:02, Tunnel1
            [NHO][90/1] via 10.0.0.13, 00:00:28, Tunnel0
```

No Check for Parent Routes

Building spoke-spoke tunnels with NHRP and PfRv3

- PfRv3 Controlled Data flows
  - Forwards data flows over both primary and secondary DMVPN
    - PfR controls any load-balancing
  - Uses PfR Loopback as next-hop (Ex: 172.18.0.x)
    - NHRP triggered to build spoke-spoke tunnel over both DMVPNs
      - NHRP mapping entries to Loopback (Ex: 172.18.0.x)
      - NHRP modifies RIB for Loopback next-hop
      - If routing changes → PfR controlled flows quickly rerouted
- PfRv3 Uncontrolled Data flows
  - Data flows forwarded via the RIB
  - Uses primary DMVPN
  - Need ECMP routes to load-balancing over both DMVPNs

Building spoke-spoke tunnels with NHRP and PfRv3



Forwarding over Primary and Secondary DMVPN

NHRP

SpokeA# show ip nhrp brief

Target	Via	NBMA	Mode	Intfc
10.0.0.1/32	10.0.0.1	172.16.0.1	static	Tu0
10.0.0.11/32	10.0.0.11	172.16.1.1	dyn,loc	Tu0
10.0.0.13/32	10.0.0.13	172.16.3.1	dyn,rib	Tu0
172.18.0.11/32	10.0.0.11	172.16.1.1	dyn,loc	Tu0
172.18.0.13/32	10.0.0.13	172.16.3.1	dyn,nho	Tu0
10.0.1.2/32	10.0.1.2	172.17.0.5	static	Tu1
10.0.1.11/32	10.0.1.11	172.17.1.1	dyn,loc	Tu1
10.0.1.13/32	10.0.1.13	172.17.3.1	dyn,rib	Tu1
172.18.0.11/32	10.0.1.11	172.17.1.1	dyn,loc	Tu1
172.18.0.13/32	10.0.1.13	172.17.3.1	dyn,nho	Tu1
192.168.1.0/24	10.0.1.11	172.17.1.1	dyn,loc	Tu1
192.168.3.0/24	10.0.1.13	172.17.3.1	dyn,rib	Tu1
192.168.11.0/24	10.0.1.11	172.17.1.1	dyn,loc	Tu1
192.168.13.0/24	10.0.1.13	172.17.3.1	dyn,nho	Tu1

RIB

SpokeA# show ip route next-hop-override

```
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.11/32 is directly connected, Tunnel0
H        10.0.0.13/32 is directly connected, 00:08:40, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
L        10.0.1.11/32 is directly connected, Tunnel1
H        10.0.1.13/32 is directly connected, 00:09:05, Tunnel1
      172.18.0.0/32 is subnetted, 8 subnets
D        172.18.0.1 [90/12800640] via 10.0.0.1, 02:07:25, Tunnel0
D        172.18.0.2 [90/10752640] via 10.0.1.2, 02:07:25, Tunnel1
D        172.18.0.10 [90/13312640] via 10.0.1.2, 02:07:25, Tunnel1
C        172.18.0.11 is directly connected, Loopback0
D  %     172.18.0.13 [90/16384640] via 10.0.1.2, 02:04:46, Tunnel1
             [NHO][90/1] via 10.0.0.13, 00:02:19, Tunnel0
             [NHO][90/1] via 10.0.1.13, 00:08:40, Tunnel1
D     192.168.0.0/21 [90/13312000] via 10.0.1.2, 02:07:25, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.1/32 is directly connected, Ethernet0/0
H     192.168.3.0/24 [250/1] via 10.0.1.13, 00:09:05, Tunnel1
D     192.168.10.0/24 [90/13824000] via 10.0.1.2, 02:04:46, Tunnel1
D     192.168.11.0/24 [90/1536000] via 192.168.1.2, 02:07:25, Ethernet0/0
D  %  192.168.13.0/24 [90/17408000] via 10.0.1.2, 02:04:46, Tunnel1
             [NHO][90/1] via 10.0.1.13, 00:08:59, Tunnel1
```

Summary

Routing Protocol (RP), NHRP and PfRv3

- Routing protocol (RP) – destinations outside of the DMVPN
  - Sets base forwarding for IWAN
  - Set preference for one DMVPN or can setup up ECMP routes
- PfRv3 – optimize forwarding of flows over different DMVPN paths
  - Find paths directly in RP database (except OSPF)
  - PfR RIB forwards flows over paths to MC/BR Loopback next-hop
  - Probe traffic over alternate paths
- NHRP – optimizes forwarding within a single DMVPN
  - Shortcut (spoke-spoke) tunnels
    - Triggered by data traffic and/or PfRv3 probe traffic
    - Use ‘no nhrp route-watch’ to enable shortcut tunnels over alternate paths
    - NHRP mapping/routes to MC/BR Loopback addresses

Agenda

- DMVPN Design Overview
  - DMVPN General
  - IWAN Specific
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- Recent and New Features
  - Configuration, Resiliency, Routing and Forwarding, Centralized Control

Configuration Reduction

- Issue
  - CLI commands need to be configured to recommended values because defaults are not very useful.
- Solution
  - Change CLI command defaults to recommended values
  - Set other CLI commands as default so that they don’t have to be configured at all
  - Derive CLI command values from other parts of the configuration so they don’t have to be configured.

Configuration

- New defaults (IOS/XE 16.3)
  - NHRP
    - Spoke: (ip/ipv6)
      - nhrp holdtime 600
      - nhrp shortcut
      - nhrp registration no-unique
    - Hub: (ip/ipv6)
      - nhrp holdtime 600
      - nhrp map multicast dynamic
      - nhrp max-send 10000 every 10 **(15.5(3)[S,M]2)**

- Future Defaults & Auto-config.
  - NHRP
    - ip/ipv6 nhrp network-id #
      - 1st: tunnel key #
      - 2nd: Interface tunnel#
  - Tunnel Defaults
    - tunnel vrf <tunnel-source-vrf>
  - Miscellaneous Defaults
    - ip mtu
    - ip tcp adjust-mss
    - bandwidth (inherit)

NHRP Original Configuration

Hub

```
interface Tunnel0
   bandwidth 1000
   ip address 10.0.0.1 255.255.255.0
   no ip redirects
   ip mtu 1400
   ip nhrp authentication test
   ip nhrp map multicast dynamic
   ip nhrp map multicast 172.17.0.1
   ip nhrp map 10.0.0.1 172.17.0.1
   ip nhrp network-id 100000
   ip nhrp holdtime 600
   ip nhrp nhs 10.0.0.1
   ip nhrp redirect
   ip tcp adjust-mss 1360
   delay 1000
   tunnel source Serial2/0
   tunnel mode gre multipoint
   tunnel key 100000
   tunnel vrf Outside
   tunnel protection ipsec profile DMVPN
!
```

Spoke

```
interface Tunnel0
   bandwidth 1000
   ip address 10.0.0.11 255.255.255.0
   no ip redirects
   ip mtu 1400
   ip nhrp authentication test
   ip nhrp map multicast 172.17.0.1
   ip nhrp map 10.0.0.1 172.17.0.1
   ip nhrp map multicast 172.17.0.5
   ip nhrp map 10.0.0.2 172.17.0.5
   ip nhrp network-id 100000
   ip nhrp holdtime 600
   ip nhrp nhs 10.0.0.1
   ip nhrp nhs 10.0.0.2
   ip nhrp registration no-unique
   ip nhrp shortcut
   ip tcp adjust-mss 1360
   delay 1000
   tunnel source Serial1/0
   tunnel mode gre multipoint
   tunnel key 100000
   tunnel vrf Outside
   tunnel protection ipsec profile DMVPN
!
```

NHRP NHS Configuration Reduction – IOS 12.4(20)

- Main use of NHRP mapping is to create static mapping for NHS.
- Combine associated NHRP mapping and NHS commands into a single line.
- Can still configure separate NHRP mappings for other purposes.

Hub

```
interface Tunnel0
   ...
   ip nhrp map multicast 172.17.0.1
   ip nhrp map 10.0.0.1 172.17.0.1
   ...
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   ...
!
```

Spoke

```
interface Tunnel0
   ...
   ip nhrp map multicast 172.17.0.1
   ip nhrp map 10.0.0.1 172.17.0.1
   ip nhrp map multicast 172.17.0.5
   ip nhrp map 10.0.0.2 172.17.0.5
   ...
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   ip nhrp nhs 10.0.0.2 nbma 172.17.0.5 multicast
   ...
```

NHRP Configuration New Defaults – IOS/XE 16.3

- Spoke: (ip/ipv6)
  - nhrp holdtime 600
  - nhrp shortcut
  - nhrp registration no-unique
- Hub: (ip/ipv6)
  - nhrp holdtime 600
  - nhrp map multicast dynamic

Hub

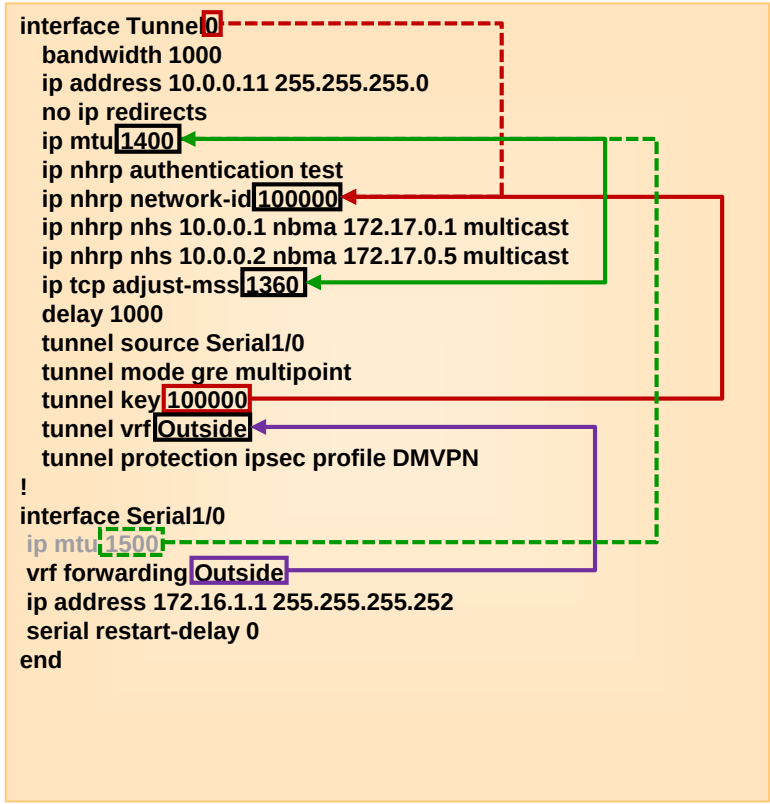
```
interface Tunnel0
   ...
   ip nhrp authentication test
   ip nhrp map multicast dynamic
   ip nhrp network-id 100000
   ip nhrp holdtime 600
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   ip nhrp redirect
   ...
!
```

Spoke

```
interface Tunnel0
   ...
   ip nhrp authentication test
   ip nhrp network-id 100000
   ip nhrp holdtime 600
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   ip nhrp nhs 10.0.0.2 nbma 172.17.0.5 multicast
   ip nhrp registration no-unique
   ip nhrp shortcut
   ...
!
```

Tunnel Configuration Automatic Settings – Future

- NHRP network-id
  - Set to tunnel key <value> if configured
  - Otherwise, set to tunnel interface <#>
- Tunnel VRF
  - Set to VRF of tunnel source interface
- MTU
  - Set to 1400 bytes
    - Use tunnel source <interface> (IPv4/IPv6) MTU – (100/120) bytes
- MSS
  - Set to (IPv4/IPv6) MTU – (40/60) bytes



NHRP Final Configuration

Hub

```
interface Tunnel0
   bandwidth 1000
   ip address 10.0.0.1 255.255.255.0
   ip nhrp authentication test
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   ip nhrp redirect
   delay 1000
   tunnel source Serial2/0
   tunnel mode gre multipoint
   tunnel key 100000
   tunnel protection ipsec profile DMVPN
!
```

Spoke

```
interface Tunnel0
   bandwidth 1000
   ip address 10.0.0.11 255.255.255.0
   ip nhrp authentication test
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   ip nhrp nhs 10.0.0.2 nbma 172.17.0.5 multicast
   delay 1000
   tunnel source Serial1/0
   tunnel mode gre multipoint
   tunnel key 100000
   tunnel protection ipsec profile DMVPN
!
```

Agenda

- DMVPN Design Overview
  - DMVPN General
  - IWAN Specific
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- Recent and New Features
  - Configuration, Resiliency, Routing and Forwarding, Centralized Control

Resiliency

- Issues
  - Many backup NHSs configured, but don't want them all up
  - Quickly failover all spokes to alternate hubs when a hub fails
  - Quickly failover a spoke to alternate hub when spoke-hub tunnel fails

- Solutions
  - Backup and FQDN NHS
  - Fast Hub Failover using BGP (BFD between hubs)
  - BFD over DMVPN (BFD on spoke-hub and spoke-spoke tunnels)

Tunnel Health Monitoring

Interface State – 15.0(1)M

- Issue
  - mGRE tunnel Interface is always “up”
  - Can’t use standard backup/recovery mechanisms
    - backup interface, static interface routes, …

- Solution
  - New Command ‘if-state nhrp’
  - Monitor NHRP registration replies
    - If all NHSs are “down” then set tunnel interface up/down
    - Continue to send NHRP registration requests
    - If a single NHS is “up” then set tunnel interface up/up
  - Combine with ‘backup interface ...’
    - Backup (tunnel) interface only up when main interface is down.

