
TOMORROW starts here.
cisco
Cisco live!

Advanced Concepts of DMVPN (Dynamic Multipoint VPN)

BRKSEC-4054

Mike Sullenberger

Distinguished Services Engineer

Cisco live!

Agenda

- DMVPN Overview
- NHRP Details
- Spoke Site Redundancy Use Case

DMVPN Overview

What is Dynamic Multipoint VPN?

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

- Relies on two proven technologies
  - Next Hop Resolution Protocol (NHRP)
    - Creates a distributed mapping database of VPN (tunnel interface) to real (public interface) 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 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.

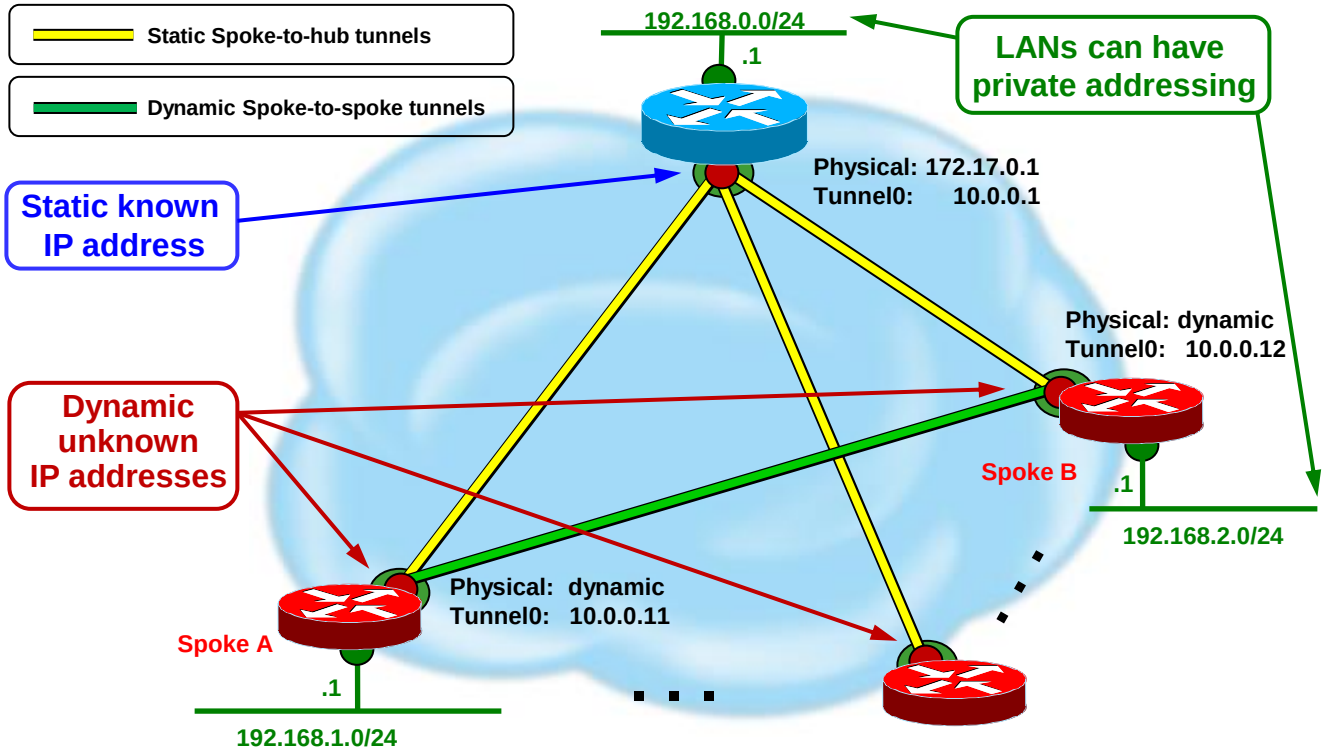
DMVPN Phases

Phase 1 – 12.2(13)T	Phase 2 – 12.3(4)T (Phase 1 +)	Phase 3 – 12.4.(6)T (Phase 2 +)
• Hub and spoke functionality • 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	• 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.

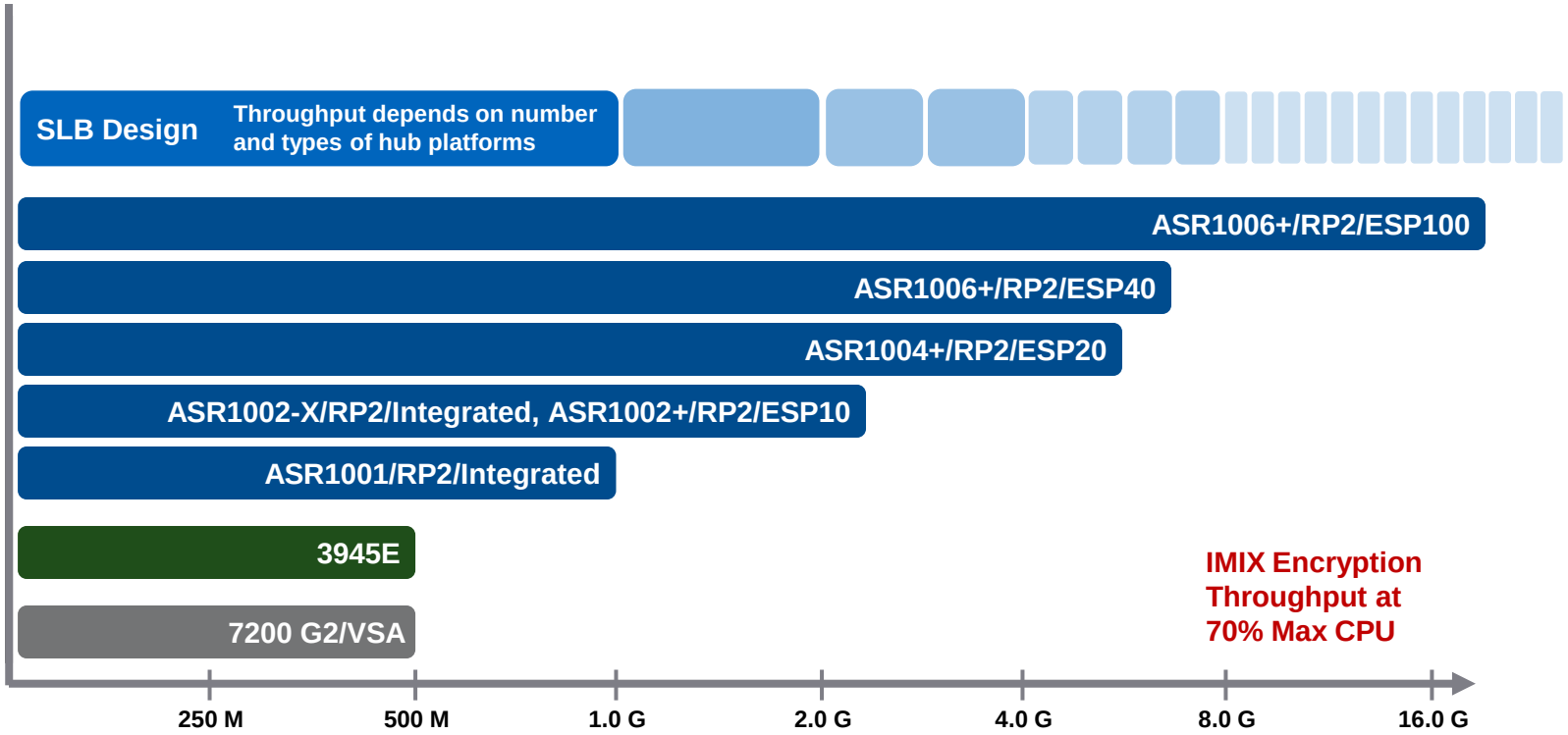
DMVPN Example



DMVPN and IPsec

- IPsec integrated with DMVPN, but not required
- Packets Encapsulated in GRE, then Encrypted with IPsec
- NHRP controls the tunnels, IPsec does encryption
- Bringing up a tunnel
  - NHRP signals IPsec to setup encryption
  - ISAKMP 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
  - IPsec can signal NHRP if encryption is cleared or lost
- ISAKMP Keepalives monitor state of spoke-spoke tunnels

Encryption Scaling



Routing over DMVPN

- Supports all routing protocols, except ISIS
- Best routing protocols are EIGRP and BGP
- 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 directly to routing table (15.2(1)T)

Routing Table Example (Spoke)

Phase 1 & 3
(with summarization)

```
C    172.16.1.0/30 is directly connected, Serial1/0
C    10.0.0.0/24 is directly connected, Tunnel0
C    192.168.1.0/24 is directly connected, Ethernet0/0
S*   0.0.0.0/0 is directly connected, Serial1/0
D    192.168.0.0/16 [90/2841600] via 10.0.0.1, 00:00:08, Tunnel0
```

Phase 1 & 3
(no summarization, next-hop **not** preserved)

```
C   172.16.1.0/30 is directly connected, Serial1/0
C   10.0.0.0/24 is directly connected, Tunnel0
D   192.168.0.0/24 [90/297372416] via 10.0.0.1, 00:02:36, Tunnel0
C   192.168.1.0/24 is directly connected, Ethernet0/0
D   192.168.2.0/24 [90/297321216] via 10.0.0.1, 00:02:36, Tunnel0
D   192.168.3.0/24 [90/297321216] via 10.0.0.1, 00:02:36, Tunnel0
. . .
S*  0.0.0.0/0 [1/0] via 172.16.1.1
```

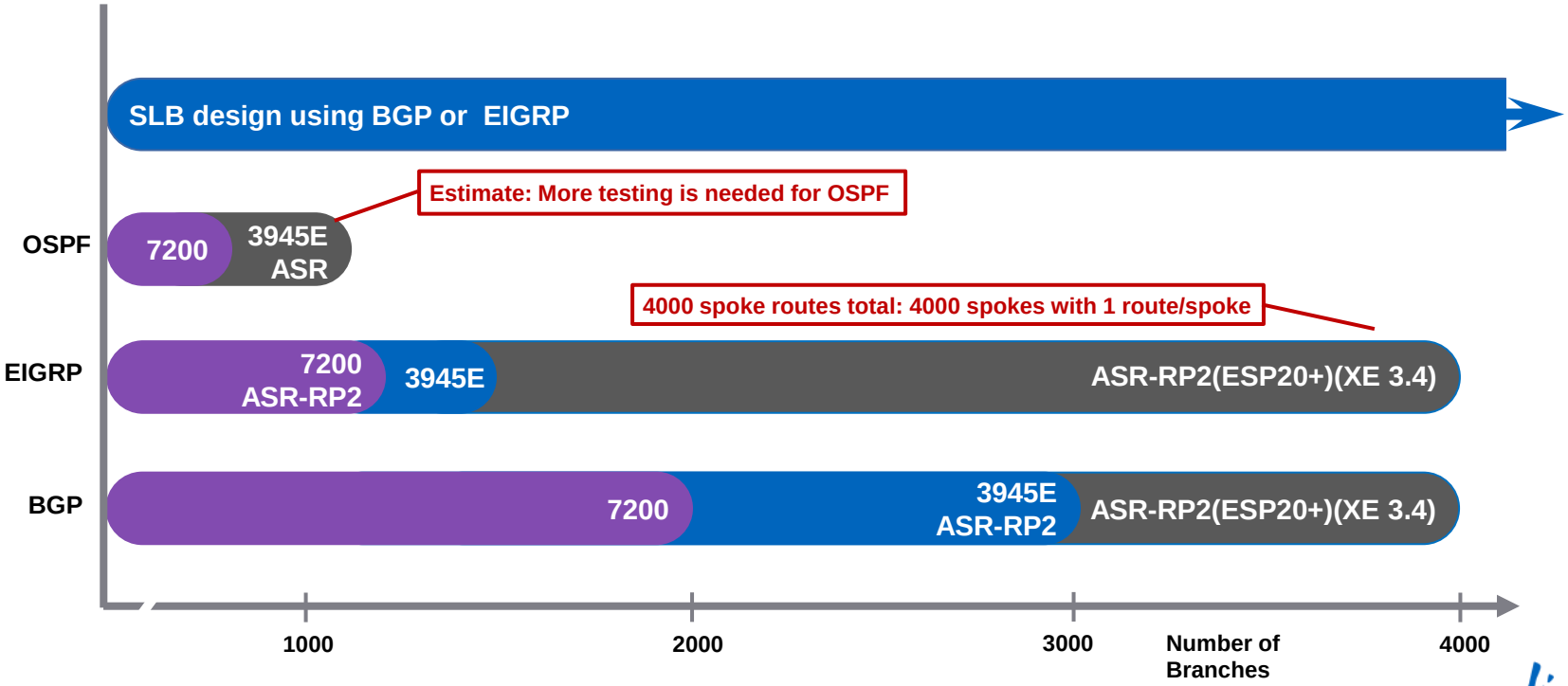
Phase 2
(no summarization, next-hop preserved)

```
C   172.16.1.0/30 is directly connected, Serial1/0
C   10.0.0.0/24 is directly connected, Tunnel0
D   192.168.0.0/24 [90/297372416] via 10.0.0.1, 00:42:34, Tunnel0
C   192.168.1.0/24 is directly connected, Ethernet0/0
D   192.168.2.0/24 [90/297321216] via 10.0.0.12, 00:42:34, Tunnel0
D   192.168.3.0/24 [90/297321216] via 10.0.0.13, 00:42:34. Tunnel0
. . .
S*  0.0.0.0/0 [1/0] via 172.16.1.1
```

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
- BUT...
  - EIGRP being an advanced distance vector protocol matches really well with DMVPN network topologies
  - BGP, specifically iBGP, can run well over DMVPN, but it is more complicated to setup and to have it act more like an IGP rather than a EGP.
  - OSPF can run over DMVPN, BUT lesser scaling and Area 0 issues can complicate the network.
  - RIP can be used, but has longer holdtime 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
  - Routing protocol routes are used to determine traffic forwarding
    - Single route: one tunnel (hub) at a time – primary/backup mode
    - Multiple routes: both tunnels (hubs) – load-balancing mode

- ISAKMP/IPsec
  - Cannot use IPsec Stateful failover (NHRP isn't supported)
  - ISAKMP invalid SPI recovery is not useful with DMVPN
    - no crypto isakmp invalid-spi-recovery
  - ISAKMP keepalives on spokes for timely hub recovery
    - crypto isakmp keepalives [periodic] *initial retry*
    - crypto isakmp nat keepalive *interval*

Redundancy (cont)

- Can use single or multiple DMVPNs for redundancy
  - Each mGRE interface is a separate DMVPN network using different tunnel key, NHRP network-id and IP subnet
  - Can "glue" mGRE interfaces into same DMVPN network;(*) same tunnel source, NHRP network-id and authentication, no tunnel key and different IP subnet (Phase 3 only)
  - If using same tunnel source (must use tunnel key)
    - tunnel protection ipsec profile *name* shared

- Spokes – at least two hubs (NHSs)
  - Phase 1: (Hub-and-spoke)
    - p-pGRE interfaces → two DMVPN networks, one hub on each
  - Phase 1, 2 or 3: (Hub-and-spoke or Dynamic Mesh)
    - mGRE interface → one DMVPN network, two or more hubs

Redundancy (cont)

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

Spoke-Spoke Tunnels Considerations

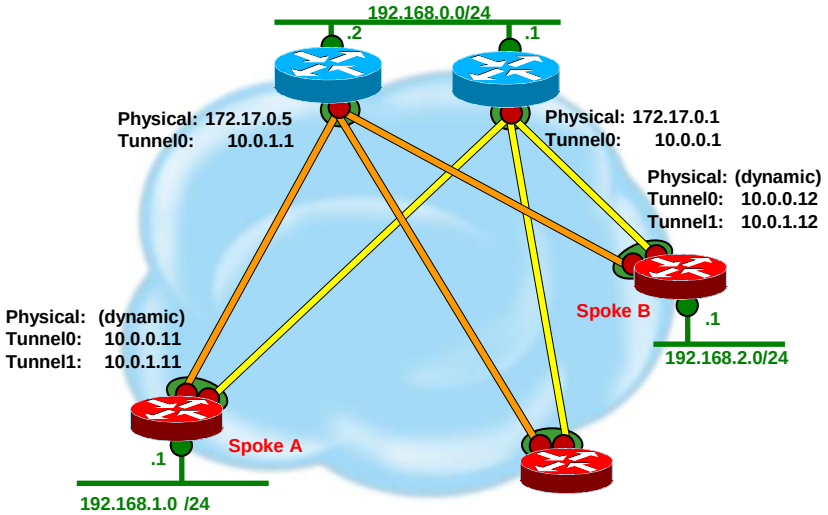
- Resiliency
  - No monitoring of spoke-spoke tunnel (use ISAKMP keepalives)
    crypto isakmp keepalives [periodic] *initial retry*
- Path Selection*
  - NHRP will always build spoke-spoke tunnel
  - No bandwidth/latency measurement of spoke-spoke vs. spoke-hub-spoke paths
- Overloading spoke routers
  - CPU or memory → IKE Call Admission Control (CAC)
    crypto call admission limit ike {sa | in-negotiation } *max-SAs*
    call admission limit percent
    show crypto call admission statistics
  - Bandwidth → Design for expected traffic
    - Hub-spoke versus Spoke-spoke; Spoke-spoke availability is best effort

Basic DMVPN Designs

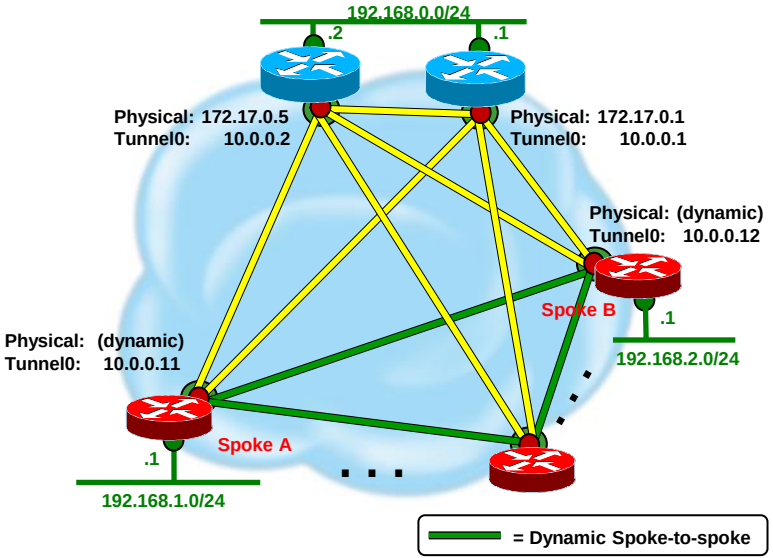
- Hub-and-spoke – Order(n)
  - Spoke-to-spoke traffic via hub
    - Phase 1: Hub bandwidth and CPU limit VPN
    - SLB: Many “identical” hubs increase CPU limit
- Spoke-to-spoke – Order(n) « Order(n^2)
  - Control traffic; Hub and spoke; Hub to hub
    - Phase 2: (single)
    - Phase 3: (hierarchical)
  - 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
  - MPLS over DMVPN (2547oDMVPN); Single DMVPN

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

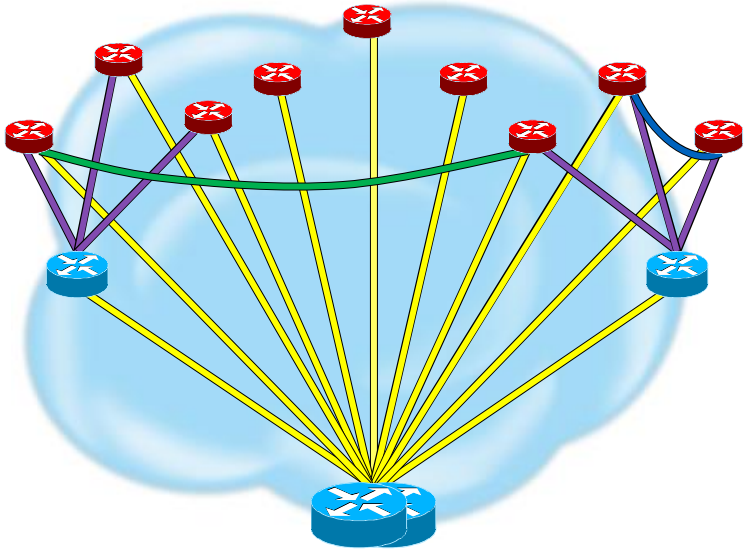


Multiple DMVPNs versus Single DMVPN

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

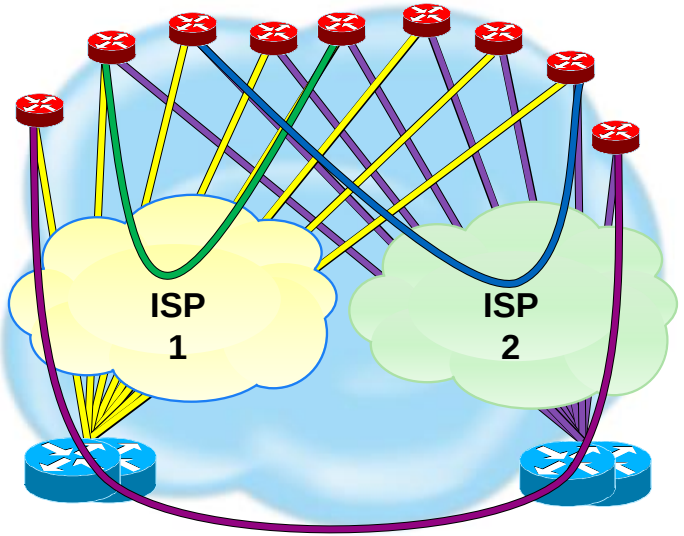
DMVPN Combination Designs

Retail/Franchise



Spoke-to-hub tunnels
Spoke-to-spoke tunnels

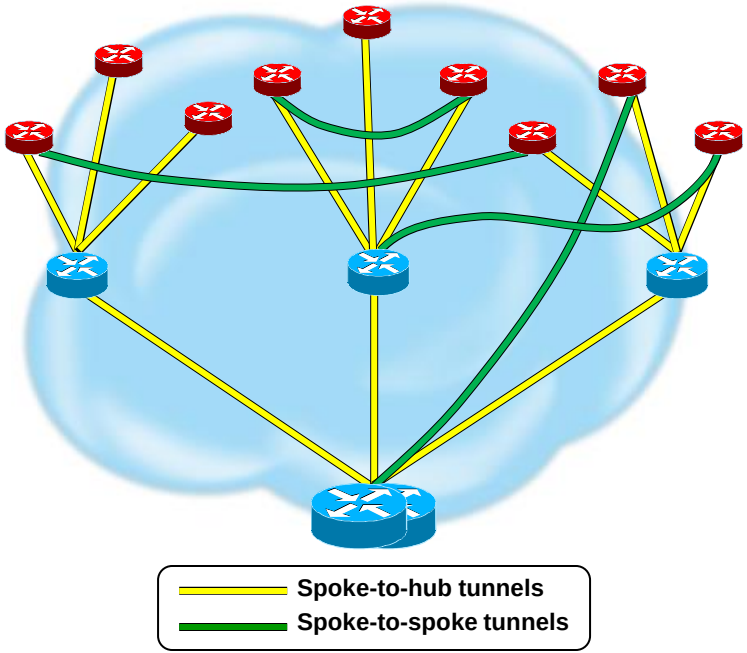
Dual ISP



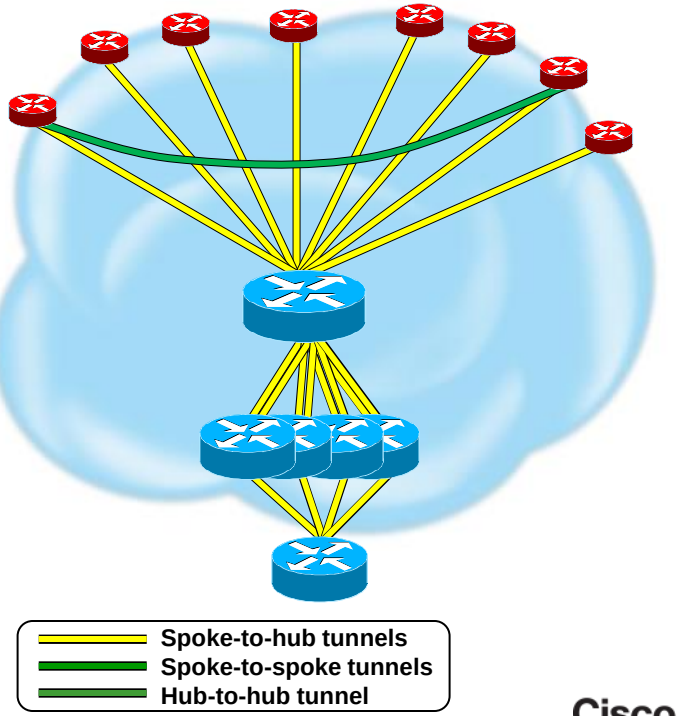
Spoke-to-hub tunnels
Spoke-to-spoke tunnels
Spoke-hub-hub-spoke tunnel

DMVPN Combination Designs (cont)

Hierarchical



Server Load Balancing



Network Virtualization
Separate DMVPNs – VRF-lite

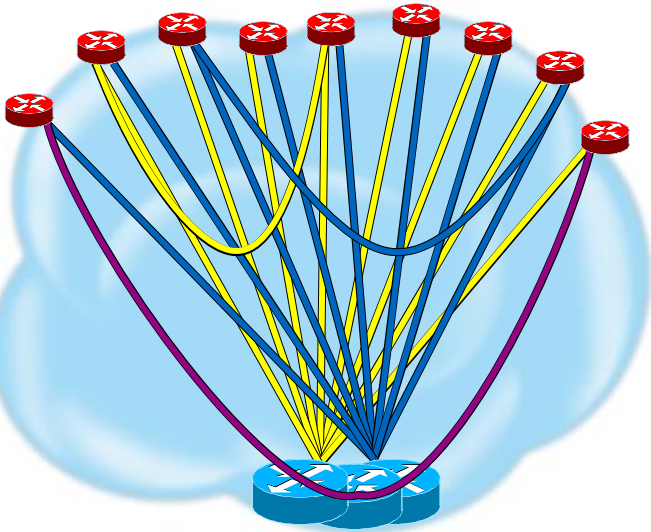
- Separate DMVPN mGRE tunnel per VRF
- Hub routers handle all DMVPNs
  - Multiple Hub routers for redundancy and load
- IGP used for routing protocol outside of and over DMVPNs on Spokes and Hubs
  - Address family per VRF
  - Routing neighbor per spoke per VRF
- BGP used only on the hub
  - Redistribute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

Network Virtualization
MPLS over DMVPN – 2547oDMVPN

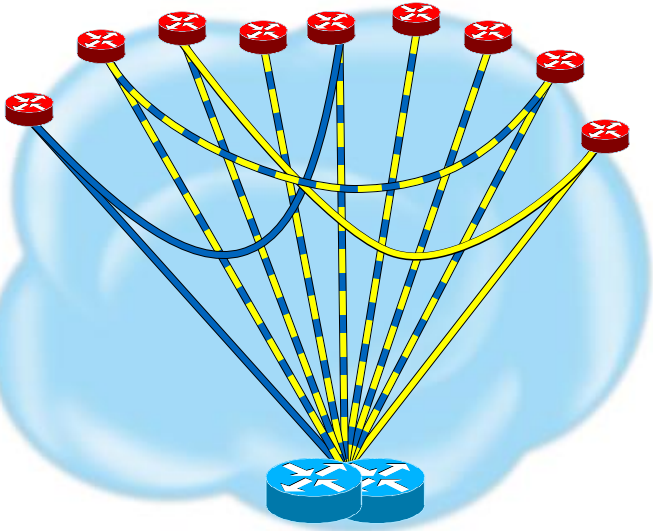
- Single DMVPN (Hub-and-spoke Only)
  - MPLS VPN over DMVPN
  - Single mGRE tunnel on all routers
- MPLS configuration
  - Hub and Spoke routers are MPLS PEs
- Multiple Hub routers for redundancy and load
- IGP is used for routing outside of DMVPN network
- BGP used for routing protocol over DMVPN
  - Redistribute between IGP and BGP for transport over DMVPN
  - Import/export of routes between VRFs and Global (or Internet VRF)
  - One routing neighbor per spoke
- Global routing table for routing DMVPN tunnel packets