```
interface Tunnel0
  ip address 10.0.0.11 255.255.255.0
  …
  ip nhrp map multicast 172.17.0.1
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp map multicast 172.17.0.5
  ip nhrp map 10.0.0.2 172.17.0.5
  …
  ip nhrp nhs 10.0.0.1
  ip nhrp nhs 10.0.0.2
  …
  if-state nhrp
  …
```

Tunnel Health Monitoring – Interface State (cont.)

```
#show ip nhrp nhs detail
   10.0.0.1  RE req-sent 100  req-failed 0  repl-recv 90 (00:01:38 ago)
   10.0.0.2  RE req-sent 125  req-failed 0  repl-recv 79 (00:01:38 ago)

#show interface tunnel0
   Tunnel0 is up, line protocol is up
   ------------------------------------------------------------------------

*Apr 19 21:32:52 NHRP: NHS-DOWN: 10.0.0.1
*Apr 19 21:32:52 NHRP: NHS 10.0.0.1 Tunnel0 vrf 0 Cluster 0 Priority 0 Transitioned to 'E' from 'RE'
*Apr 19 21:32:53 NHRP: NHS-DOWN: 10.0.0.2
*Apr 19 21:32:53 NHRP: NHS 10.0.0.2 Tunnel0 vrf 0 Cluster 0 Priority 0 Transitioned to 'E' from 'RE'

*Apr 19 21:33:02 %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to down
*Apr 19 21:33:02 NHRP: if_down: Tunnel0 proto IPv4

#show ip nhrp nhs detail
   10.0.0.1   E req-sent 105  req-failed 0  repl-recv 90 (00:02:12 ago)
   10.0.0.2   E req-sent 130  req-failed 0  repl-recv 79 (00:02:12 ago)

#show interface tunnel0
   Tunnel0 is up, line protocol is down
   ------------------------------------------------------------------------

*Apr 19 21:33:12 NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 92
*Apr 19 21:33:13 NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 92
…
*Apr 19 21:34:36 NHRP: NHS 10.0.0.1 Tunnel0 vrf 0 Cluster 0 Priority 0 Transitioned to 'RE' from 'E'
*Apr 19 21:34:36 NHRP: NHS-UP: 10.0.0.1
*Apr 19 21:34:42 %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
*Apr 19 21:34:42 NHRP: if_up: Tunnel0 proto 0

#show ip nhrp nhs detail
   10.0.0.1  RE req-sent 110  req-failed 0  repl-recv 96 (00:00:19 ago)
   10.0.0.2     E req-sent 135  req-failed 0  repl-recv 79 (00:04:09 ago)

#show interface tunnel0
   Tunnel0 is up, line protocol is up
```

Backup and FQDN NHS – 15.1(2)T

- Issue
  - Backup NHSs only needed when primary NHSs are down
  - Backup NHSs can be over subscribed
- Solution
  - Set NHS ‘max-connections’
    - Can set NHS priority (default=0 (best)) – Can have multiple hubs at the same priority
    - Can group NHSs into clusters (default=0) – Separate max-connection value per cluster
    - Configuration reduction – Single line NHS configuration and FQDN NHS
  - Functionality
    - NHSs are brought up in priority order, until cluster max-connections
    - Down NHS at same priority is probed if not at max-connections
    - Down NHS at a lower priority than an active NHS is probed even when max-connections is reached
    - FQDN resolved when bringing up NHS

Backup and FQDN NHS (cont.)

```
interface Tunnel0
   …
   ip nhrp map 10.0.0.1 172.17.0.1
   ip nhrp map multicast 172.17.0.1
   ip nhrp map 10.0.0.2 172.17.0.5
   ip nhrp map multicast 172.17.0.5
   ip nhrp map 10.0.0.3 172.17.0.9
   ip nhrp map multicast 172.17.0.9
   ip nhrp map 10.0.0.4 172.17.0.13
   ip nhrp map multicast 172.17.0.13
   …
   ip nhrp nhs 10.0.0.1
   ip nhrp nhs 10.0.0.2
   ip nhrp nhs 10.0.0.3
   ip nhrp nhs 10.0.0.4
   ip nhrp nhs cluster 0 max-connections 2
   …
```

```
#show ip nhrp
   10.0.0.1/32 via 10.0.0.1   Tunnel0 Type: static, Flags: used
      NBMA address: 172.17.0.1
   10.0.0.2/32 via 10.0.0.2   Tunnel0 Type: static, Flags: used
      NBMA address: 172.17.0.5
   10.0.0.3/32 via 10.0.0.3   Tunnel0 Type: static, Flags: used
      NBMA address: 172.17.0.9                    (no-socket)
   10.0.0.4/32 via 10.0.0.4   Tunnel0 Type: static, Flags: used
      NBMA address: 172.17.0.13                   (no-socket)

#show ip nhrp nhs
   Legend: E=Expecting replies, R=Responding, W=Waiting
   Tunnel0:
   10.0.0.1  RE priority = 0 cluster = 0
   10.0.0.2  RE priority = 0 cluster = 0
   10.0.0.3   W priority = 0 cluster = 0
   10.0.0.4   W priority = 0 cluster = 0
```

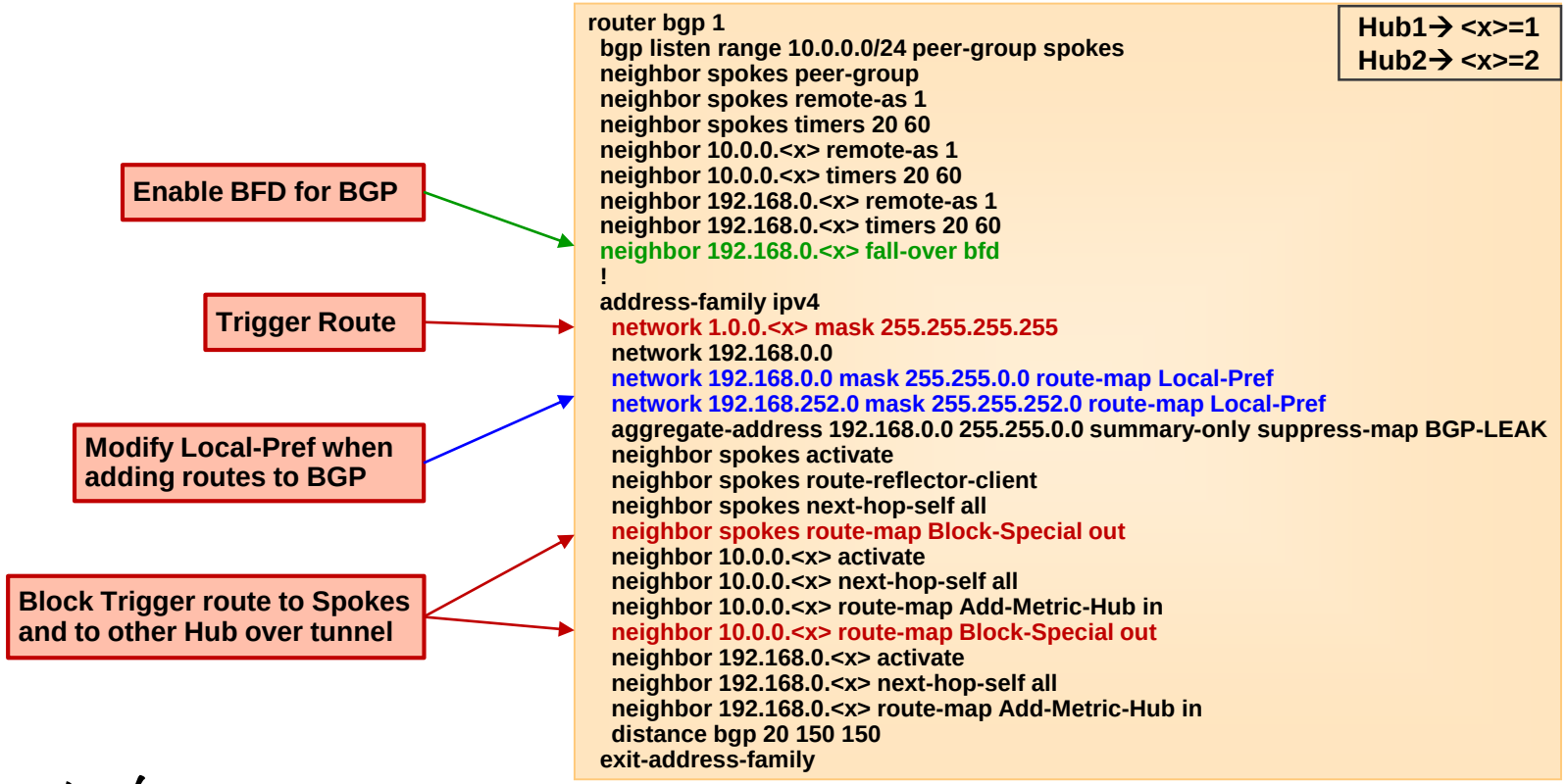
```
interface Tunnel0
   …
   ip nhrp nhs 10.0.0.1 nbma Hub1.cisco.com multicast priority 10 cluster 1
   ip nhrp nhs 10.0.0.2 nbma 172.17.0.5 multicast priority 20 cluster 1
   ip nhrp nhs 10.0.0.3 nbma 172.17.0.9 multicast priority 10 cluster 2
   ip nhrp nhs 10.0.0.4 nbma 172.17.0.13 multicast priority 10 cluster 2
   ip nhrp nhs cluster 1 max-connections 1
   ip nhrp nhs cluster 2 max-connections 1