DMVPN Network Virtualization Designs

VRF-lite



2547oDMVPN



Cisco IOS Code and Platform Support

- 3800, 2800, 1800, 870
  - 12.4(25g) **(not 870)**; 12.4(15)T17, 12.4(24)T8; 15.1(4)M6; 15.1(3)T4
- 7200(G1), 7200(G2, VSA)
  - 12.4(25g) **(7200(G1) only)**;12.4(24)T8; 15.1(4)M6, 15.2(4)M3; 15.2(4)S3
- 3900(E), 2900, 1900, 890, 880
  - 12.4(24)T8 **(880 only)**; 15.1(4)M6, 15.2(4)M3; 15.2(3)T3, 15.3(2)T
- ASR1000 (RP2) (ASR1002-X Released 3.7.0 - 15.2(4)S)
  - (3.7.3S) 15.2(4)S3, (3.8.2S)15.3(1)S2, (3.9.0S)15.3(2)S

NHRP Details

Agenda

- DMVPN Overview
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
    - Phase 2
    - Phase 3
- Spoke Site Redundancy Use Case

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
  - 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 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 spoke-spoke path

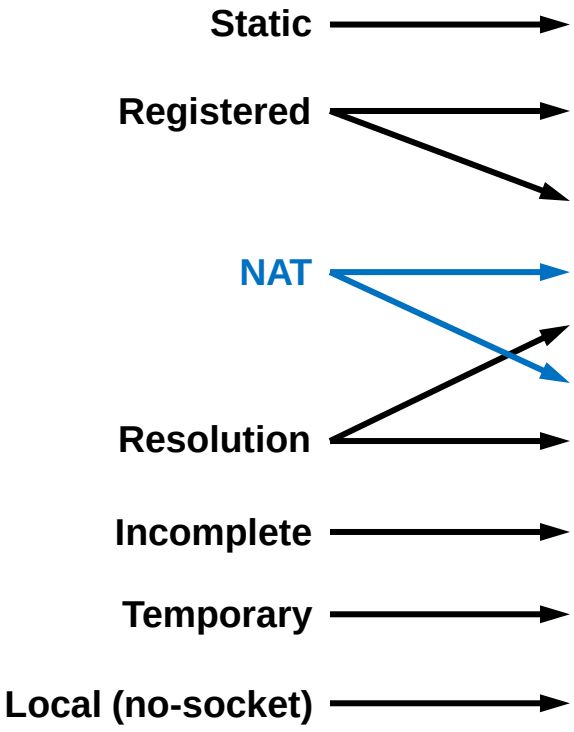
NHRP Message Extension Types

- Responder Address Extension:
  - Address mapping for Responding node (Reply messages)
- Forward Transit NHS Record Extension:
  - List of NHSs that NHRP request message traversed – copied to reply message
- Reverse Transit NHS Record Extension:
  - List of NHSs that NHRP reply message traversed
- Authentication Extension:
  - NHRP Authentication
- NAT Address Extension: **(12.4(6)T)**
  - Address mapping for peer (Registration request/reply)
  - Address mapping for self (Resolution request/reply)

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 NAT address
  - **Learned (/32, /128 or /<x>)**
    - From NHRP Resolution; NAT – record both inside and outside NAT address
- **Incomplete (/32, /128)** (also see Temporary)
  - Rate-limit sending of NHRP Resolution Requests
  - Process-switching of data packet while building spoke-spoke tunnels.
- **Local (/32, /128 or /<x>)**
  - Mapping for local network sent in an NHRP Resolution Reply
  - Record which nodes were sent this mapping
- **Temporary (/32)** (**12.4(22)T**)
  - Same as “Incomplete” mapping except that NBMA is set to Hub
  - CEF-switching of data packets while building spoke-spoke tunnels.
- **(no socket)**
  - Not used to forward data packets; Do not trigger IPsec encryption

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, removed 12.4(15)T)	Remote peer supports the NHRP NAT extension
rib (12.2(33)XNE – ASR1k, 15.2(1)T)	Routing Table entry created
nho (12.2(33)XNE – ASR1k, 15.2(1)T)	Next-Hop-Override Routing Table entry created

Agenda

- DMVPN Overview
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
    - Phase 2
    - Phase 3
- Spoke Site Redundancy Use Case

Hub-and-Spoke Features

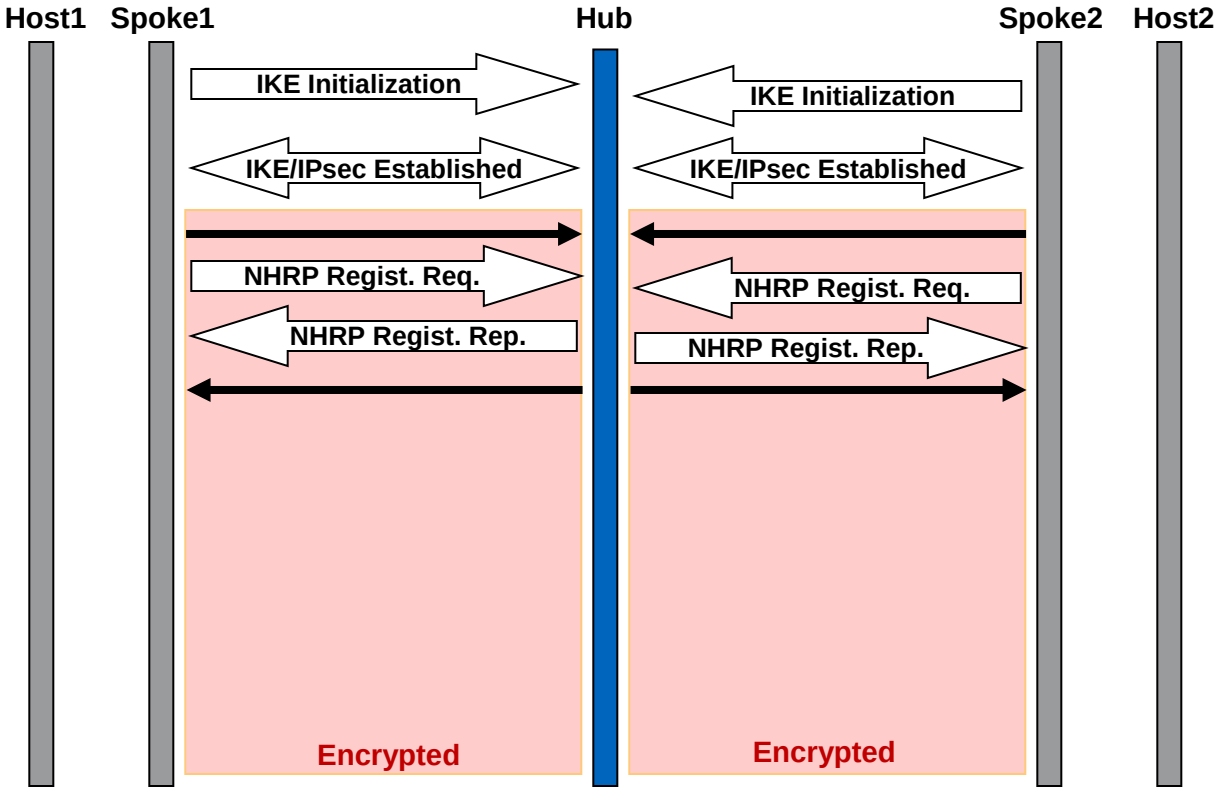
- GRE, NHRP and IPsec configuration
  - p-pGRE or mGRE on spokes; mGRE on hubs
  - ISAKMP Authentication
    - Certificate, (Pairwise/Wildcard) Pre-shared Key

- NHRP Registration
  - Static NHRP mapping for Hub on Spoke
  - Dynamically learn NHRP mapping for Spoke on Hub
    - Dynamically addressed spokes (DHCP, NAT , …)
  - NAT detection support

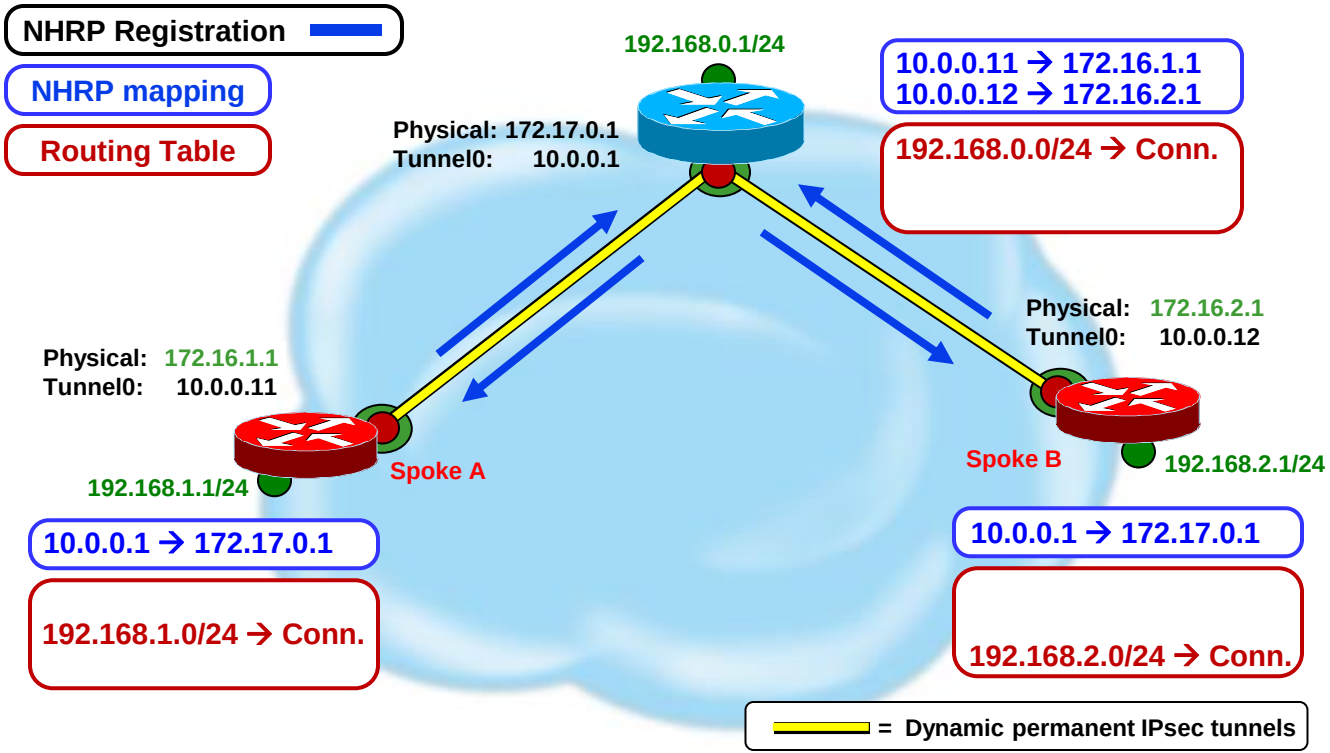
NHRP Registration

- Builds base hub-and-spoke network
  - Hub-and-spoke data traffic
  - Control traffic; NHRP, Routing protocol, IP multicast
  - Phase 2 – Single level 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
  - Supplies outside NAT address of Hub
  - NHRP-group for per-Tunnel QoS (12.4(22)T)
- NHS registration reply gives liveliness of NHS
  - Supplies outside NAT address of spoke

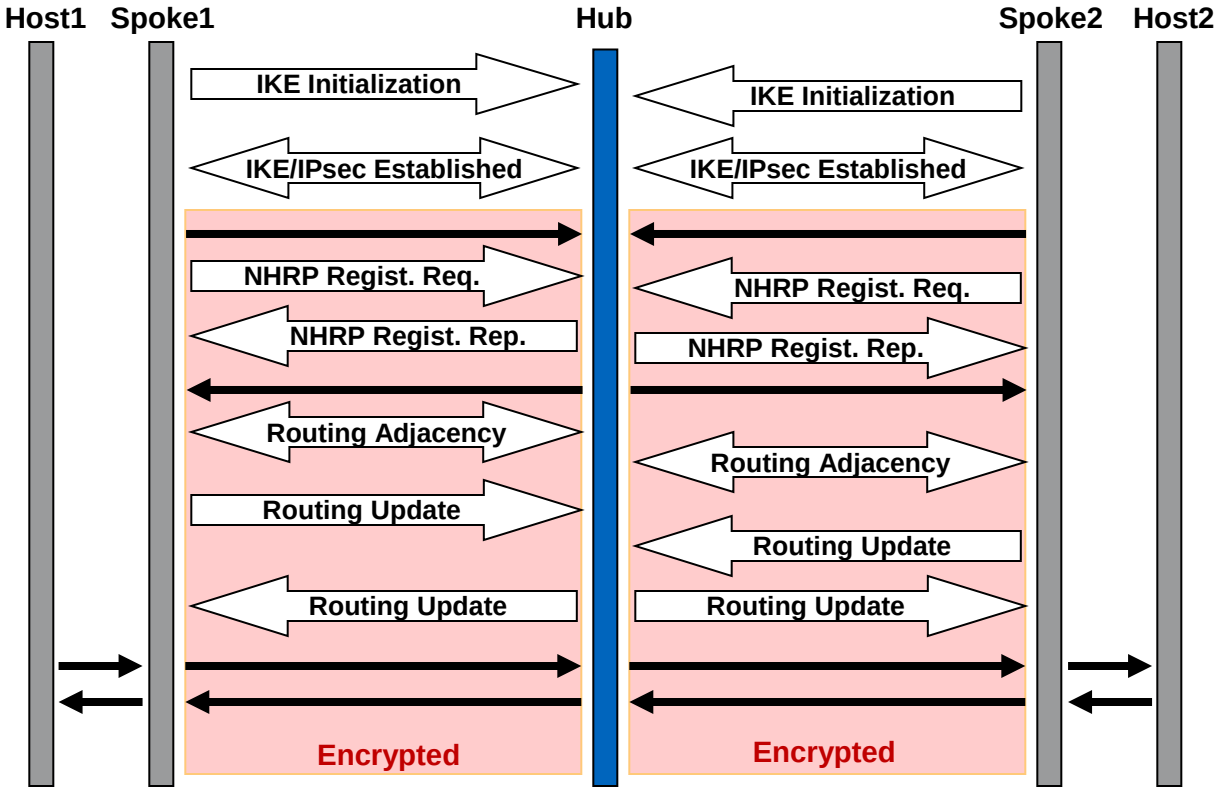
NHRP Registration
Building Spoke-Hub Tunnels



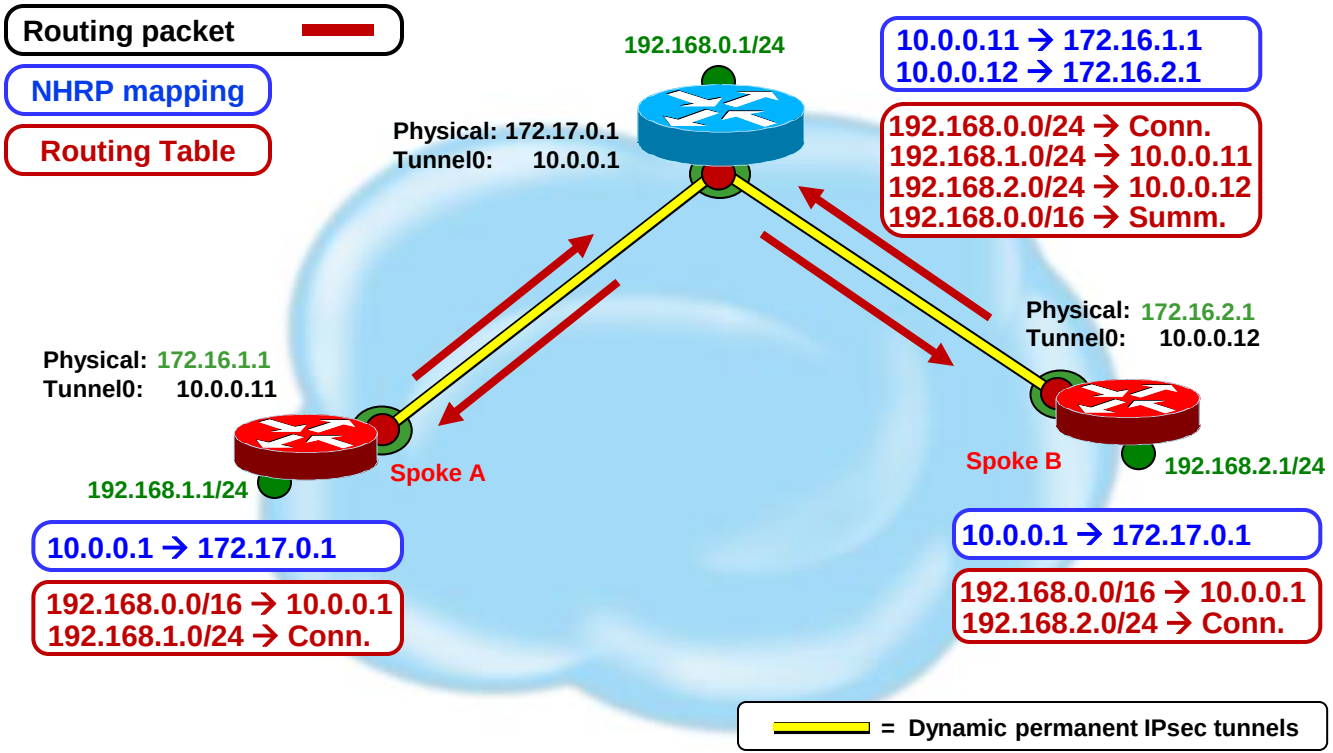
NHRP Registration
Building Spoke-Hub Tunnels



NHRP Registration (cont)
Routing Adjacency



NHRP Registration (cont)
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 Overview
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
    - Phase 2
    - Phase 3
- Spoke Site Redundancy Use Case

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
  - ISAKMP 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
Triggering NHRP Resolutions

- 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 destination
  - 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

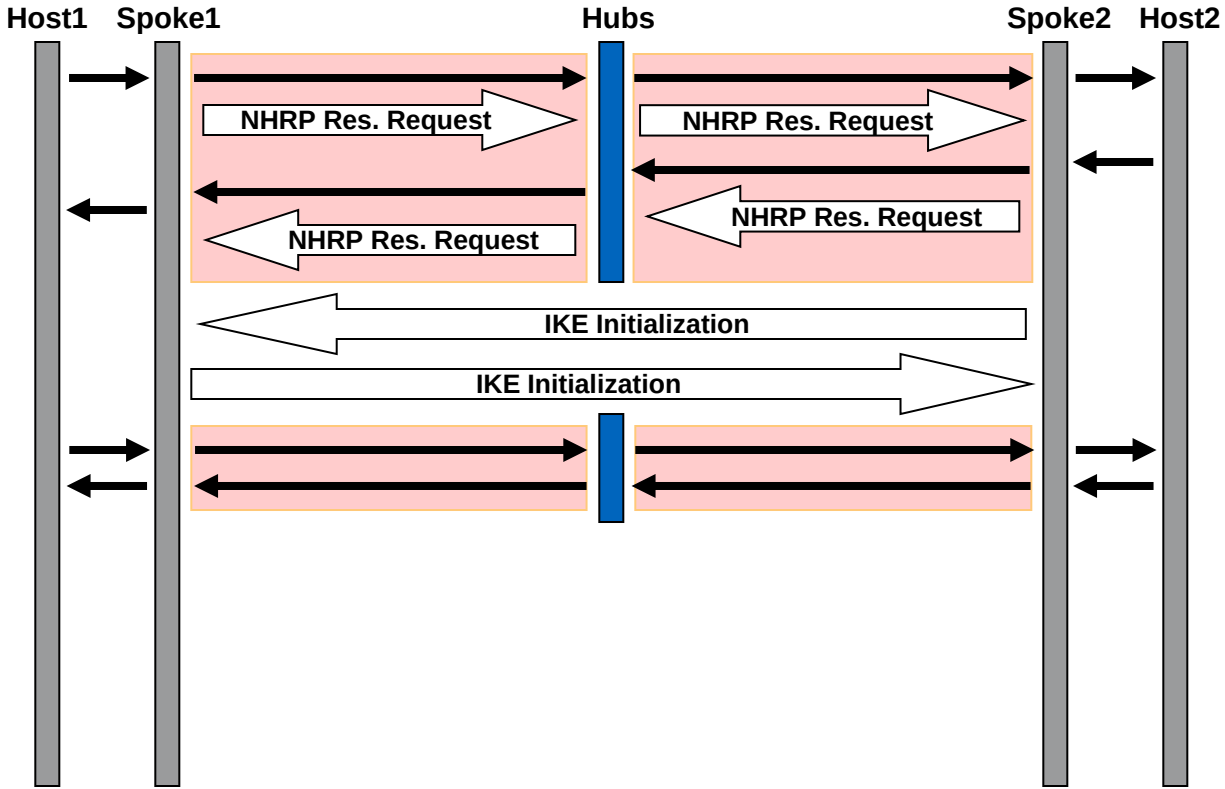
Phase 2 – CEF Switching
Triggering NHRP Resolutions

- IP Data packet is forwarded out tunnel interface to IP next-hop from CEF 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
    - Resolution reply is used to create NHRP mapping and to complete the Adjacency

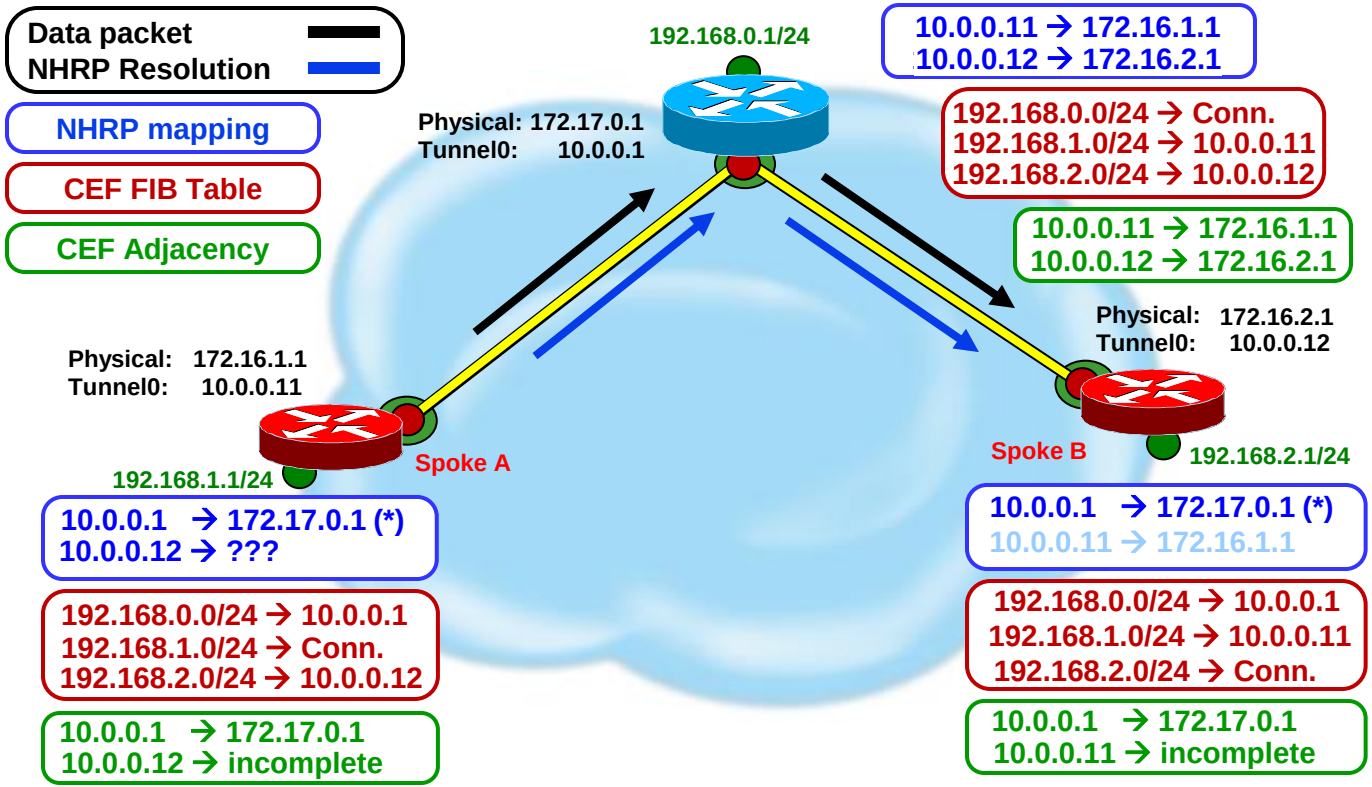
Phase 2
NHRP Resolution process changes

- When:
  - 12.4(6)T, 12.2(33)XNE and later
- Why:
  - To Support spoke-spoke tunnels when spokes are behind NAT
- How:
  - Registered NHRP mappings on hub are **not** marked Authoritative
- Effect:
  - Resolution request will be forwarded via NHS path **all** the way to the remote spoke
  - Resolution request is answered by the remote spoke
  - Spoke-spoke tunnel is built
  - Resolution reply forwarded back via spoke-spoke tunnel