#show ip nhrp nhs
   Legend: E=Expecting replies, R=Responding, W=Waiting
   Tunnel0:
   10.0.0.1  RE NBMA Address: 172.17.0.1 (Hub1.Cisco.com) priority = 10 cluster = 1
   10.0.0.2  W NBMA Address: 172.17.0.5 priority = 20 cluster = 1
   10.0.0.3  RE NBMA Address: 172.17.0.9 priority = 10 cluster = 2
   10.0.0.4  W NBMA Address: 172.17.0.13 priority = 10 cluster = 2
```

Fast Hub Failover using BGP

- Normal forwarding
  - Few summary routes advertised to spokes
    - Covering all spoke site networks
    - May have separate summary for Hub site networks
    - Use **MED** to load balance or prefer one hub over the other
- Hubs “watch” each other
  - Use BFD on physical link or tunnel link between hubs
  - Special trigger route advertised only between hubs over BFD link
    - Example: 1.0.0.[1,2]/32 on Hub[1,2]
- Hubs “watch” each other (cont.)
  - Track loss of trigger route
    - When lost
      - Install static null0 route with special tag for the summary routes
      - Use BGP route-map to increase the **Local-Pref** on tagged routes
      - Spokes use **Local-Pref** over **MED**
    - Recovery
      - Remove static null0 route with special tag
      - Local-Pref reverts back to normal
      - Spokes go back to using **MED**

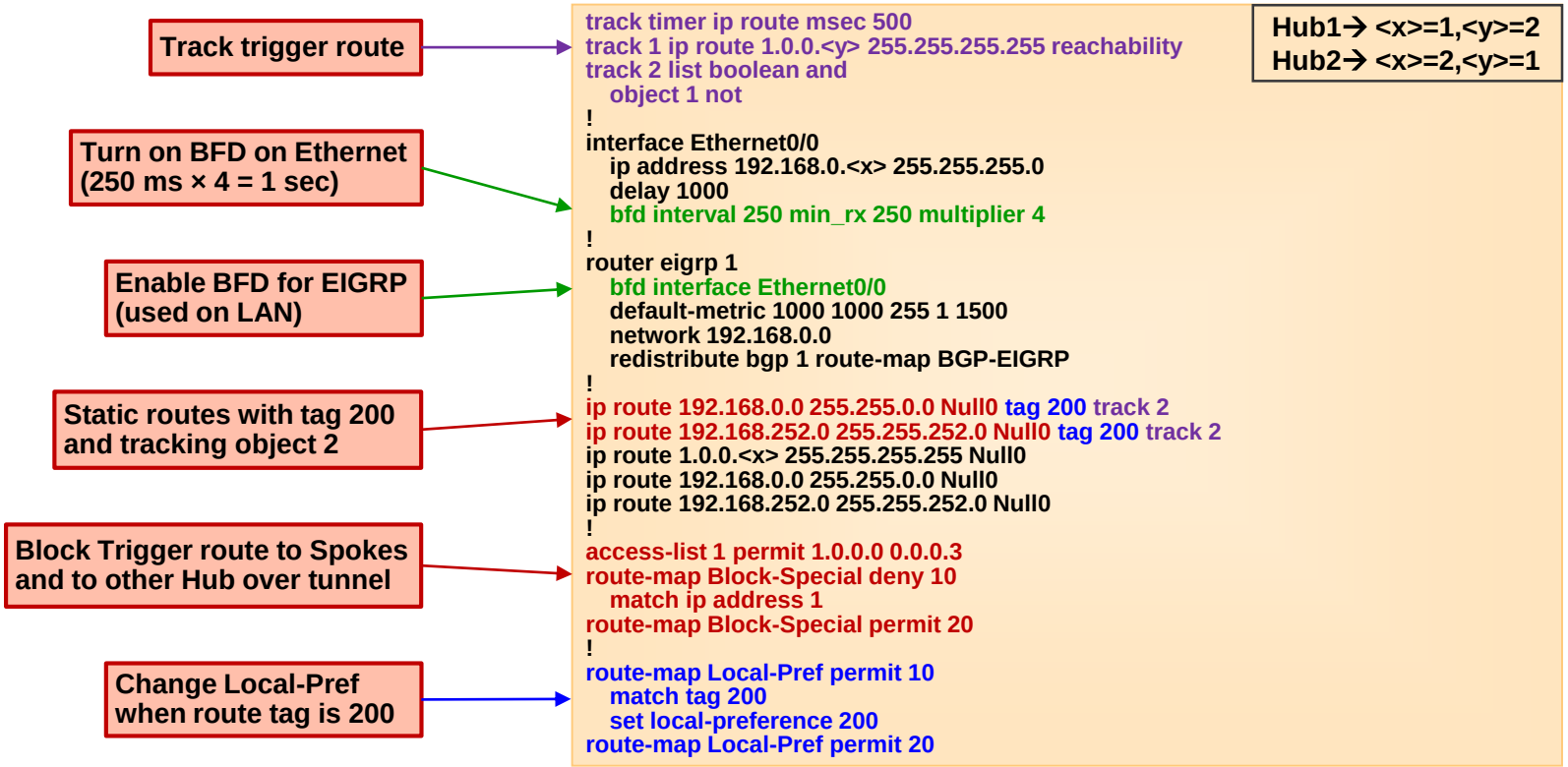
Fast Hub Failover using BGP

BGP Configuration

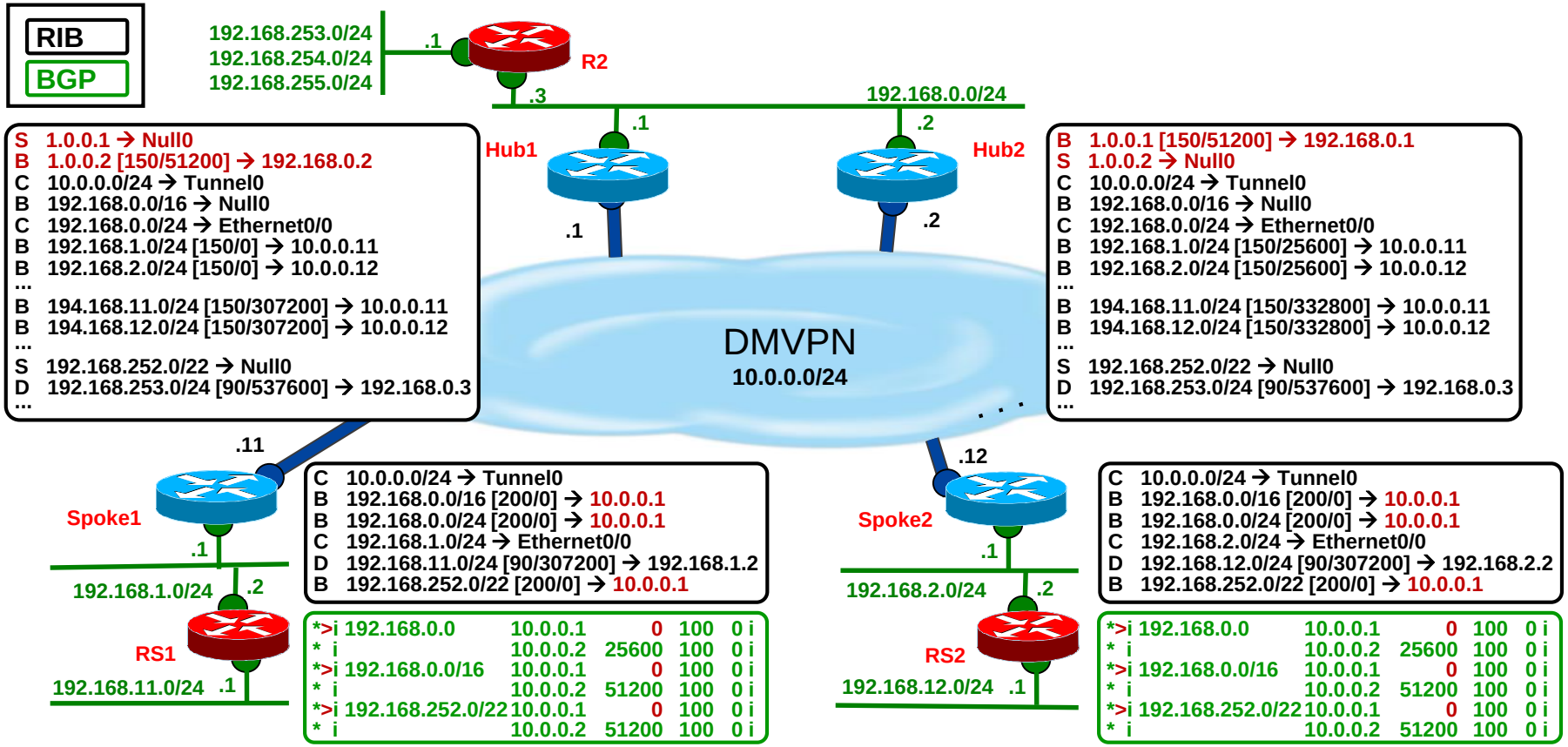


Fast Hub Failover using BGP

Tracking, Route-maps and Routes Configuration

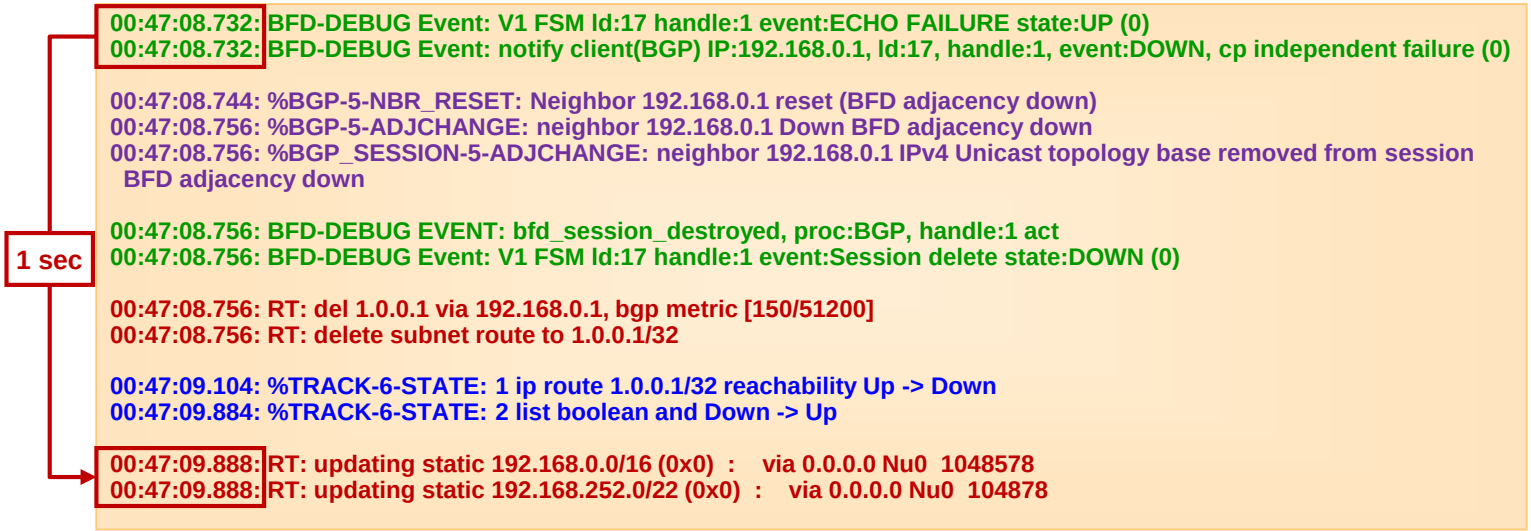


Fast Hub Failover using BGP (normal)



Fast Hub Failover using BGP

Hub2 Debugs



Fast Hub Failover using BGP

Spoke1 Debugs

```
00:47:10.025: RT: updating bgp 192.168.0.0/16 (0x0)  :    via 10.0.0.2   1048577
00:47:10.025: RT: closer admin distance for 192.168.0.0, flushing 1 routes
00:47:10.025: RT: add 192.168.0.0/16 via 10.0.0.2, bgp metric [200/51200]

00:47:10.025: RT: updating bgp 192.168.252.0/22 (0x0)  :    via 10.0.0.2   1048577
00:47:10.025: RT: closer admin distance for 192.168.252.0, flushing 1 routes
00:47:10.025: RT: add 192.168.252.0/22 via 10.0.0.2, bgp metric [200/51200]
```

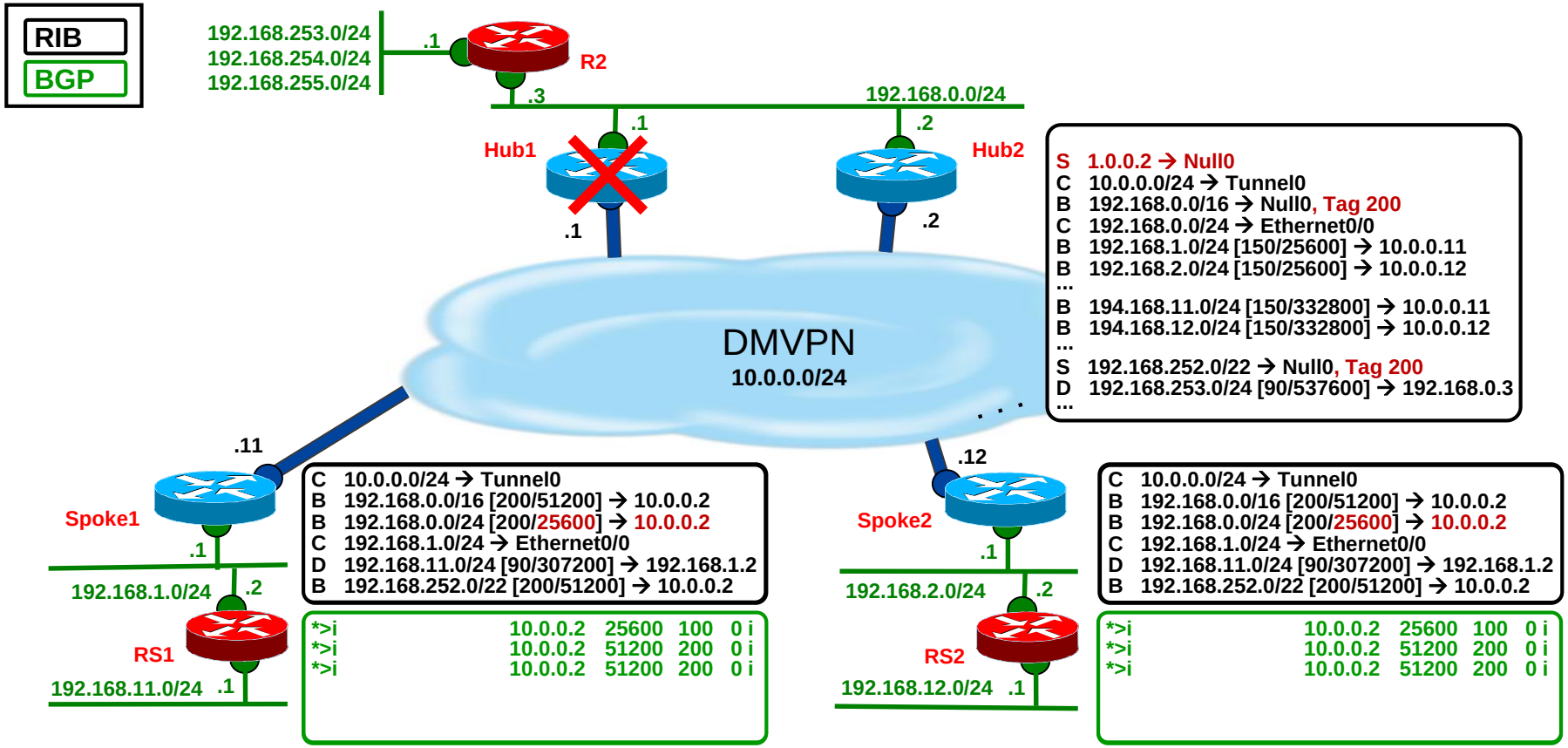
Switch routing to Hub2 (~1.5 secs)