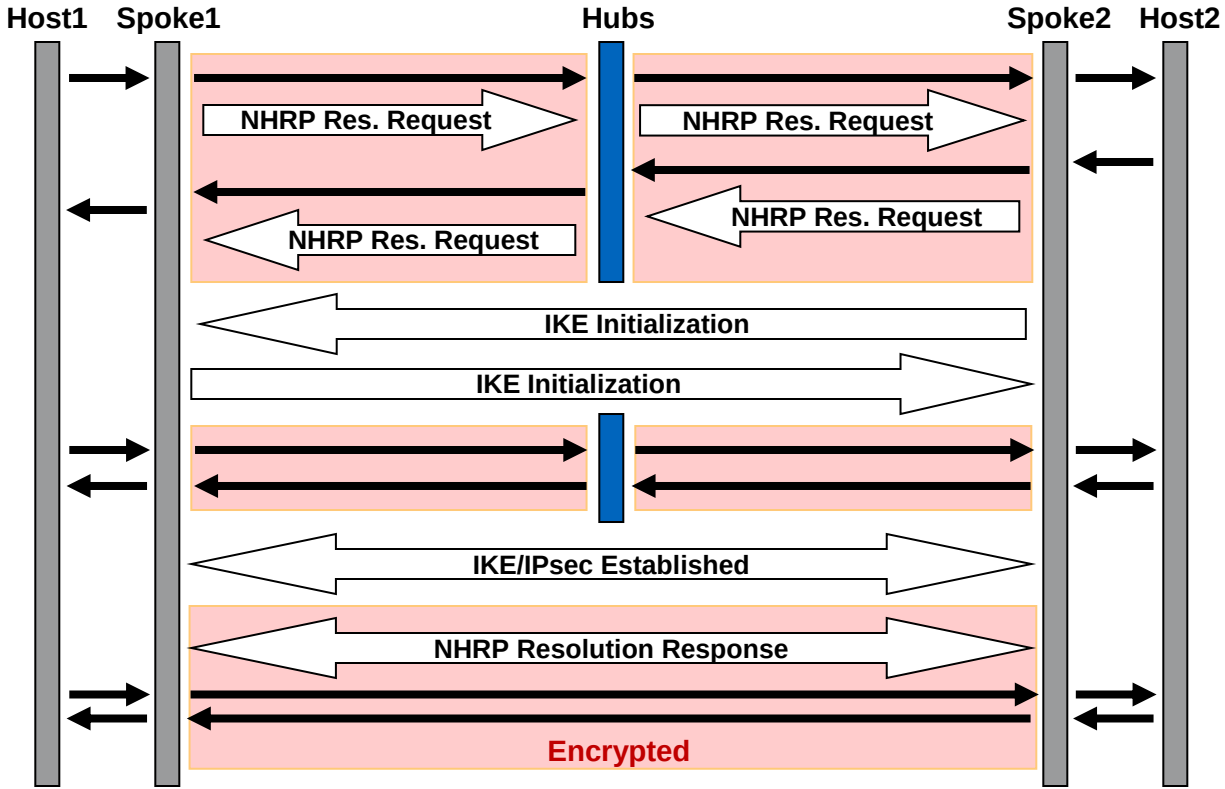
Phase 2
NHRP Resolution Request



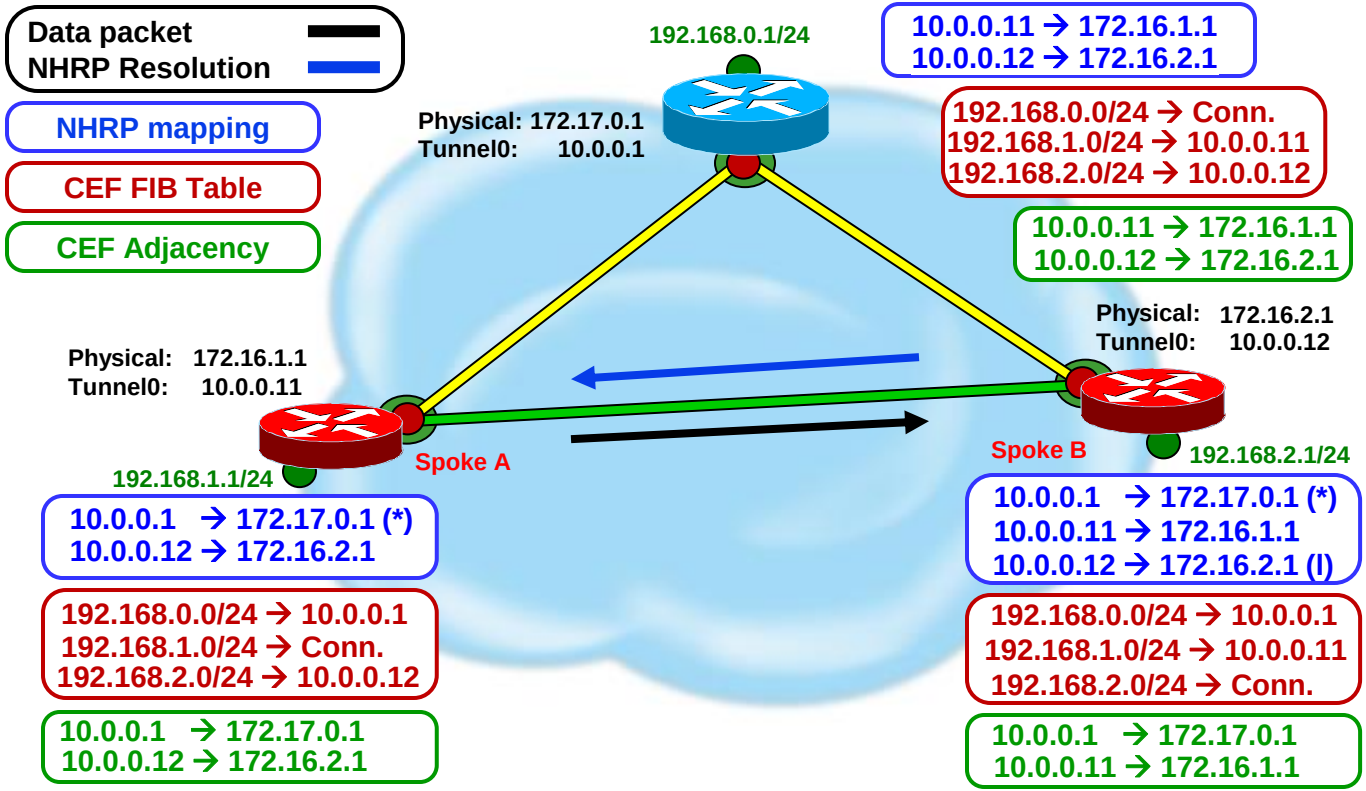
Phase 2
NHRP Resolution Request



Phase 2
NHRP Resolution Reply



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 – Dynamic mappings
Refresh or Remove

- 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

Agenda

- DMVPN Overview
- NHRP Details
  - NHRP Overview
  - NHRP Registrations
  - NHRP Resolutions/Redirects
    - Phase 2
    - Phase 3
- Spoke Site Redundancy Use Case

Phase 3 – Features

- Used to increase scale of DMVPN networks
  - Increase number of spokes, with same spoke/hub ratio
  - Distribution hubs off load central hub
    - Manage local spoke-spoke tunnels
    - IP multicast and routing protocol
- No hub daisy-chain
  - Use routing and CEF switching to forward data and NHRP packets through hubs
  - Reduces complexity and load for routing protocol
- OSPF routing protocol not limited to 2 hubs
  - Network point-multipoint mode
  - Still single OSPF area and no summarization

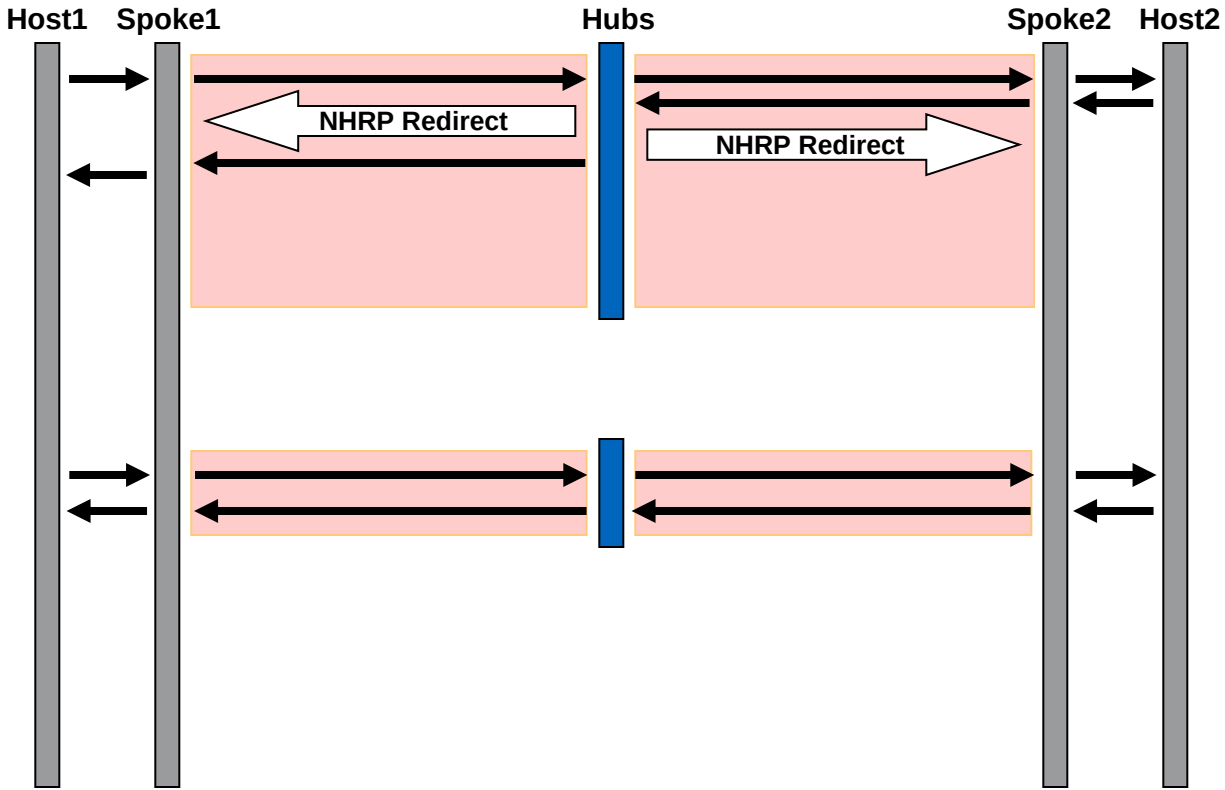
Phase 3 – Features (cont)

- Spokes do not need full routing tables
  - Can summarize routes at the hub
  - Reduced space and load on small spokes
  - Reduced routing protocol load on hub
    - 1000 spokes, 1 route per spoke;
    - hub advertises 1 route to 1000 spokes → 1000 advertisements
- Not available on 6500 or 7600
- Not recommended to mix Phase 2 and Phase 3 on same DMVPN
  - Migrate spokes from Phase 2 DMVPN to Phase 3 DMVPN

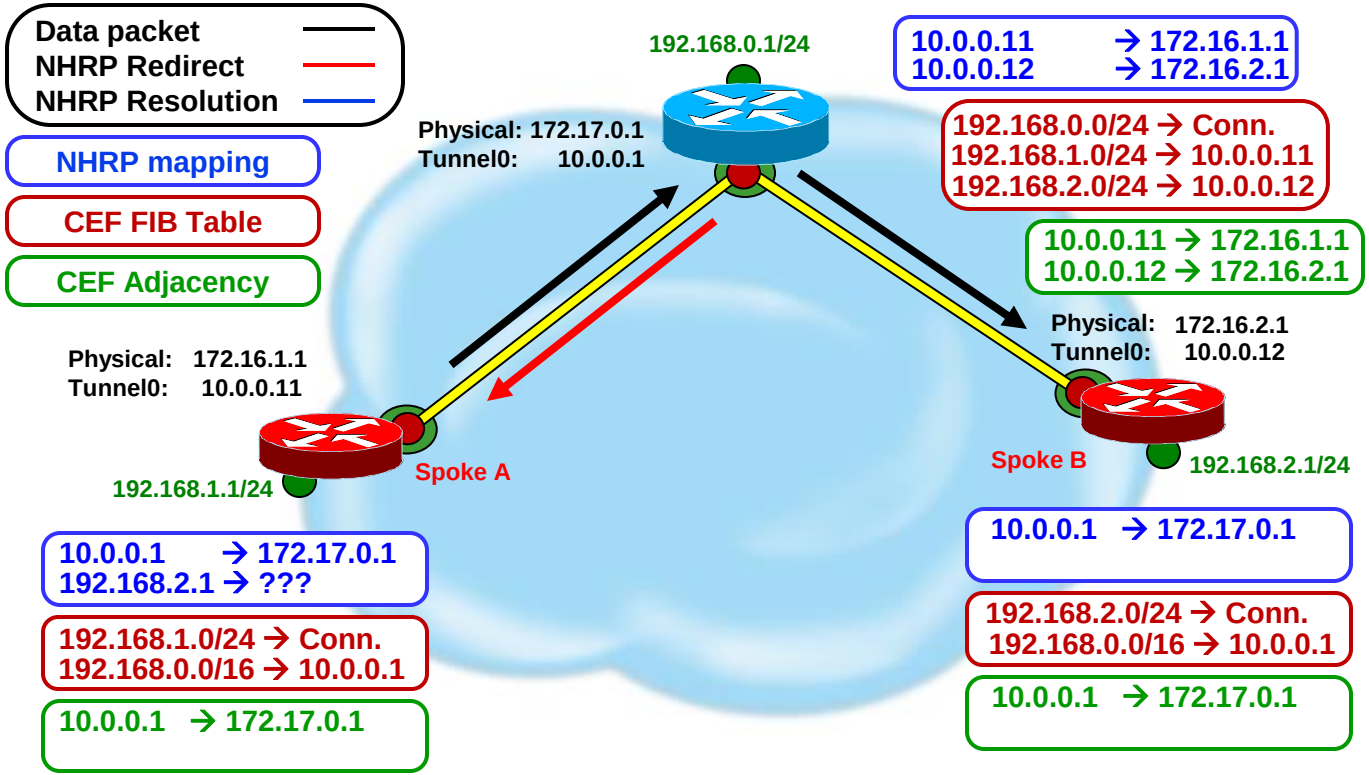
Phase 3
Building Spoke-spoke Tunnels

- 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.
  - Sends NHRP Redirect message to originating spoke.
- Originating spoke
  - Receives NHRP redirect message
  - Sends NHRP Resolution Request for Data IP packet destination via NHS
- Destination spoke
  - Receives NHRP Resolution Request
  - Builds spoke-spoke tunnel
  - Sends NHRP Resolution Reply over spoke-spoke tunnel

Phase 3
NHRP Redirects



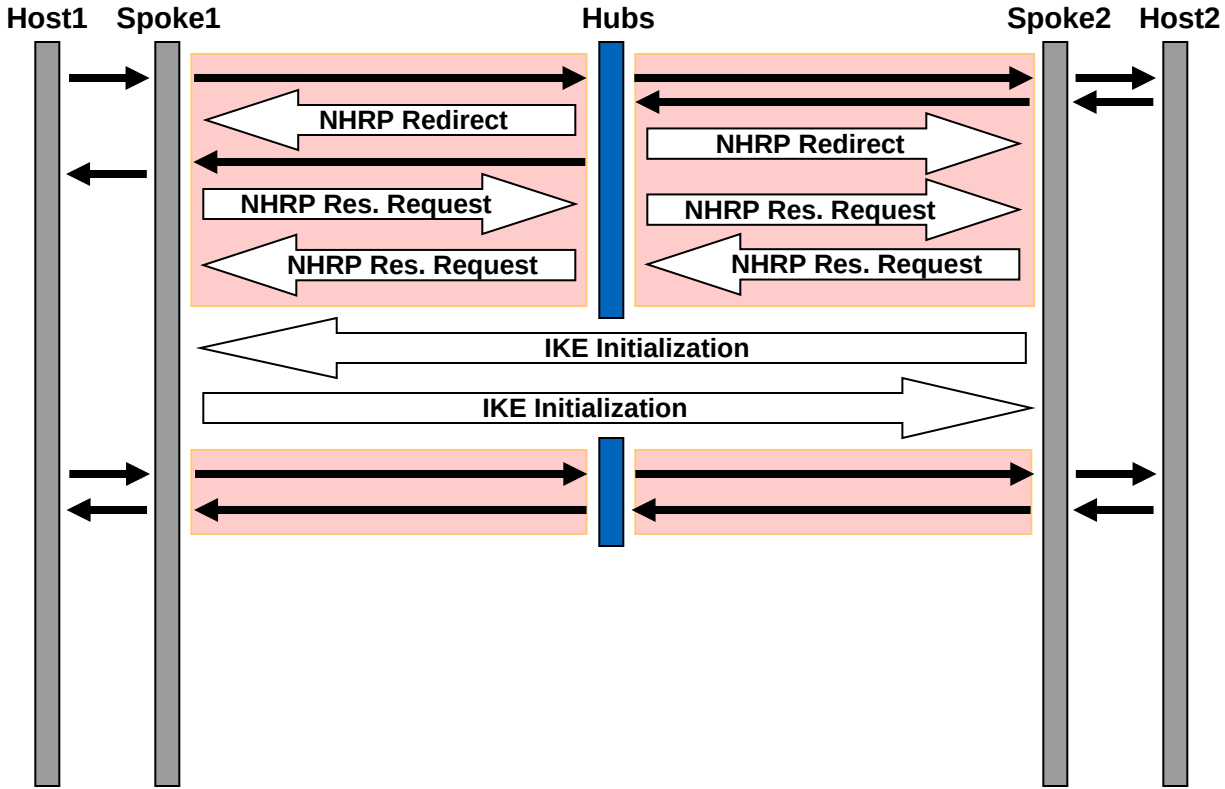
Phase 3
NHRP Redirects



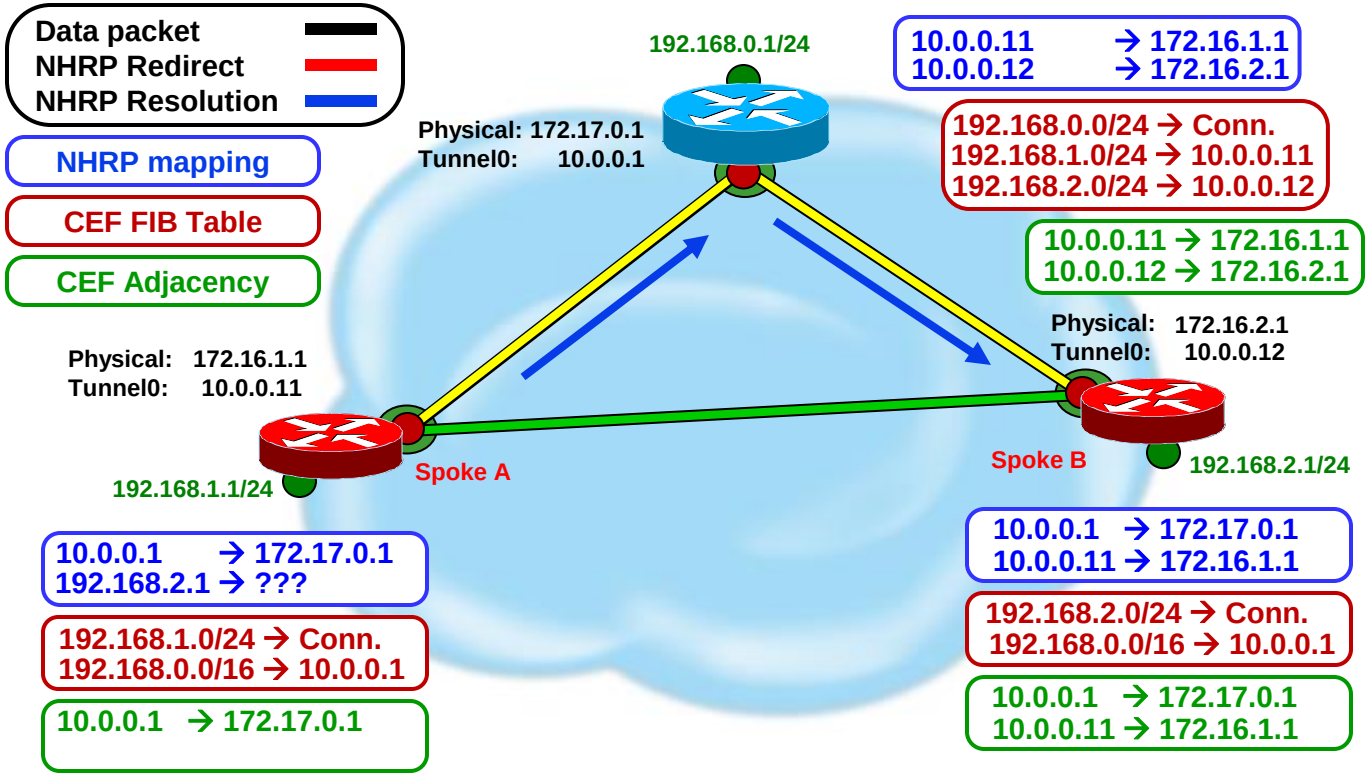
Phase 3
NHRP Redirect Processing

- Sender
  - Insert (GRE IP header source, packet destination IP address) in NHRP redirect table – used to rate-limit NHRP redirect messages
  - Send NHRP redirect to GRE/IP header source
  - 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, the IP destination is permitted by 'ip nhrp interest *ACL*' and 'ip nhrp shortcut' is configured
      - then trigger an NHRP resolution request to IP destination
  - Otherwise drop redirect

Phase 3
NHRP Resolution Request



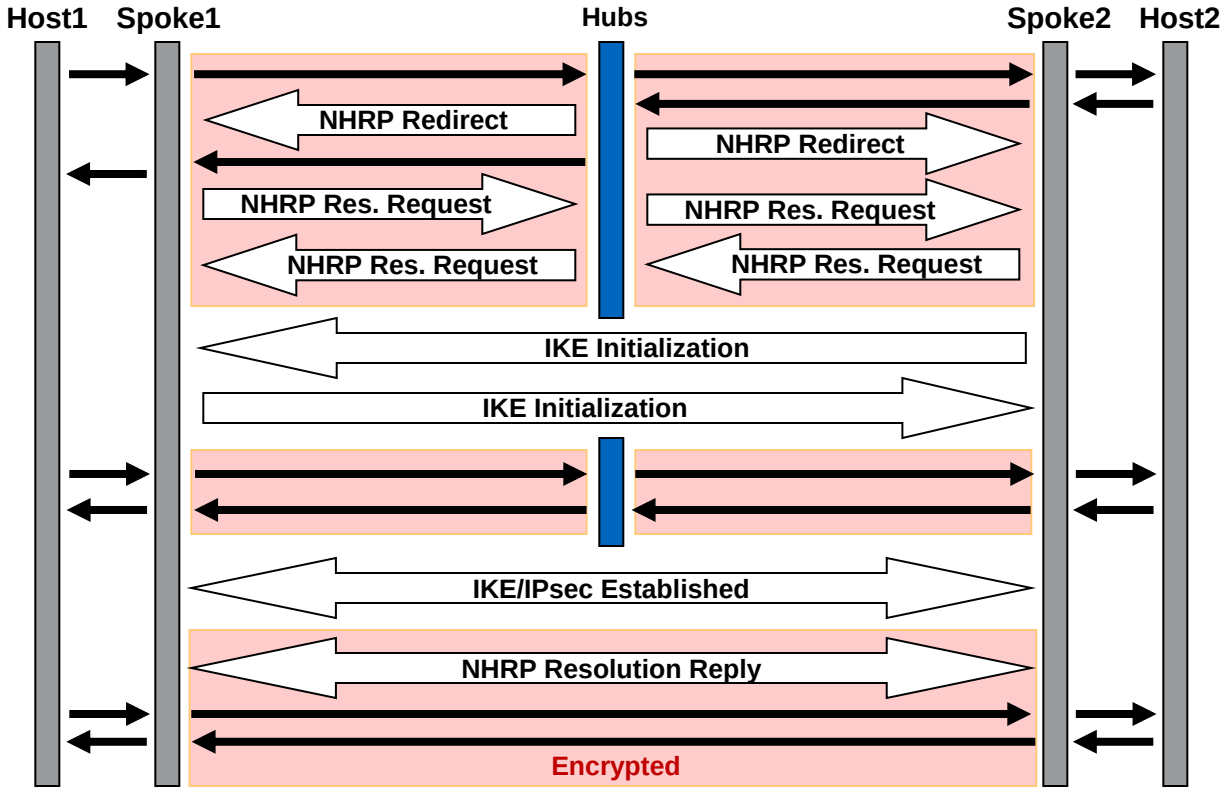
Phase 3
NHRP Resolution Request



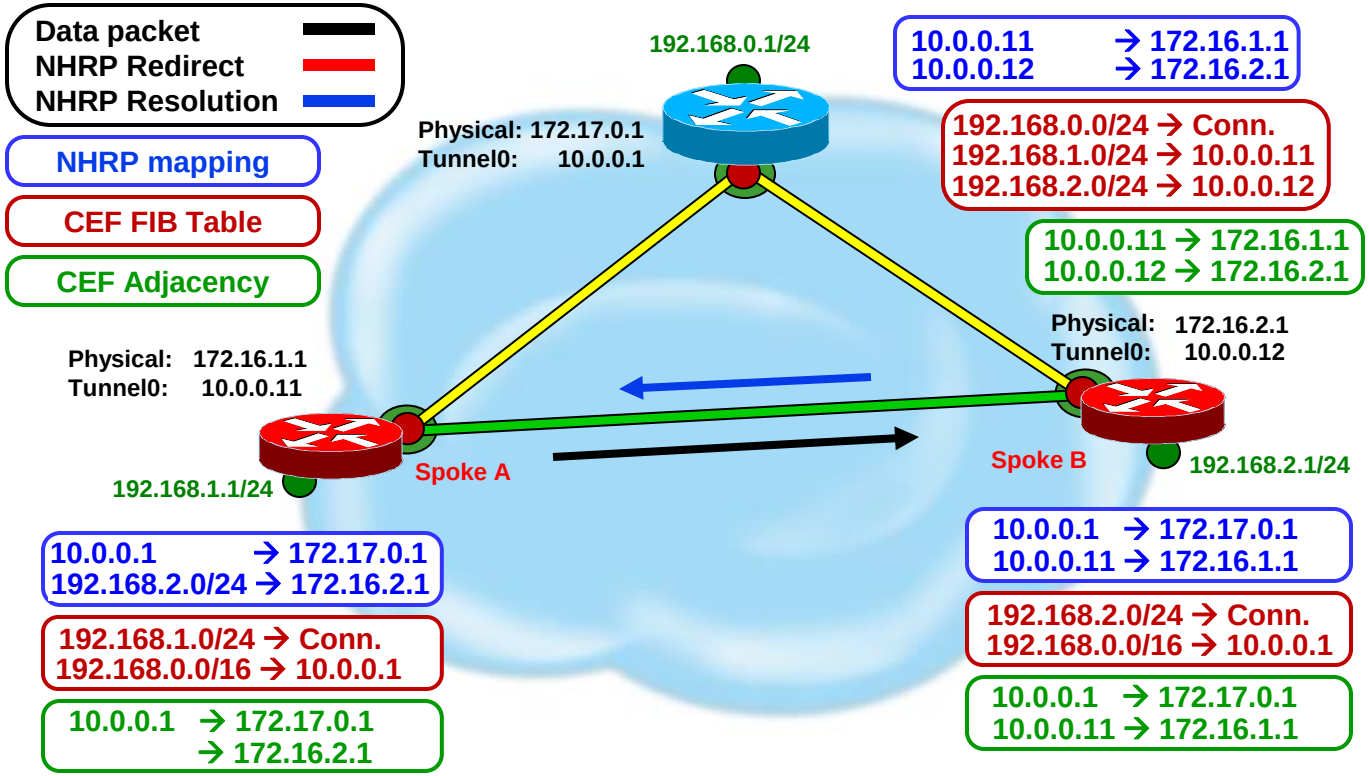
Phase 3
NHRP Resolution Processing

- 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 message
- 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
NHRP Resolution Reply



Phase 3 – CEF Switching
Data Packet Forwarding

(Current – ISR, 7200)

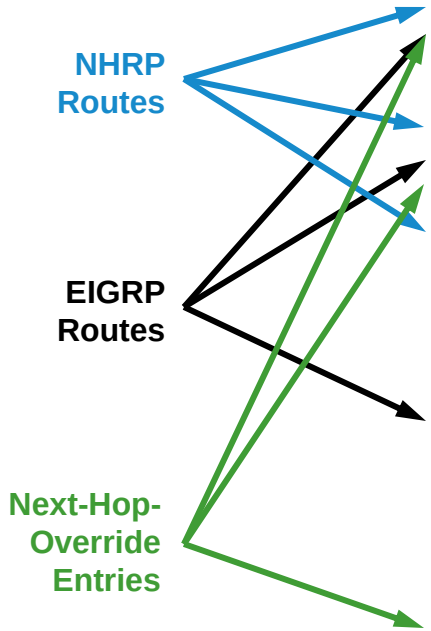
- IP Data packet is forwarded out tunnel interface
  1. IP next-hop from CEF FIB mapped to Adjacency
     If adjacency is:
     - Glean or Incomplete → Punt to process switching
     - Valid → Select adjacency for the packet
  2. NHRP in Outbound CEF Feature path
     Look up packet IP destination in 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

Phase 3 – NHRP and Routing Table
Data Packet Forwarding (ASR1k; 15.2(1)T – ISR, 7200)

- 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 more specific than RT Net/Mask
      - Add new route owned by NHRP **(Type = H)**
      - Monitor parent route
      - If parent route changes outbound interface then remove NHRP route.
    - 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 less specific than RT Net/Mask
      - Reduce NHRP mask to = RT Mask
      - Add Override Alternate Next-hop **(% flag)**
      - Route still owned by original owner