```
00:48:00.725: %BGP-3-NOTIFICATION: sent to neighbor 10.0.0.1 4/0 (hold time expired) 0 bytes
00:48:00.725: %BGP-5-NBR_RESET: Neighbor 10.0.0.1 reset (BGP Notification sent)
00:48:00.725: %BGP-5-ADJCHANGE: neighbor 10.0.0.1 Down BGP Notification sent
00:48:00.725: %BGP_SESSION-5-ADJCHANGE: neighbor 10.0.0.1 IPv4 Unicast topology base removed from session
  BGP Notification sent

00:48:00.725: RT: updating bgp 192.168.0.0/24 (0x0)  :    via 10.0.0.2   1048577
00:48:00.725: RT: closer admin distance for 192.168.0.0, flushing 1 routes
00:48:00.725: RT: add 192.168.0.0/24 via 10.0.0.2, bgp metric [200/25600]
```

BGP Hub1 neighbor down (60 secs)

Fast Hub Failover using BGP (failover)

(after 60 secs)



BFD over DMVPN

16.3.1,
15.7(3)M, 15.6(2)S

- BFD configured on mGRE tunnel interface
  - Use Echo mode
  - BFD maximum probe interval increased to 10 seconds (9999 msec)
  - Spoke-hub tunnel → Only Spoke sends/receives BFD probes*
  - Spoke-spoke tunnel → Both spokes send/receive BFD probes
- NHRP is a BFD client
  - BFD notifies NHRP when tunnel endpoint is down
- NHRP provides a registry for other applications (RP, PfR, IPsec, ...)
  - Applications register with NHRP for a tunnel endpoint (peer, neighbor) address
  - NHRP notifies application when tunnel endpoint is down

*** Currently both Hub and Spoke will send/receive separate BFD probe sets**

```
bfd-template single-hop DMVPN
 interval min-tx 2000 min-rx 2000 multiplier 3
 echo
!
interface Tunnel0
 …
 bfd template DMVPN
 …
```

**Echo mode BFD
2/6 second keepalive/hold**

Apply on Tunnel interface

BFD over DMVPN

BFD session for NHRP static peer (hub)

- If the BFD session is reporting the static peer as down NHRP will:
  - Notify upper layer applications (RP, PFR, …).
  - Initiate NHRP registration requests, if the peer is an NHS
    - If NHRP registration reply is received, peer is up
      - BFD should reflect this state
      - Upper layers should have reset and re-attached to the peer
      - No change in lower layer (IKE/IPsec) crypto session stayed up
    - If NHRP registration reply is not received after 3 retransmissions (~15 seconds)
      - Notify Lower layers (IKE/IPsec) to tear down the crypto session
      - NHRP continues to send registration requests – trigger (IKE/IPsec) crypto session back up
  - Eventually an NHRP registration reply is received
    - The upper layer application sessions (RP and PFR) come back up
  - Note, BFD session is not cleared

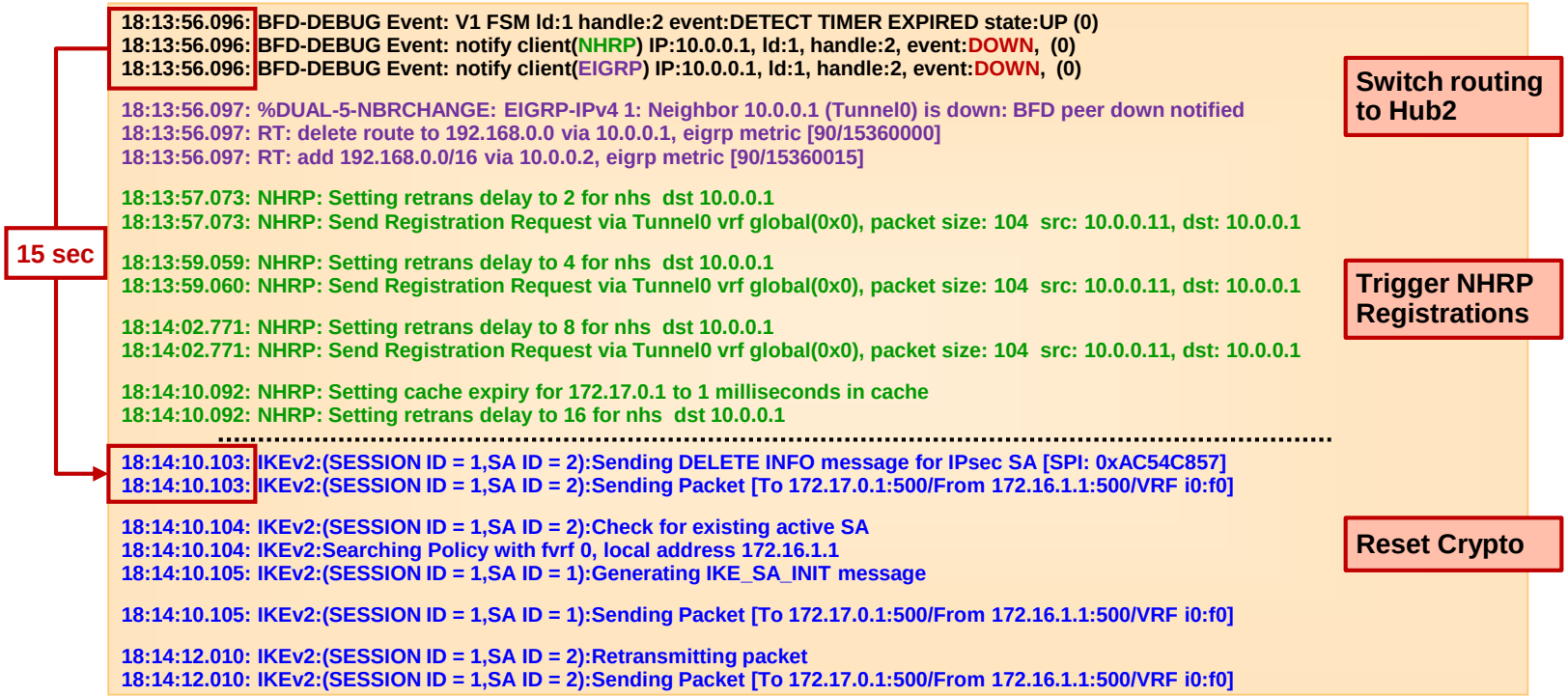
BFD over DMVPN

BFD session for NHRP dynamic peer (spoke)

- If the BFD session is reporting the dynamic peer as down NHRP will:
  - Notify upper layer applications (RP, PFR, …)
  - Notify lower layer applications (IPsec) to clear the crypto session
  - Clear the BFD session, NHRP mapping and associated RIB routes
  - Routing will revert back to spoke-hub-spoke
  - A new spoke-spoke tunnel will be attempted if there is more data traffic
- Detect when spoke-spoke tunnel is no longer used for data packets
  - Use packet count estimates to detect when only BFD probes are using tunnel
  - NHRP mapping times out normally
    - Clear NHRP mapping, BFD session, RIB routes and IKE/IPsec session

BFD over DMVPN

Spoke-Hub tunnel



BFD over DMVPN

Spoke-Spoke tunnel

```
18:46:52.695: NHRP: Receive Traffic Indication via Tunnel0 vrf global(0x0), packet size: 96
18:46:52.705: NHRP: Send Resolution Request for dest: 192.168.12.1 to nexthop: 192.168.12.1 src: 10.0.0.11
18:46:52.784: NHRP: Receive Resolution Request via Tunnel0 vrf global(0x0), packet size: 104
18:46:52.839: %BFD-6-BFD_SESS_CREATED: bfd_session_created, neigh 10.0.0.12 proc:NHRP, idb:Tunnel0 handle:7 act
18:46:52.839: NHRP: Send Resolution Reply via Tunnel0 vrf global(0x0), packet size: 132
18:46:52.875: %BFDFSM-6-BFD_SESS_UP: BFD session ld:2 handle:7 is going UP
18:46:52.875: NHRP: Receive Resolution Reply via Tunnel0 vrf global(0x0), packet size: 132
-------------------------------------------------------------------------------
18:56:52.875: %BFD-6-BFD_SESS_DESTROYED: bfd_session_destroyed,  ld:2 neigh proc:NHRP, handle:7 act
```

Normal tunnel down (no data traffic) (10 min)

```
19:19:04.622: NHRP: Receive Traffic Indication via Tunnel0 vrf global(0x0), packet size: 96
19:19:04.632: NHRP: Send Resolution Request for dest: 192.168.12.1 to nexthop: 192.168.12.1 using our src: 10.0.0.11
19:19:04.703: NHRP: Receive Resolution Request via Tunnel0 vrf global(0x0), packet size: 104
19:19:04.734: %BFD-6-BFD_SESS_CREATED: bfd_session_created, neigh 10.0.0.12 proc:NHRP, idb:Tunnel0 handle:7 act
19:19:04.734: NHRP: Send Resolution Reply via Tunnel0 vrf global(0x0), packet size: 132
19:19:04.771: NHRP: Receive Resolution Reply via Tunnel0 vrf global(0x0), packet size: 132
19:19:04.782: %BFDFSM-6-BFD_SESS_UP: BFD session ld:10 handle:7 is going UP
-------------------------------------------------------------------------------
19:19:24.209: %BFDFSM-6-BFD_SESS_DOWN: BFD session ld:10 handle:7,is going Down Reason: DETECT TIMER EXPIRED
19:19:24.209: BFD-DEBUG Event: notify client(NHRP) IP:10.0.0.12, ld:10, handle:7, event:DOWN,  (0)
19:19:24.211: NHRP: Calling for delete of Tunnel Endpoints (VPN: 10.0.0.12, NBMA: 172.16.2.1)
19:19:24.211: %BFD-6-BFD_SESS_DESTROYED: bfd_session_destroyed,  ld:10 neigh proc:NHRP, handle:7 act

19:19:24.800: NHRP: Receive Traffic Indication via Tunnel0 vrf global(0x0), packet size: 96
```

Abnormal tunnel down (BFD triggered) (20 sec)

Agenda

- DMVPN Design Overview
  - DMVPN General
  - IWAN Specific
- NHRP Details
  - NHRP Overview
  - NHRP Registrations/Resolutions/Redirects
- Recent and New Features
  - Configuration, Resiliency, Routing and Forwarding, Centralized Control

NHRP routes and routing

- Issues
  - Can’t control NHRP short-cut routes on spokes
    - Can’t prefer/order routes using multiple short-cut tunnels
    - Can’t summarize NHRP short-cut routes like you can with RP routes
  - Routing protocol limits scale of DMVPN on hubs (IoT)
  - Need separate DMVPN hub router per Cloud (Transport)
- Solutions
  - NHRP Route Metric control
  - NHRP Route Summarization
  - NHRP Route Advertisement
  - Multiple Tunnel Termination (MTT) on Hub routers

Controlling NHRP routes

15.5(3)S&M,15.6(2)T

NHRP route metric control per tunnel interface

- Egress Load-balancing or Ingress traffic engineering
- Peer NHS path preference used to calculate NHRP route metric
- Preference value is (1-255); best = 1; default = 255
- NHRP route metric = (255^2 = 65025)/preference
  - Examples: (preference = 16 → metric = 4064); (preference = 255 → metric = 255)
- Strict preference: (p1 > 16×p2)
  - Example: (p1=32, p2=1) → 2032/65025 = 1/32
- Unequal load-balancing: (p1 ≤ 16×p2)
  - Example: (p1=16, p2=4) → 4064/16256 = 1/4

```
interface Tunnel0
 ip address 10.0.0.11 ...
 …
 ip nhrp path preference 16
 …
```

```
# show ip route nhrp
...
    192.168.11.0/27 is subnetted, 1 subnets
H       192.168.11.32 [250/4064] via 10.0.0.11, 00:00:09, Tunnel0
```

```
#show ip nhrp
...
192.168.11.0/24 via 10.0.0.11
  Tunnel0 created 00:01:46, expire 00:08:13
  Type: dynamic, Flags: router rib
  NBMA address: 172.16.1.1
```

Controlling NHRP routes

15.5(3)S, 15.5(3)M

NHRP summarization

- Current Behavior
  - NHRP answers resolution request with most specific RIB network/mask
  - Ability to summarize NHRP mappings and routes like RP routes
    - ip nhrp summary-map { *network/mask-length* | *network mask* }
    - Used in resolution responses instead of matching RIB network/mask*
    - Similar to a summary route for a Routing Protocol
- Use Cases
  - Summary of spoke subnets for NHRP resolution replies
  - Fixes 1st subnet of summary route use at spoke, spoke-spoke refresh issue
  - Default (0/0) → NHRP /32 resolution replies mitigation rather than static routes

*** NHRP summary network/mask used even if more specific than RIB network/mask**

```
interface Tunnel0
 ip nhrp summary-map 192.168.11.0/24
```

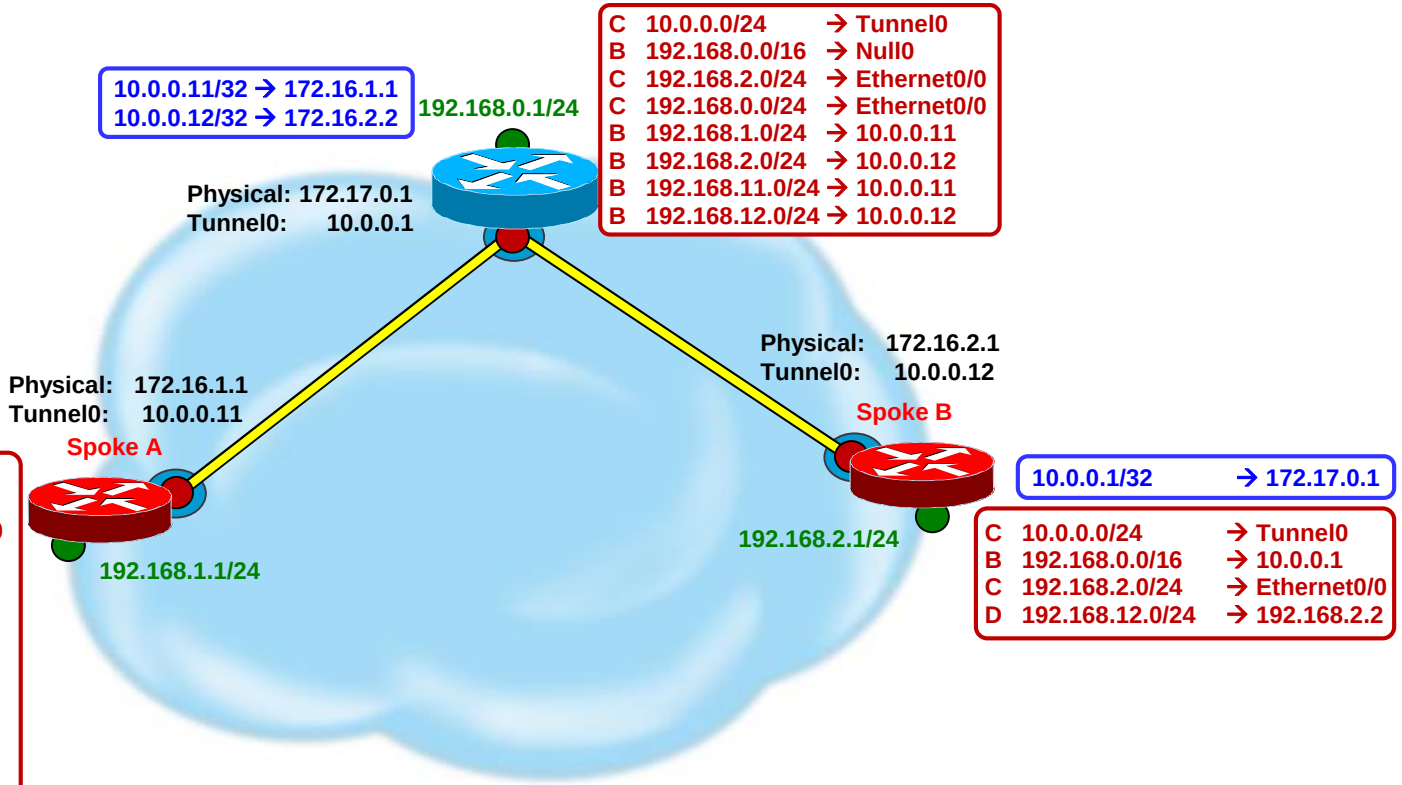
```
#show ip nhrp 192.168.11.0

192.168.11.0/24 via 10.0.0.11
  Tunnel0 created 00:00:07, never expire
  Type: static, Flags: local
  NBMA address: 172.16.1.1 (no-socket)
```

NHRP Summary Map

Original Setup

NHRP mapping

RIB Table

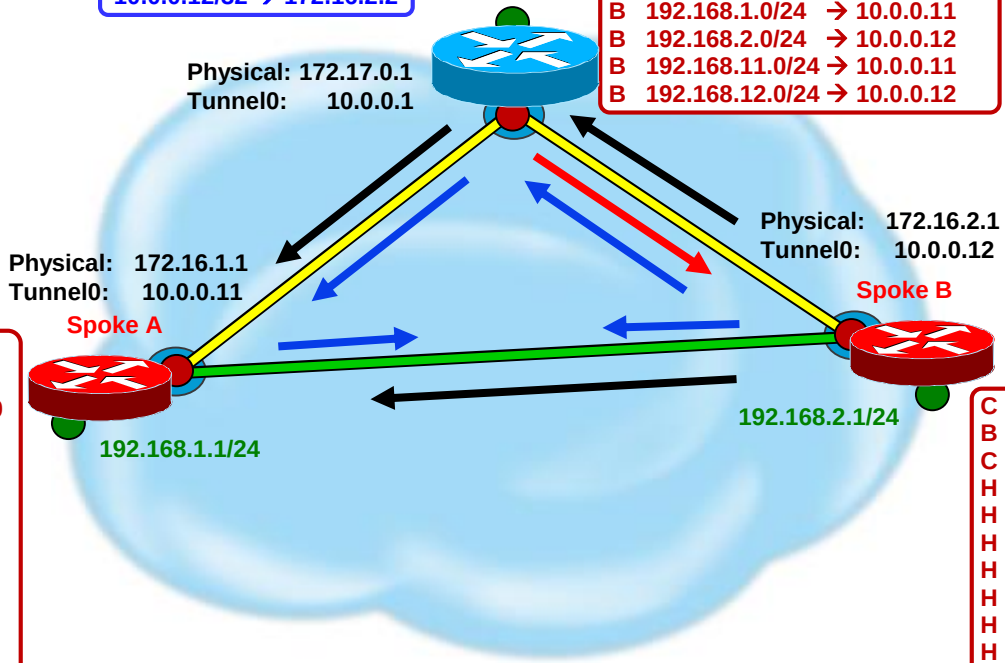


NHRP Summary Map

Without Summary-Map

NHRP mapping

RIB Table



10.0.0.1/32	→	172.17.0.1
192.168.11.0/27	→	172.16.1.1 (I)
192.168.11.32/27	→	172.16.1.1 (I)
192.168.11.64/27	→	172.16.1.1 (I)
192.168.11.96/27	→	172.16.1.1 (I)
192.168.11.128/27	→	172.16.1.1 (I)
192.168.11.160/27	→	172.16.1.1 (I)
192.168.11.192/27	→	172.16.1.1 (I)
192.168.11.224/27	→	172.16.1.1 (I)

C	10.0.0.0/24	→	Tunnel0
B	192.168.0.0/16	→	10.0.0.1
C	192.168.1.0/24	→	Ethernet0/0
B	192.168.11.0/24	→	Null0
D	192.168.11.0/27	→	192.168.1.2
D	192.168.11.32/27	→	192.168.1.2
D	192.168.11.64/27	→	192.168.1.2
D	192.168.11.96/27	→	192.168.1.2
D	192.168.11.128/27	→	192.168.1.2
D	192.168.11.160/27	→	192.168.1.2
D	192.168.11.192/27	→	192.168.1.2
D	192.168.11.224/27	→	192.168.1.2

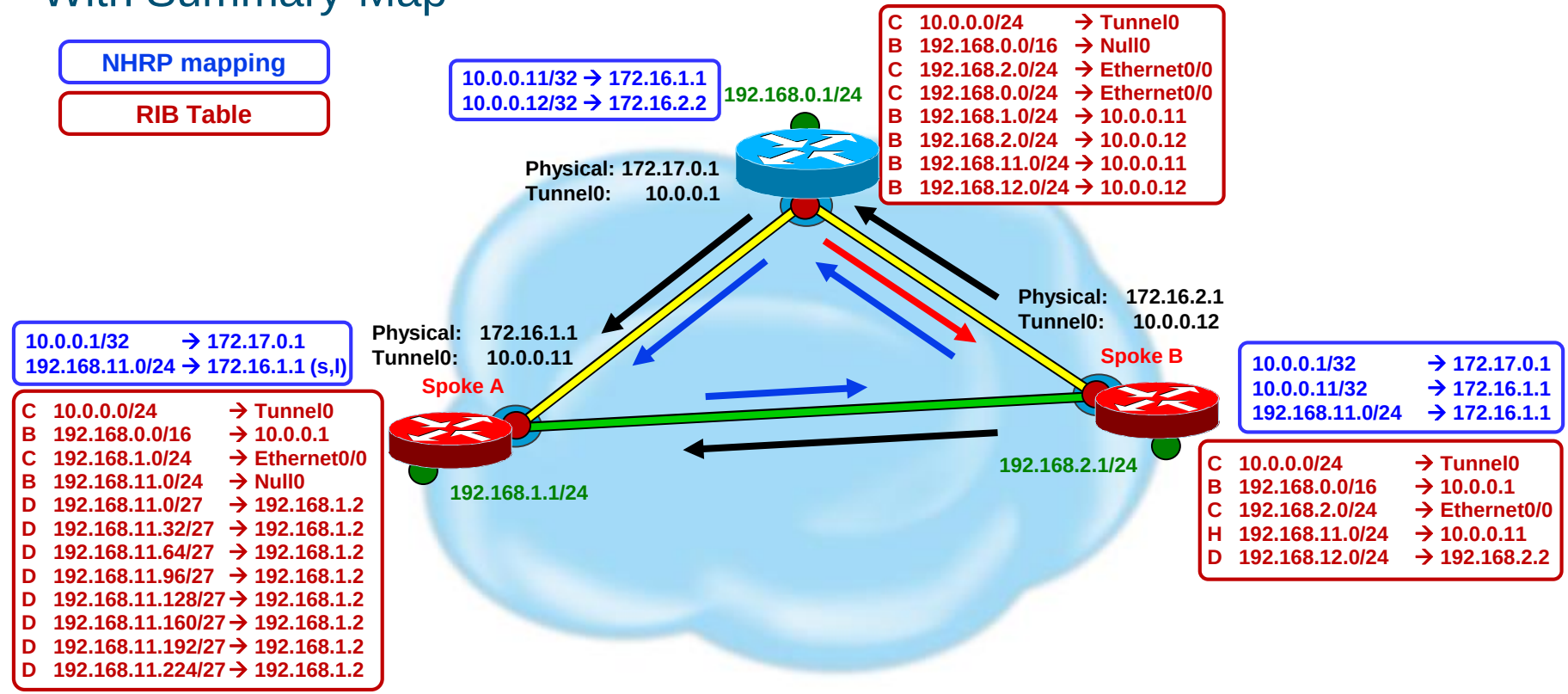
C	10.0.0.0/24	→	Tunnel0
B	192.168.0.0/16	→	Null0
C	192.168.2.0/24	→	Ethernet0/0
C	192.168.0.0/24	→	Ethernet0/0
B	192.168.1.0/24	→	10.0.0.11
B	192.168.2.0/24	→	10.0.0.12
B	192.168.11.0/24	→	10.0.0.11
B	192.168.12.0/24	→	10.0.0.12

10.0.0.1/32	→	172.17.0.1
10.0.0.11/32	→	172.16.1.1
192.168.11.0/27	→	172.16.1.1
192.168.11.32/27	→	172.16.1.1
192.168.11.64/27	→	172.16.1.1
192.168.11.96/27	→	172.16.1.1
192.168.11.128/27	→	172.16.1.1
192.168.11.160/27	→	172.16.1.1
192.168.11.192/27	→	172.16.1.1
192.168.11.224/27	→	172.16.1.1

C	10.0.0.0/24	→	Tunnel0
B	192.168.0.0/16	→	10.0.0.1
C	192.168.2.0/24	→	Ethernet0/0
H	192.168.11.0/27	→	10.0.0.11
H	192.168.11.32/27	→	10.0.0.11
H	192.168.11.64/27	→	10.0.0.11
H	192.168.11.96/27	→	10.0.0.11
H	192.168.11.128/27	→	10.0.0.11
H	192.168.11.160/27	→	10.0.0.11
H	192.168.11.192/27	→	10.0.0.11
H	192.168.11.224/27	→	10.0.0.11
D	192.168.12.0/24	→	192.168.2.2

NHRP Summary Map

With Summary-Map



NHRP Route Advertisement

- Route advertisement between hub and spoke in NHRP registration message
- Ability to redistribute routes between NHRP and other routing protocols
  - redistribute nhrp ...
- Control NHRP routing using standard 'router nhrp ...' CLI construct
- Not a replacement for regular routing protocols (EIGRP, BGP, ...)
  - RPs handle much more complex networks
- For simple hub-spoke and spoke-spoke DMVPNs (IoT)
  - 10,000s small spoke sites with one or few subnets
  - 100,000s of spokes sites in hub-spoke IoT networks
  - Preliminary scaling to 10,000 → 30,000+ spokes per hub (CSR)

Routing Protocol Features – BGP

- iBGP Local-AS (15.2(2)T, 15.1(3)S (CSCtj48063))
  - Run iBGP over DMVPN
    - Tunnel end-point routers may have different native BGP ASs
    - Allows 'neighbor ... local-as #' and 'neighbor ... remote-as #' to be the same (iBGP)
    - 'neighbor ... local-as #' is different from local native BGP AS, 'router bgp #'
      - Almost like eBGP within the router between the native AS and the AS over DMVPN
    - BGP Dynamic Neighbors to reduce configuration on hub
      - Added IPv6 Dynamic Neighbor support in 16.3, 15.6(3)M