Phase 3 – NHRP and RT Routing Table

(ASR1k; 15.2(1)T – ISR, 7200)




```
#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 – Dynamic Mappings
Refresh or Remove

- Dynamic NHRP mapping entries have finite lifetime
  - Controlled by 'ip nhrp holdtime …' on source of mapping (spoke)
  - Two types of mapping entries
    - Master entry – Remote Spoke Tunnel IP address
    - Child entries – Remote Network address(es)
- Background process checks mapping entries every 60 seconds
  - Master entry: Timing out* → mark CEF adjacency stale and mark Child entries used.
    - If CEF adjacency is used → refresh Master entry
  - Child entry: If marked used and timing out → refresh Child entry
- 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 NBMA address
    - Trigger IPsec to remove IPsec and ISAKMP SAs

*** Expire timer < 120 seconds**

NHRP Purge Messages

- Used to clear invalid NHRP mapping information from the network
- NHRP “local” 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
    - To see use ‘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

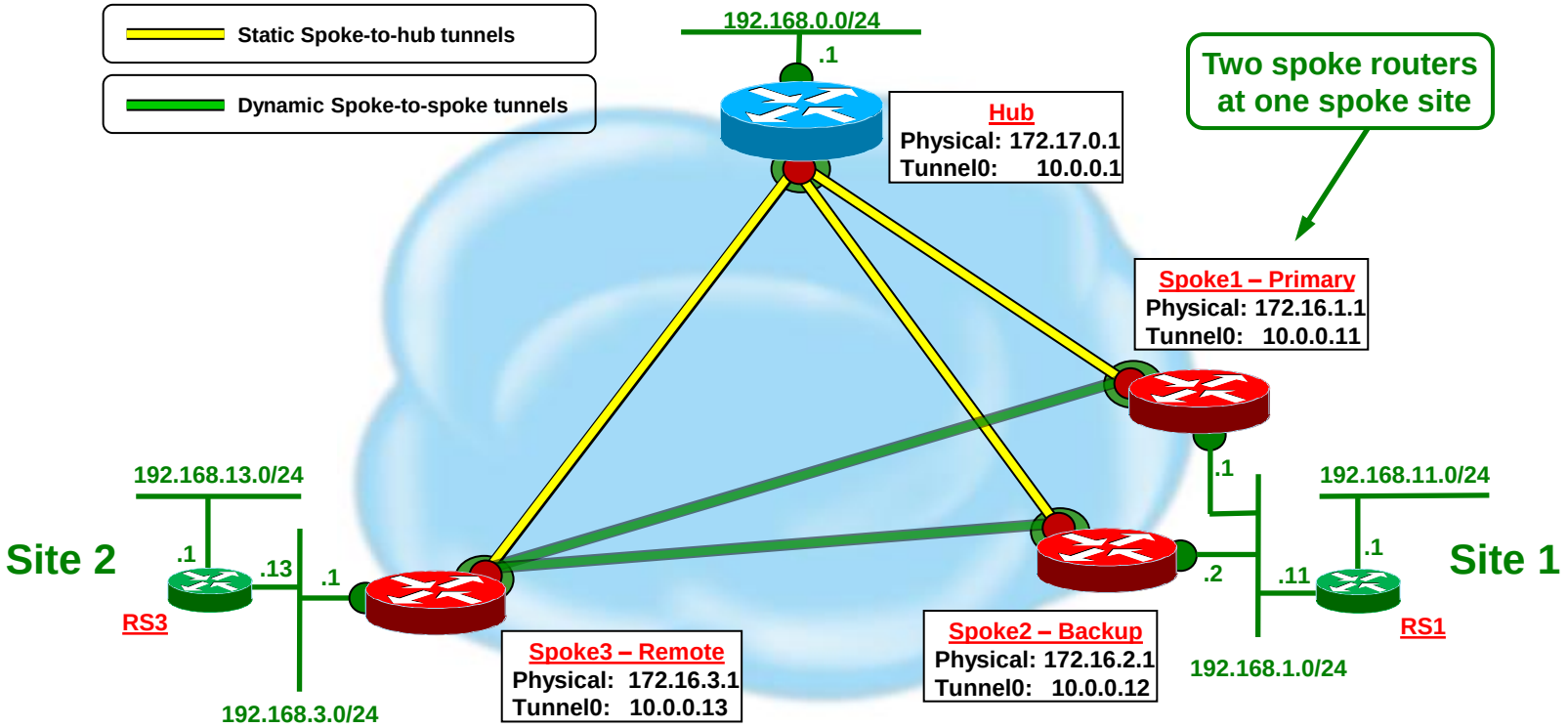
Agenda

- DMVPN Overview
- NHRP Details
- Spoke Site Redundancy Use Case
  - The Problem
  - The Solution using DMVPN Smart-spoke

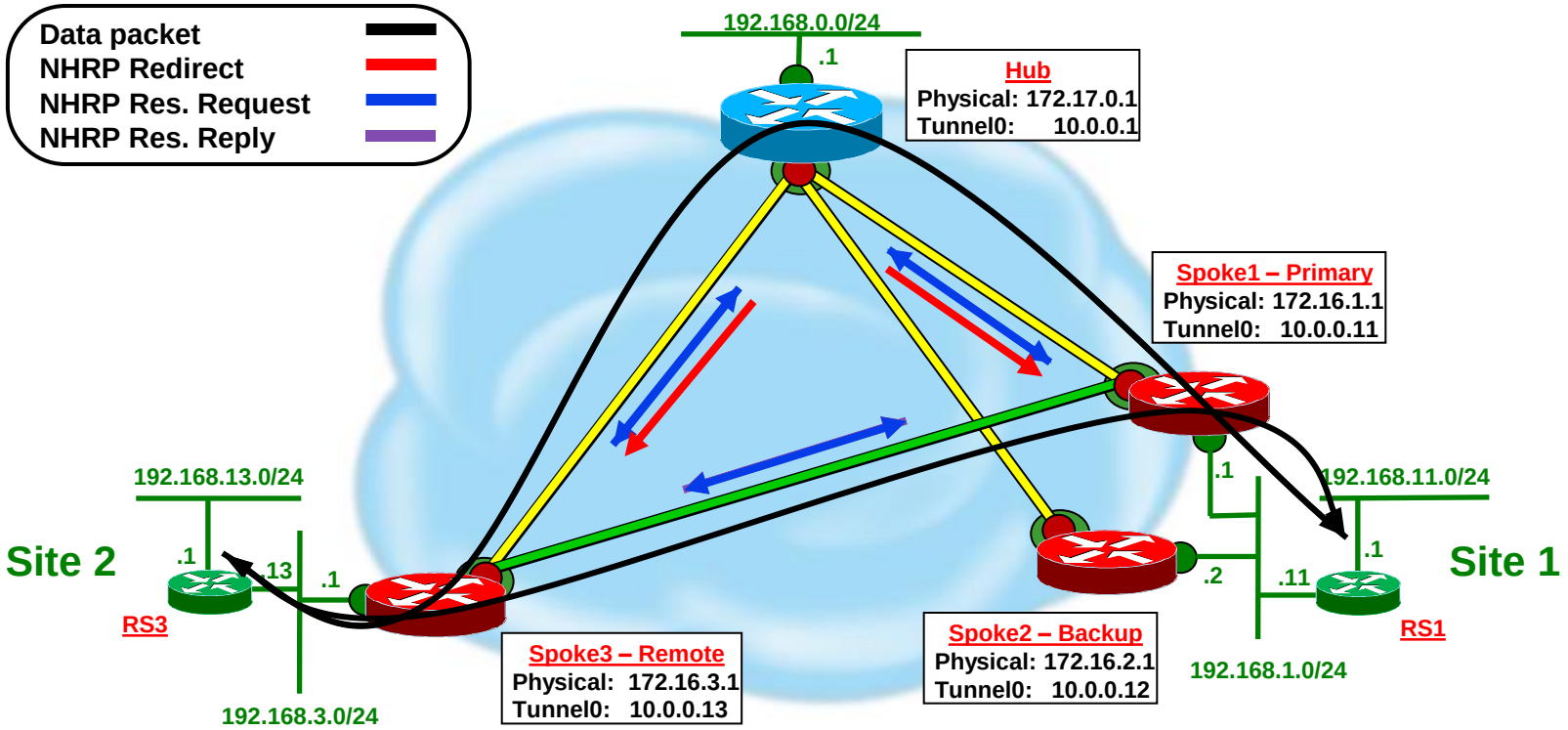
DMVPN Redundant Spoke

- Network Layout and Requirements
  - Standard DMVPN Phase 3
  - Single layer hub-and-spoke with dynamic spoke-spoke capabilities
  - Two DMVPN spoke routers at each spoke site for redundancy
    - Primary: Higher speed link, to be used if up
    - Backup: Lower speed link to be used only if Primary not available
  - EIGRP routing protocol used to prefer Primary over Backup path

DMVPN Redundant Spoke Network Diagram



DMVPN Redundant Spoke
Normal Operation



DMVPN Redundant Spoke
Normal Operation – Routing

Spoke1 - Primary

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.11/32 is directly connected, Tunnel0
D    192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
     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    192.168.11.0/24 [90/307200] via 192.168.1.2, Ethernet0/0
H    192.168.13.0/24 [250/1] via 10.0.0.13, Tunnel0
```

Spoke2 - Backup

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.12/32 is directly connected, Tunnel0
D    192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, Ethernet0/0
L       192.168.1.129/32 is directly connected, Ethernet0/0
D    192.168.11.0/24 [90/320000] via 192.168.1.2, Ethernet0/0
```

Hub

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.1/32 is directly connected, Tunnel0
D    192.168.1.0/24 [90/2841600] via 10.0.0.11, Tunnel0
D    192.168.3.0/24 [90/2841600] via 10.0.0.13, Tunnel0
D    192.168.11.0/24 [90/2867200] via 10.0.0.11, Tunnel0
D    192.168.13.0/24 [90/2867200] via 10.0.0.13, Tunnel0
```

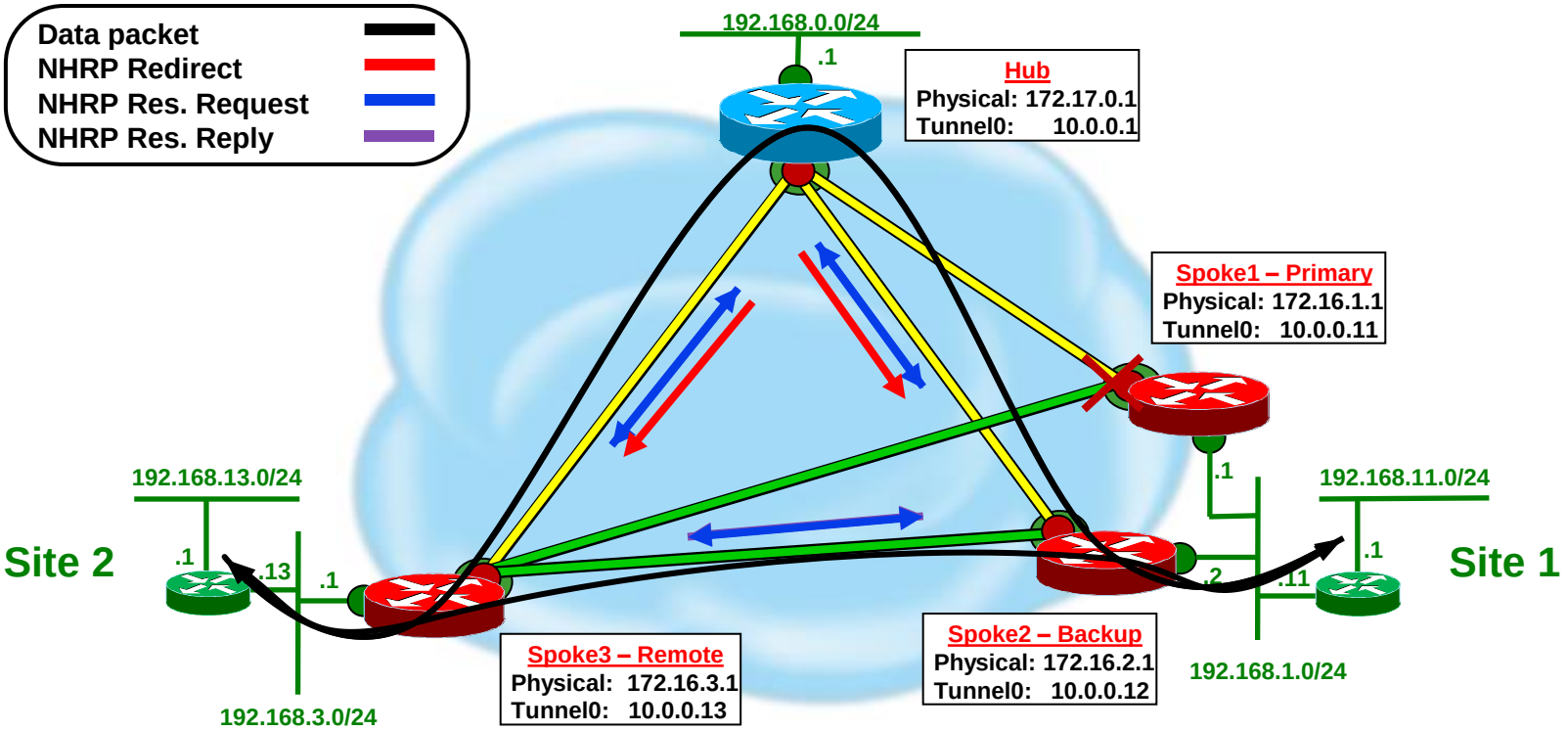
Spoke 3 - Remote

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.13/32 is directly connected, Tunnel0
D    192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, Ethernet0/0
L       192.168.3.1/32 is directly connected, Ethernet0/0
H    192.168.11.0/24 [250/1] via 10.0.0.11, Tunnel0
D    192.168.13.0/24 [90/307200] via 192.168.3.2, Ethernet0/0
```

RS1

```
C    192.168.11.0/24 is directly connected, Ethernet1/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
D    192.168.0.0/16 [90/2880000] via 192.168.1.1, Ethernet0/0
S*   0.0.0.0/0 [1/0] via 192.168.1.1
```

DMVPN Redundant Spoke Backup Operation



DMVPN Redundant Spoke
Backup Operation – Routing

Spoke1 - Primary

DOWN

Spoke2 - Backup

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.12/32 is directly connected, Tunnel0
D    192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, Ethernet0/0
L       192.168.1.129/32 is directly connected, Ethernet0/0
D    192.168.11.0/24 [90/320000] via 192.168.1.2, Ethernet0/0
H    192.168.13.0/24 [250/1] via 10.0.0.13, Tunnel0
```