```
router bgp 65000
 bgp listen range 10.0.0.0/24 peer-group spokes
 ...
 neighbor spokes peer-group
 neighbor spokes remote-as 65001
 neighbor spokes local-as 65001
 ...
```

BGP Dynamic Neighbors

iBGP Local-AS

Routing Protocol Features – EIGRP

- Equal Cost MultiPath (15.2(3)T, 15.2(1)S (CSCsj31328))
  - Destination network is reachable via more than one DMVPN (mGRE tunnel) and the ip next-hop needs to be preserved (Phase 2).

```
no ip next-hop-self eigrp <as> [no-ecmp-mode]
```

- Add-path (15.3(1)S (CSCtw86791))
  - Spoke site has multiple DMVPN spoke routers and want to be able to load-balance spoke-spoke tunnels (Phase 2).
    - **Requires new "named" EIGRP router configuration**

```
router eigrp <name>
  address-family ipv4 unicast autonomous-system 1
    af-interface Tunnel0
      no next-hop-self
      add-path <paths>  (<paths> = number of extra paths)
      no split-horizon
    ...
```

Agenda

- DMVPN Design Overview
  - DMVPN General
  - IWAN Specific
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
- Recent and New Features
  - Configuration, Resiliency, Routing and Forwarding, Centralized Control

Centralized Control

Separating Control and Data Planes

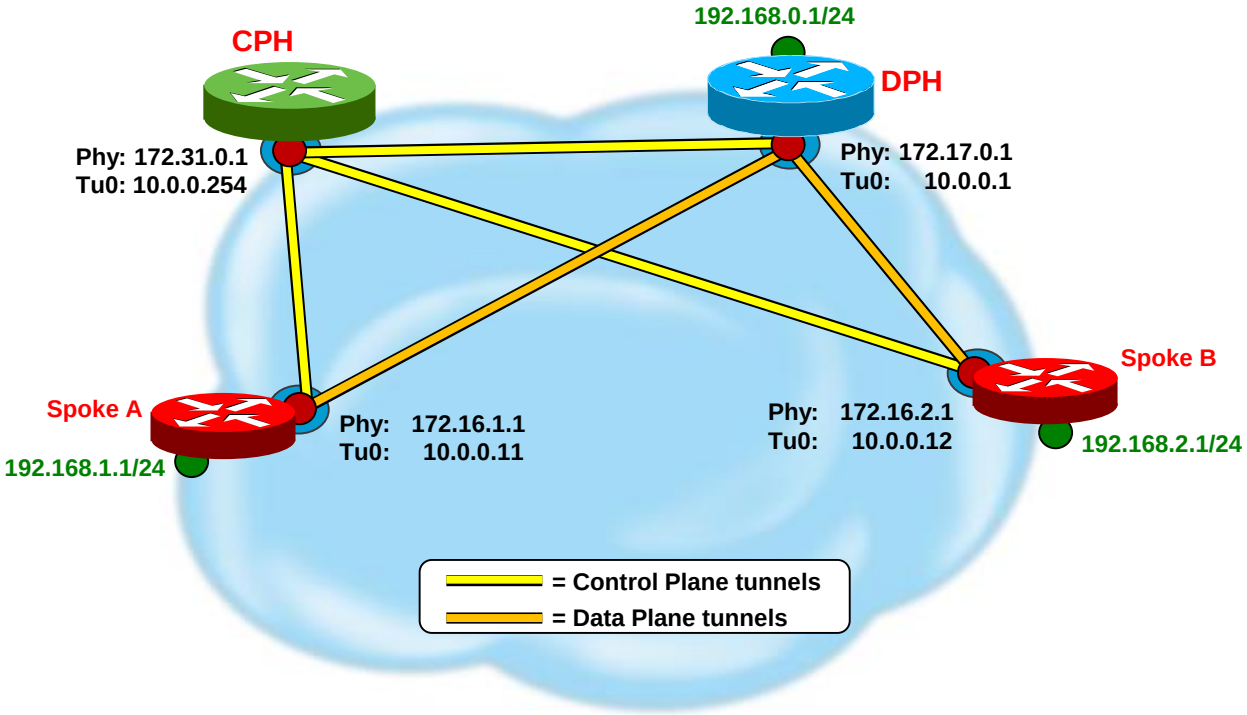
- Issues – Converged control and data planes on hubs
  - Routing Protocol scaling limits scale of DMVPN hubs
  - ISP managed DMVPN – want Hubs in ISP network
    - Data traffic traverses DMVPN hubs while short-cut tunnel is built
    - Multicast traffic (replication*) goes through DMVPN hubs
- Solution – Separate control plane from data plane
  - Separate DMVPN Control Plane Hub (CPH) and Data Plane Hub (DPH)
    - Scale CPH and DPH independently of each other
    - Data traffic only ever goes through DPHs never CPH
  - Other central control services at CPH
    - Routing Protocol (BGP, EIGRP*); Key Management (ESON*)
- ISP managed DMVPN → CPH in ISP network; DPHs in customer network

Centralized Routing and NHS

Separating the Control and Data planes

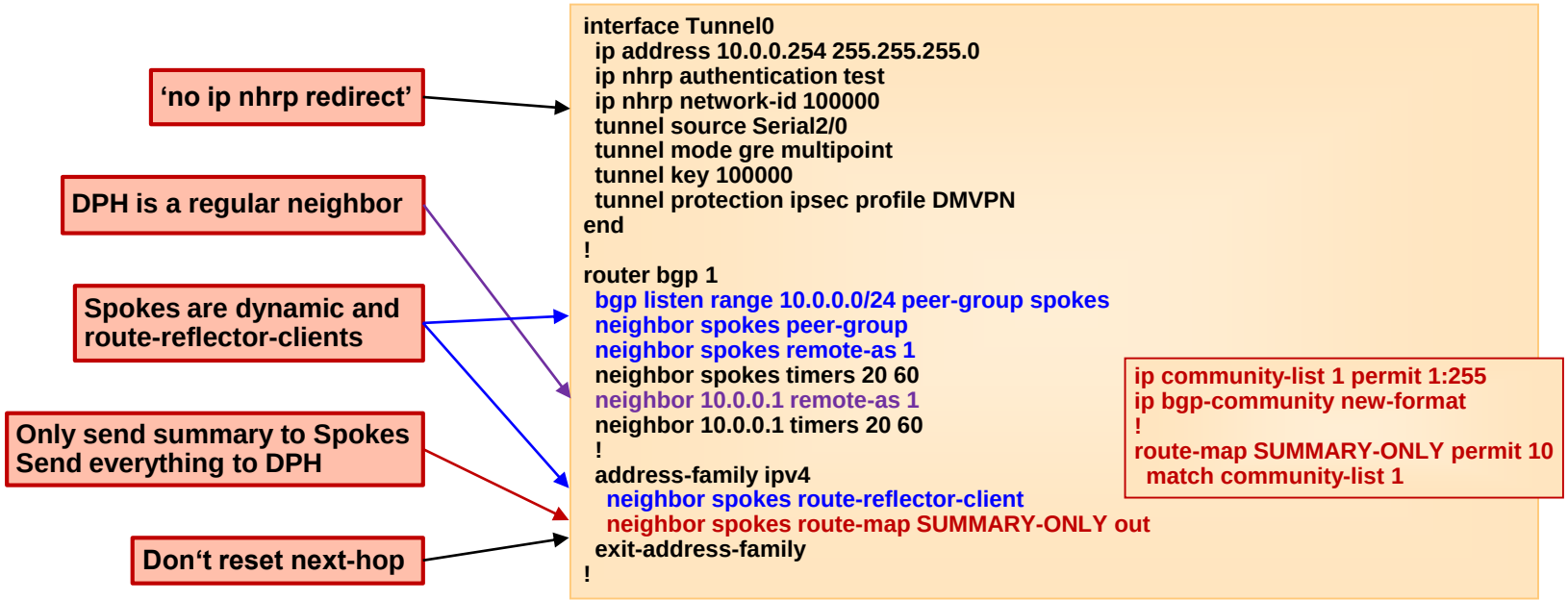
- Control Plane Hub (CPH)
  - Routing – peer with DPHs and spokes
    - iBGP (route-reflector), in future EIGRP (OTP route-reflector)
  - NHS → NHRP registrations and resolution request processing
  - Future:
    - Smart data plane hub selection pushed to spokes
    - Optional Centralized Key Server (ESON)
- Data Plane Hub (DPH)
  - Provide data path for spoke-hub-spoke
  - Routing – peer with CPH
    - Advertise network and/or regional summaries to CPH
  - NHS → NHRP redirect; NHRP registrations (for no-drop); Backup CPH

Centralized Routing and NHS



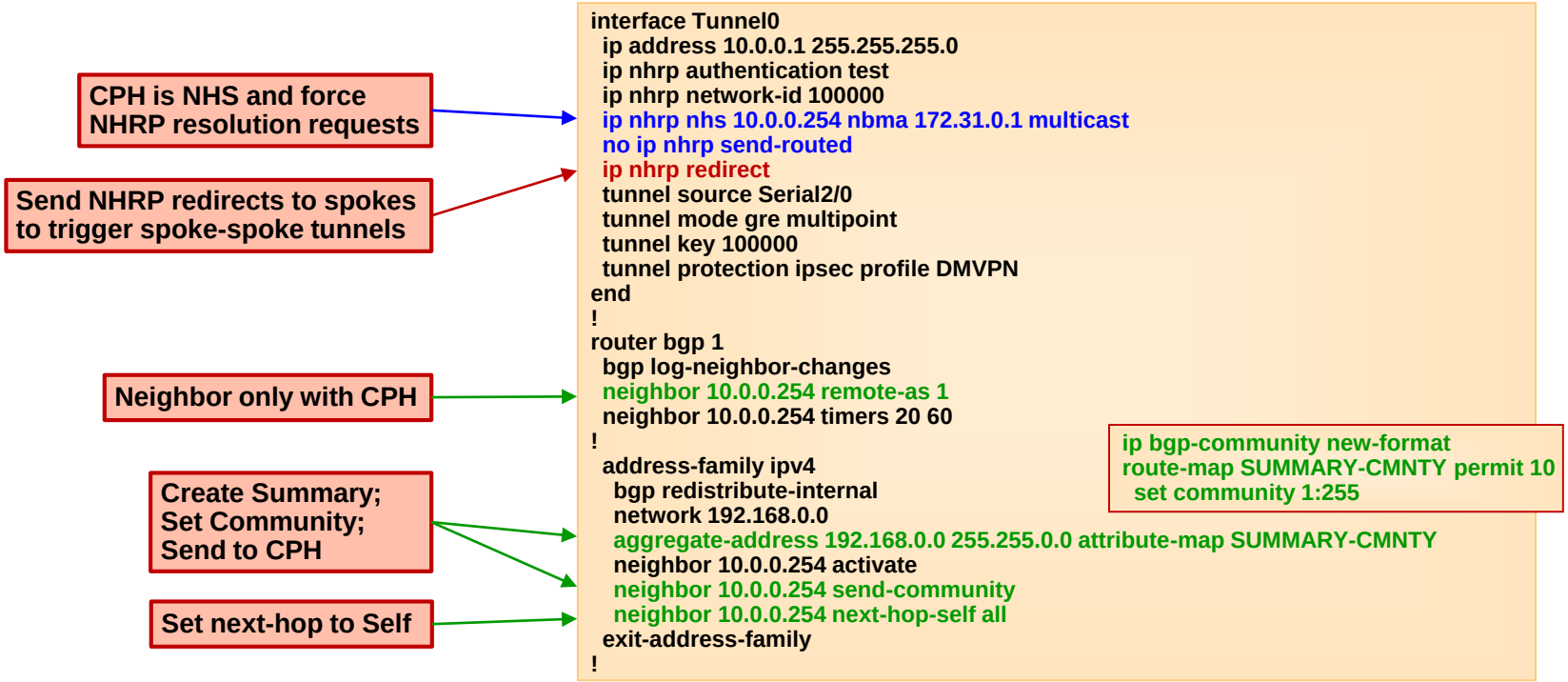
Centralized Routing and NHS

CPH Configuration



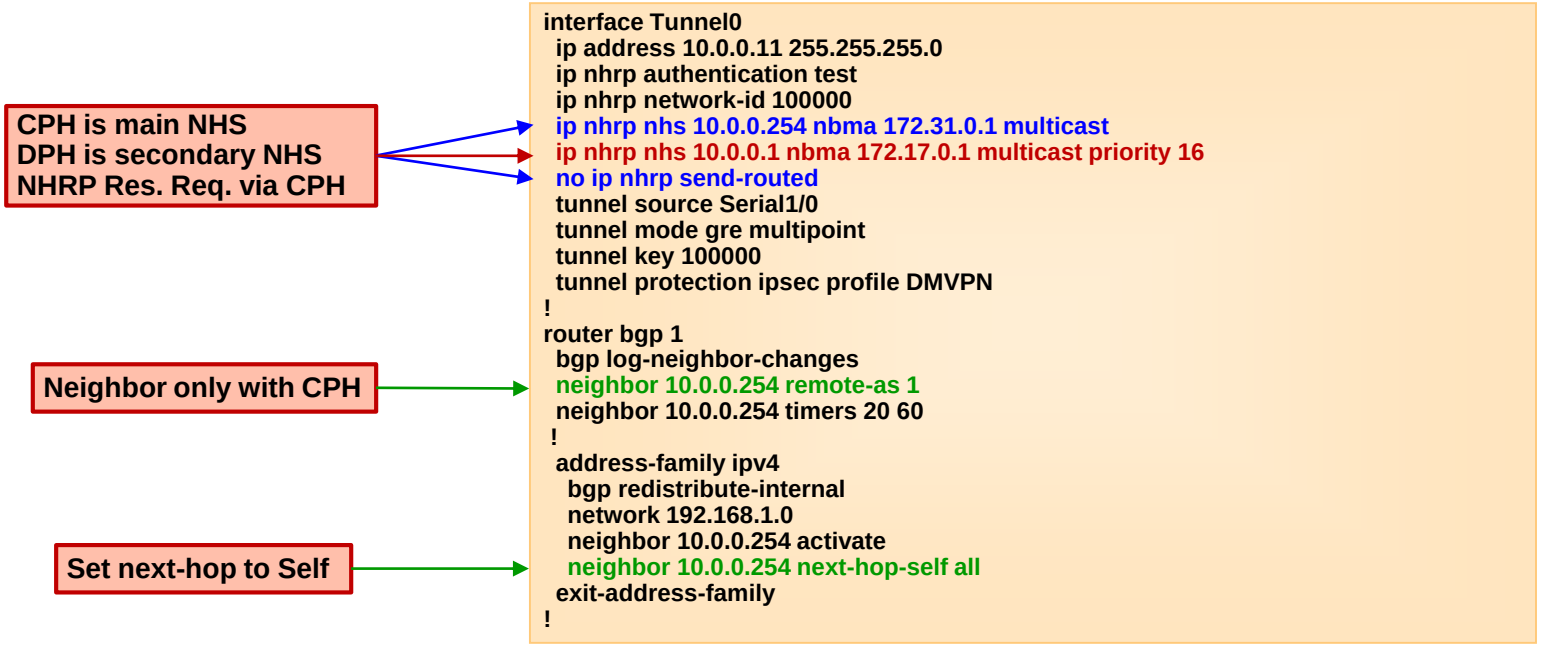
Centralized Routing and NHS

DPH Configuration



Centralized Routing and NHS

Spoke Configuration

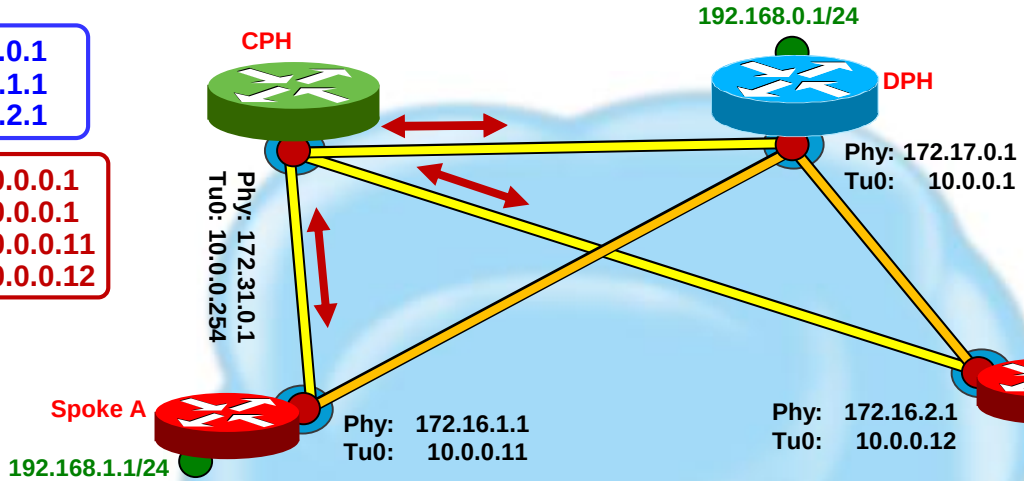


Centralized Routing and NHS

NHRP Registration and Routing

NHRP mapping

Routing Table



10.0.0.1 → 172.17.0.1
10.0.0.11 → 172.16.1.1
10.0.0.12 → 172.16.2.1

192.168.0.0/16 → 10.0.0.1
192.168.0.0/24 → 10.0.0.1
192.168.1.0/24 → 10.0.0.11
192.168.2.0/24 → 10.0.0.12

10.0.0.11 → 172.16.1.1
10.0.0.12 → 172.16.2.1
10.0.0.254 → 172.31.0.1

192.168.0.0/16 → Null0
192.168.0.0/24 → Ethernet0/0
192.168.1.0/24 → 10.0.0.11
192.168.2.0/24 → 10.0.0.12

10.0.0.1 → 172.17.0.1
10.0.0.254 → 172.31.0.1

192.168.0.0/16 → 10.0.0.1
192.168.1.0/24 → Ethernet0/0

10.0.0.1 → 172.17.0.1
10.0.0.254 → 172.31.0.1

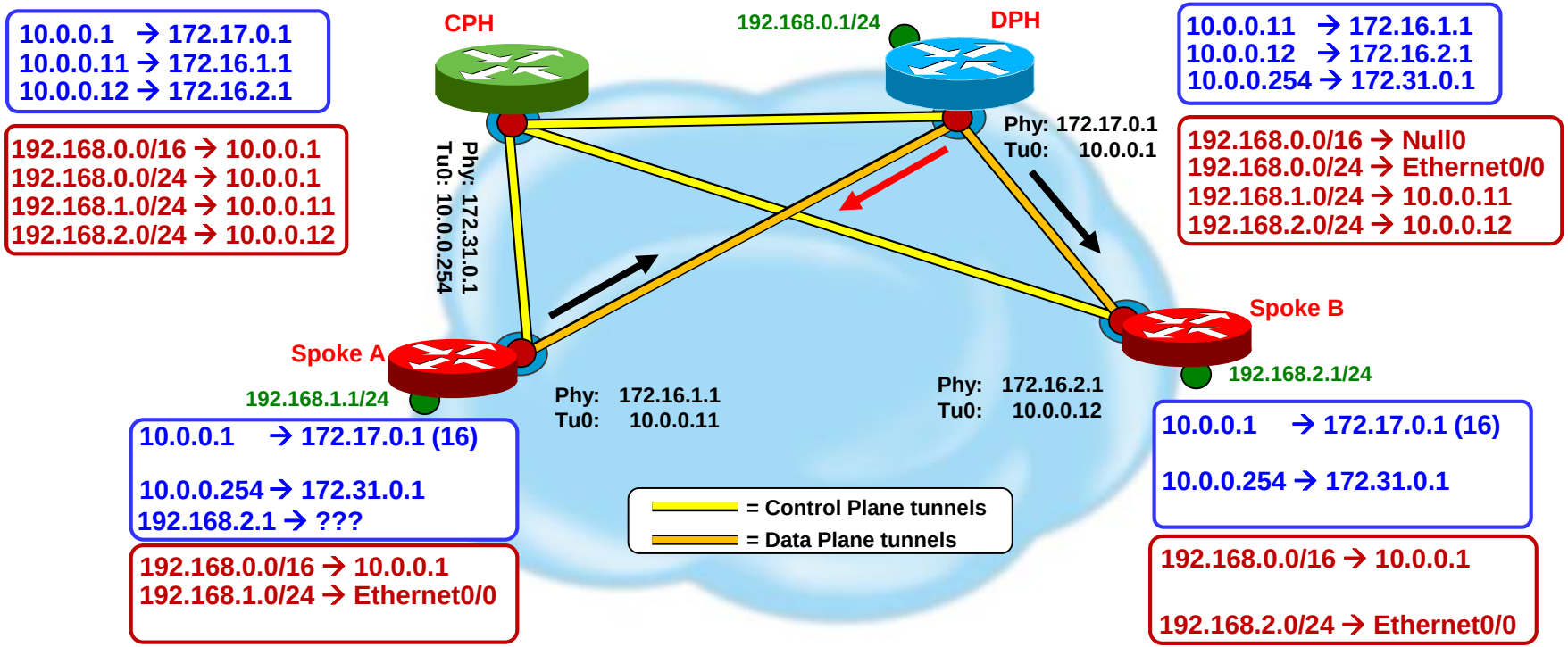
192.168.0.0/16 → 10.0.0.1
192.168.2.0/24 → Ethernet0/0

Centralized Routing and NHS

Data packets via DPH

NHRP mapping

Routing Table

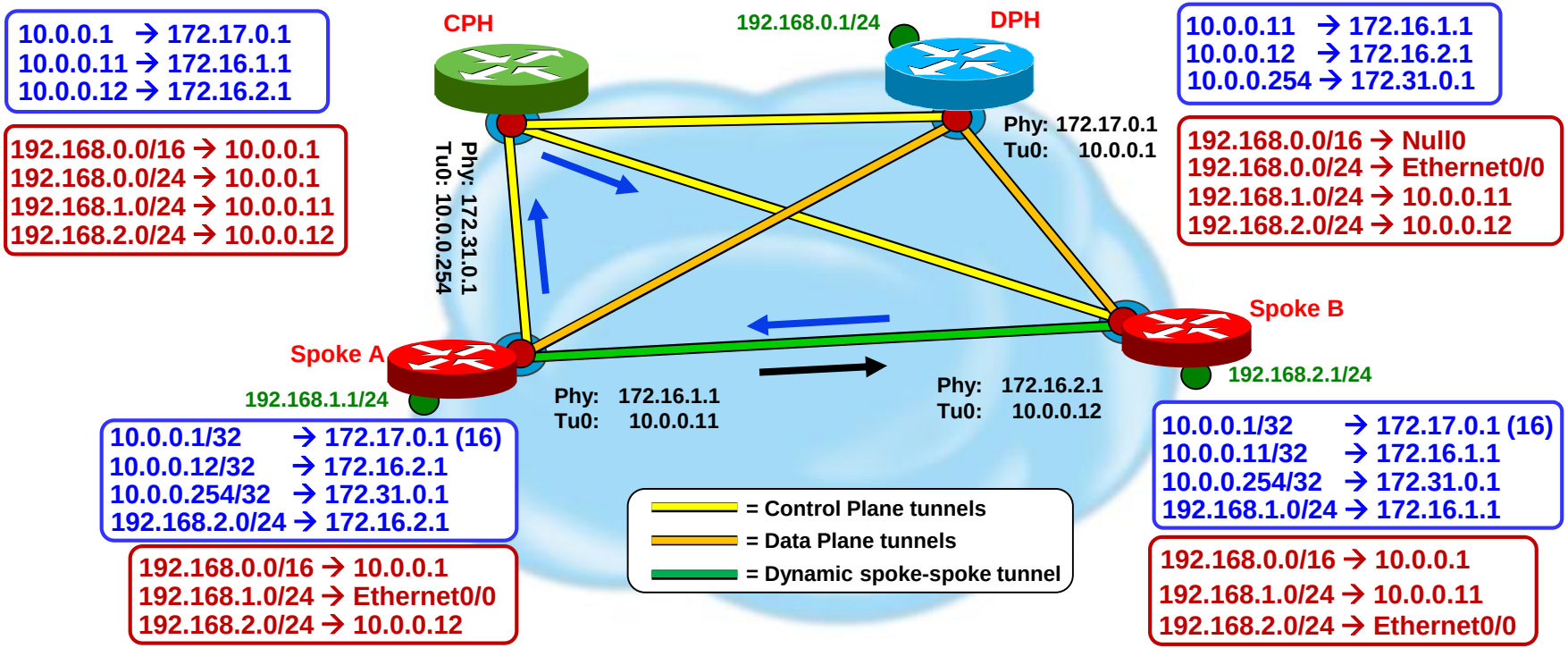


Centralized Routing and NHS

Control packets via CPH

NHRP mapping

Routing Table



Centralized Routing and NHS

- Summary
  - Separation of Control and Data Planes
    - ISP Managed DMVPN Service (CPH in ISP network, DPHs in customer network)
    - Separate scaling for CPH (RP peers) and DPH (Encryption throughput)
    - Uses the same DMVPN/mGRE infrastructure
      - Main NHS at CPH, Natural backup NHS at DPH
- Future
  - Download from CPH to spokes, NHS summary-map configuration for DPH

    {ip | ipv6} nhrp summary-map {all-routed [*nbma*] | *prefix* [[*nbma* [preference *pref*]]} [multicast] [resolve]
    [match {group *group_name* | geo-location *geo-location* | topo-location *topo-location* | attribute *attr_type attr_value*}]

    - All-routed: RP advertises summary → temporary map to use NBMA as DPH
    - Prefix: Default/summary prefix passed to spokes
    - Resolve: Prefix is specified, but not NBMA → forces resolution for all packets; hub-less model
    - Match: Push different summary maps depending on attributes from spoke registration to CPH

Centralized Control

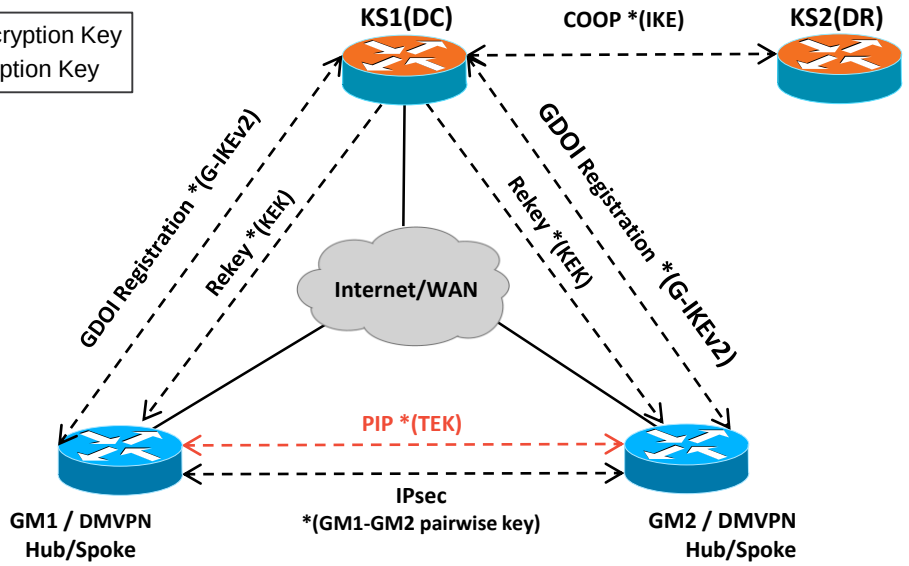
Extensible Security for Overlay Network (ESON)

- A Centralized Key Server Solution with pairwise key capability.
  - Centralized management of policy & pairwise and group keys for IPsec overlay VPNs
  - Leverages GetVPN control plane (GDOI/G-IKEv2) as underlying infrastructure
    - **GM-KS:** G-IKEv2 Registrations for initial pull of policy & **keying material**
    - **KS-GM:** KS pushes periodic rekeys (unicast/multicast)
    - **KS-KS:** Multiple KSs for redundancy using COOP over IKEv2
- IKEv2 is not used between GMs (no Diffee Hellman (DH))
- Peer Introduction Protocol (PIP) is lightweight control plane (2 messages) between GMs
  - **GM-GM:** Exchanges cryptographic identities and nonces for pair-wise key generation and detects NAT between Peers

DMVPN with ESON

G-IKEv2 based centralized management of pairwise and group IPsec session keys

TEK: Traffic Encryption Key
KEK: Key Encryption Key



Group Keys: TEK, KEK
GM1 Key Material
GM1 Identity
GM1-GM2 Pairwise key

Group Keys: TEK, KEK
GM2 Key Material
GM2 Identity
GM1-GM2 Pairwise key

Control Plane

GM – KS
- G-IKEv2 (KEK, TEK and key material from KS)

GM – GM
- PIP*(TEK): (Encrypted with TEK key)

Control Plane Redundancy

KS – KS
- COOP over IKEv2

Data Plane

GM – GM
- IPsec *(GM1-GM2 Pairwise key)

Data Plane Redundancy

GM – GM
- Redundant Hubs

DMVPN with ESON - Value Proposition

- Centralized key server and management
  - Centralized authentication & authorization of GMs (DMVPN Hub/spoke)
  - Centralized management of crypto policy and keys
  - Crypto Control-plane/Data-plane separation, no IKEv2 or DH between GMs
- Easier to manage
- Elasticity of scale; Reduced setup latency; Virtualized Key Server
- Faster & more effective removal of compromised GMs
- Better enforcement of enterprise security policy & centralized trust management
- Allows varying key management schemes
  - Group keys: Control Plane (PIP); Data Plane (Native Multicast)
  - Pairwise keys for better security – Data Plane (Unicast)
  - Various rekey policies/schemes are possible

IKEv2 with DMPVN

- DMVPN works with ISAKMP (IKEv1) and/or IKEv2
  - Transparent to DMVPN
  - Node can be responder for both ISAKMP and IKEv2
    - Both ISAKMP and IKEv2 are configured.
  - Node can be Initiator for either ISAKMP or IKEv2 not both
    - Configure under the 'crypto ipsec profile ...'

```
crypto isakmp policy 2
  encr aes
  authentication pre-share
  group 2
crypto ikev2 keyring DMVPN
  peer DMVPN
    address 0.0.0.0 0.0.0.0
    pre-shared-key cisco123
crypto ikev2 profile DMVPN
  match identity remote address 0.0.0.0
  authentication local pre-share
  authentication remote pre-share
  keyring DMVPN
```

```
crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set DMVPN esp-aes esp-sha-hmac
  mode transport [require]

crypto ipsec profile DMVPN
  set transform-set DMVPN
  set ikev2-profile DMVPN

interface Tunnel0
  ...
  tunnel protection ipsec profile DMVPN
```

With → initiate IKEv2
Without → initiate IKEv1

Per-tunnel QoS

(hub→spoke) 12.4(22)T; (spoke→hub, spoke→spoke) 15.5(1)S,T

```
interface Tunnel0
 nhrp group name
 ...
 nhrp map group name1 service-policy output qos-template1
 nhrp map group name2 service-policy output qos-template2
 ...
```

- QoS per tunnel on hub and spokes
  - Dynamically select Hierarchical (parent/child) QoS Policy
    - **Receiving Node:** Configure NHRP group name on tunnel
    - **Sending Node:** Configure QoS template policies; Map NHRP group name to QoS template policy
  - Nodes with same NHRP group name are mapped to separate instances of QoS policy
  - Same policy used for both IPv4 and IPv6
- QoS policy applied at outbound physical interface
  - Classification done before GRE encapsulation by tunnel
    - ACL matches against Data IP packet
    - Don't configure 'qos pre-classify' on tunnel interface
  - Shaping/policing done on physical after IPsec encryption
  - On physical may have separate aggregate QoS policy
    - With only a class-default shaper (15.2(2)T,S)
- CPU intensive; can reduce hub scaling by about 50% on software forwarding platforms

Per-tunnel QoS – Configurations

Hub and Spokes

```
class-map match-all typeA_voice
   match access-group 100
class-map match-all typeB_voice
   match access-group 100
class-map match-all typeA_Routing
   match ip precedence 6
class-map match-all typeB_Routing
   match ip precedence 6

policy-map typeA
   class typeA_voice
      priority 1000
   class typeA_Routing
      bandwidth percent 20

policy-map typeB
   class typeB_voice
      priority percent 20
   class typeB_Routing
      bandwidth percent 10

policy-map typeA_parent
   class class-default
      shape average 3000000
      service-policy typeA

policy-map typeB_parent
   class class-default
      shape average 2000000
      service-policy typeB
```

Hub

```
interface Tunnel0
   ip address 10.0.0.1 255.255.255.0
   …
   ip nhrp map multicast dynamic
   nhrp group typeB
   …
   nhrp map group typeA service-policy output typeA_parent
   nhrp map group typeB service-policy output typeB_parent
   …
   ip nhrp redirect
   …
```

Spoke1,3

```
interface Tunnel0
   ip address 10.0.0.[11,13] 255.255.255.0
   …
   nhrp group typeA
   …
   nhrp map group typeA service-policy output typeA_parent
   nhrp map group typeB service-policy output typeB_parent
   …
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   …
```

Spoke2

```
interface Tunnel0
   ip address 10.0.0.12 255.255.255.0
   …
   nhrp group typeB
   …
   nhrp map group typeA service-policy output typeA_parent
   nhrp map group typeB service-policy output typeB_parent
   …
   ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast
   …
```

Per-tunnel QoS – QoS Output on Hub

Hub#show ip nhrp

```
10.0.0.11/32 via 10.0.0.11
   Tunnel0 created 21:24:03, expire 00:04:01
   Type: dynamic, Flags: unique registered
   NBMA address: 172.16.1.1
   Group: typeA
10.0.0.12/32 via 10.0.0.12
   Tunnel0 created 21:22:33, expire 00:05:30
   Type: dynamic, Flags: unique registered
   NBMA address: 172.16.2.1
   Group: typeB
10.0.0.13/32 via 10.0.0.13
   Tunnel0 created 00:09:04, expire 00:04:05
   Type: dynamic, Flags: unique registered
   NBMA address: 172.16.3.1
   Group: typeA
```

Hub#show ip nhrp group-map

```
Interface: Tunnel0
   NHRP group: typeA
      QoS policy: typeA_parent
      Tunnels using the QoS policy:
      Tunnel destination overlay/transport address
      10.0.0.11/172.16.1.1
      10.0.0.13/172.16.3.1
   NHRP group: typeB
      QoS policy: typeB_parent
      Tunnels using the QoS policy:
      Tunnel destination overlay/transport address
      10.0.0.12/172.16.2.1
```

Hub#show policy-map multipoint tunnel 0 <spoke> output

```
Interface Tunnel0 ↔ 172.16.1.1
  Service-policy output: typeA_parent
    Class-map: class-default (match-any)
      19734 packets, 6667163 bytes
      shape (average) cir 3000000, bc 12000, be 12000
      Service-policy : typeA
        Class-map: typeA_voice (match-all)      3737 packets, 4274636 bytes
        Class-map: typeA_Routing (match-all)    14424 packets, 1269312 bytes
        Class-map: class-default (match-any)    1573 packets, 1123215 bytes
Interface Tunnel0 ↔ 172.16.2.1
  Service-policy output: typeB_parent
    Class-map: class-default (match-any)
      11420 packets, 1076898 bytes
      shape (average) cir 2000000, bc 8000, be 8000
      Service-policy : typeB
        Class-map: typeB_voice (match-all)      1005 packets, 128640 bytes
        Class-map: typeB_Routing (match-all)    10001 packets, 880088 bytes
        Class-map: class-default (match-any)    414 packets, 68170 bytes
Interface Tunnel0 ↔ 172.16.3.1
  Service-policy output: typeA_parent
    Class-map: class-default (match-any)
      5458 packets, 4783903 bytes
      shape (average) cir 3000000, bc 12000, be 12000
      Service-policy : typeA
        Class-map: typeA_voice (match-all)      4914 packets, 4734392 bytes
        Class-map: typeA_Routing (match-all)    523 packets, 46004 bytes
        Class-map: class-default (match-any)    21 packets, 14995 bytes
```

Thank you