Hub

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.1/32 is directly connected, Tunnel0
D    192.168.1.0/24 [90/2841600] via 10.0.0.12, Tunnel0
D    192.168.3.0/24 [90/2841600] via 10.0.0.13, Tunnel0
D    192.168.11.0/24 [90/2867200] via 10.0.0.12, Tunnel0
D    192.168.13.0/24 [90/2867200] via 10.0.0.13, Tunnel0
```

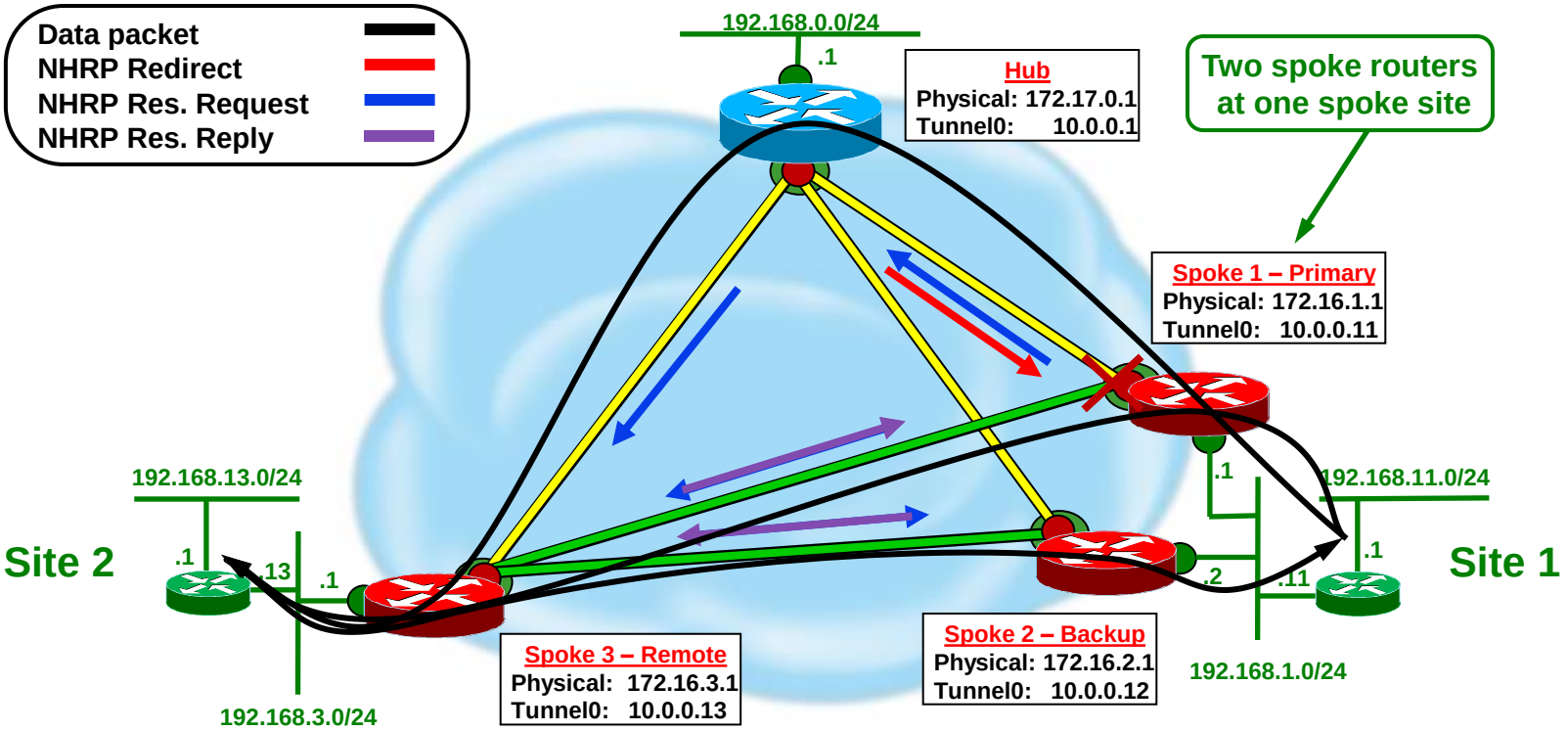
Spoke 3 - Remote

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.13/32 is directly connected, Tunnel0
D    192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, Ethernet0/0
L       192.168.3.1/32 is directly connected, Ethernet0/0
H    192.168.11.0/24 [250/1] via 10.0.0.12, Tunnel0
D    192.168.13.0/24 [90/307200] via 192.168.3.2, Ethernet0/0
```

RS1

```
C    192.168.11.0/24 is directly connected, Ethernet1/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
D    192.168.0.0/16 [90/2880000] via 192.168.1.129, Ethernet0/0
S*   0.0.0.0/0 [1/0] via 192.168.1.1
```

DMVPN Redundant Spoke Recovery Operation



DMVPN Redundant Spoke
Restored Operation – Routing

Spoke1 - Primary

DOWN

Spoke2 - Backup

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.12/32 is directly connected, Tunnel0
D    192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, Ethernet0/0
L       192.168.1.129/32 is directly connected, Ethernet0/0
D    192.168.11.0/24 [90/320000] via 192.168.1.2, Ethernet0/0
H    192.168.13.0/24 [250/1] via 10.0.0.13, Tunnel0
```

Hub

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.1/32 is directly connected, Tunnel0
D    192.168.1.0/24 [90/2841600] via 10.0.0.11, Tunnel0
D    192.168.3.0/24 [90/2841600] via 10.0.0.13, Tunnel0
D    192.168.11.0/24 [90/2867200] via 10.0.0.11, Tunnel0
D    192.168.13.0/24 [90/2867200] via 10.0.0.13, Tunnel0
```

Spoke 3 - Remote

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/24 is directly connected, Tunnel0
L       10.0.0.13/32 is directly connected, Tunnel0
D    192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, Ethernet0/0
L       192.168.3.1/32 is directly connected, Ethernet0/0
H    192.168.11.0/24 [250/1] via 10.0.0.12, Tunnel0
D    192.168.13.0/24 [90/307200] via 192.168.3.2, Ethernet0/0
```

RS1

```
C    192.168.11.0/24 is directly connected, Ethernet1/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
D    192.168.0.0/16 [90/2880000] via 192.168.1.1, Ethernet0/0
S*   0.0.0.0/0 [1/0] via 192.168.1.1
```

DMVPN Redundant Spoke
What is Happening?

- Normal Operation
  - Spoke-spoke tunnel between Remote and Primary routers
    - Bi-directional data traffic using this primary←→remote spoke-spoke tunnel
- Backup Operation (Primary goes down)
  - Path through Primary goes down
  - Hub routes through Backup
  - RS1 routes through Backup
  - Triggers new spoke-spoke tunnel backup←→remote
    - Bi-directional data traffic
- Recovery Operation (after Primary comes back up)
  - Adds spoke-spoke tunnel between Remote and Primary spoke routers
    - Forward data traffic uses primary→remote spoke-spoke tunnel
    - Return data traffic uses remote→backup spoke-spoke tunnel
  - Will continue as long as there is data traffic using remote→backup tunnel

DMVPN Redundant Spoke
What is Happening? (cont)

- Why Does this Happen?
  - Remote spoke is using its local RIB routes
    - Route added to RIB by NHRP points to backup as next-hop
      - Efficient data packet forwarding using CEF over remote→backup tunnel
      - Also used to route NHRP resolution request to refresh NHRP mapping
    - Spoke cannot “see” that routing has changed at remote site

- What do we need to do?
  - Need to have remote→backup tunnel drop
    - NHRP will then remove route via backup
    - Only backup can make this decision
  - Remote then reverts to using remote→primary tunnel
    - NHRP Resolution Request via Hub
    - NHRP Resolution Reply over primary→remote tunnel

Agenda

- DMVPN Overview
- NHRP Details
- Spoke Site Redundancy Use Case
  - The Problem
  - The Solution using DMVPN Smart-spoke

DMVPN Redundant Spoke
How do we do this?

- Backup spoke needs to “know” Primary is available
  - One way is to use a “flag” route from Hub
    - Only Primary will accept “flag” route over DMVPN tunnel
    - Primary will advertise “flag” route to Backup over local LAN
    - Backup “knows” Primary’s spoke-hub tunnel is up and working
  - Another way is to have the Backup probe the Primary
    - RSH/SSH session
    - IOS-SLA probe
    - Backup “knows” Primary is available, but not that Primary spoke-hub tunnel is working

- Use Smart-spoke feature to accept/reject spoke-spoke tunnel
  - Capture NHRP Resolution Request and hand to EEM Tcl script
  - Tcl script checks “flag” route and accepts/rejects NHRP Resolution Request

DMVPN Redundant Spoke
Add “flag” Route

- On Hubs

```
ip route 192.168.254.199 255.255.255.255 Null0
router eigrp 1
 ...
 network 192.168.254.199 0.0.0.0

Interface Tunnel0:
 ...
 ip summary-address eigrp 1 192.168.0.0 255.255.0.0 leak-map Leak_Flag

ip prefix-list Leak_Flag seq 10 permit 192.168.254.199/32
route-map Leak_Flag permit 10
 match ip address prefix-list Leak_Flag
```

May need to leak flag route if within summary

- On Backup spoke

```
router eigrp 1
 distribute-list prefix Block_Flag_Route in Tunnel0
 ...
ip prefix-list Block_Flag_Route seq 10 deny 192.168.254.199/32
ip prefix-list Block_Flag_Route seq 20 permit 0.0.0.0/0 le 32
```

Block flag route on Backup spoke on Tunnel interface

DMVPN Redundant Spoke "Flag" Route

Spoke1 - Primary

DOWN

Spoke2 - Backup

```
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.12/32 is directly connected, Tunnel0
D     192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.129/32 is directly connected, Ethernet0/0
D     192.168.11.0/24 [90/320000] via 192.168.1.2, Ethernet0/0
D     192.168.254.199 [90/2854400] via 192.168.1.1, Ethernet0/0
```

Hub

```
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.1/32 is directly connected, Tunnel0
D     192.168.1.0/24 [90/2841600] via 10.0.0.11, Tunnel0
D     192.168.3.0/24 [90/2841600] via 10.0.0.13, Tunnel0
D     192.168.11.0/24 [90/2867200] via 10.0.0.11, Tunnel0
D     192.168.13.0/24 [90/2867200] via 10.0.0.13, Tunnel0
S     192.168.254.199/32 is directly connected, Null0
```

Spoke 3 - Remote

```
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.13/32 is directly connected, Tunnel0
D     192.168.0.0/16 [90/2854400] via 10.0.0.1, Tunnel0
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, Ethernet0/0
L        192.168.3.1/32 is directly connected, Ethernet0/0

D     192.168.13.0/24 [90/307200] via 192.168.3.2, Ethernet0/0
```

RS1

```
C     192.168.11.0/24 is directly connected, Ethernet1/0
C     192.168.1.0/24 is directly connected, Ethernet0/0
D     192.168.0.0/16 [90/2880000] via 192.168.1.1, Ethernet0/0
D     192.168.254.199 [90/2841600] via 192.168.1.1, Ethernet0/0
S*    0.0.0.0/0 [1/0] via 192.168.1.1
```

DMVPN Smart-spoke Feature

- Implemented in IOS 15.2(2)T
  - Allows calling an EEM Tcl script for processing an NHRP resolution request
  - Request can be accepted or rejected
    - Any information available to EEM can be used for this decision
    - Includes information from the NHRP resolution request
- Turning on DMVPN Smart-spoke
  - Configure on tunnel interface
    - At least one (name, value) pair to be passed in NHRP resolution
      - ip nhrp attribute set *name value*
    - NHRP event publisher
      - nhrp event-publisher max-event-timeout *value*
  - Configure in global
    - Extended Event Manager (EEM)
      - event manager directory user policy "*dir*:/"
      - event manager policy *tcl-script-name* type user

Must be on both initiator and responder

NHRP event publisher only supports Tcl scripts

DMVPN Smart-spoke Feature Configuration

Spoke1 (Primary)

```
interface Tunnel0
 …
 ip address 10.0.0.11 255.255.255.0
 …
 ip nhrp attribute set Node PRIMARY
```

Set attributes (type, value)

Spoke2 (Backup)

```
interface Tunnel0
 …
 ip address 10.0.0.11 255.255.255.0
 …
 ip nhrp attribute set Node BACKUP
 nhrp event-publisher max-event-timeout 5
!
...
event manager directory user policy “nvram:/"
event manager policy NHRP_Res-req.tcl type user
```

Set attributes (type, value)
Enable NHRP EEM events

Set EEM directory and load TcL script into EEM

Spoke3 (Remote)

```
interface Tunnel0
...
ip address 10.0.0.13 255.255.255.0
…
ip nhrp attribute set Node PRIMARY
```

Set attributes (type, value)

DMVPN Smart-spoke Feature
EEM TCL Script

- Information passed in via arr_einfo array

```
# General EEM attributes
event_trigger_num <num>      event_id <num>                    job_id <num>

event_pub_sec <sec>          event_pub_msec <msec>             event_pub_time <sec.msec>
event_type {341}             event_type_string  {nhrp}         event_severity {severity-normal}

Type {1}                     event_type {NHRP-EVENT-RES}       reqid {<reqid-num>}

# NHRP Resolution Request attributes
idb_name {<tunnel-if>}       src_addr {<req-src>}   dst_addr  {<req-dest>}   event_id {<event-id-num>}

# Local NHRP Tunnel/Node (type, value) attributes
l_attr_num {<num>}
lattr_type0 {<name>}         l_type_len0 {<len>}    lattr_val0 {<value>}    l_val_len0 {<len>}

# Remote_NHRP Tunnel/Node (type, value) attributes
r_attr_num {<num>}
rattr_type0 {<name>}         r_type_len0 {<len>}    rattr_val0 {<value>}    r_val_len0 {<len>}
```

DMVPN Smart-spoke Feature
EEM TCL Script (cont)

```
::cisco::eem::event_register_nhrp type 1

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

puts "Executing NHRP EEM Script"

array set arr_einfo [event_reqinfo]
     foreach item [array names arr_einfo] { set $item $arr_einfo($item) }

if [catch {cli_open} result] { error $result $errorInfo } else { array set cli1 $result }
if [catch {cli_exec $cli1(fd) "enable"} _cli_result] { error $_cli_result $errorInfo }

set reject-sec “10”;  set flag-route “192.168.254.199”; set next-hop “192.168.1.1”

if [catch {cli_exec $cli1(fd) "show ip route | section $flag-route"} _cli_result] { error $_cli_result $errorInfo }
set _regexp_result [regexp “$flag-route.* via $next-hop,” "$_cli_result" ignore]

if [catch {cli_exec $cli1(fd) "configure terminal"} _cli_result] { error $_cli_result $errorInfo }
if [catch {cli_exec $cli1(fd) "interface $idb_name"} _cli_result] { error $_cli_result $errorInfo }

if {$_regexp_result} {
     if [catch {cli_exec $cli1(fd) "ip nhrp reject $event_id $reject_sec"} _cli_result] { error $_cli_result $errorInfo }
} else {
     if [catch {cli_exec $cli1(fd) "ip nhrp connect $event_id"} _cli_result] { error $_cli_result $errorInfo }
}

catch {cli_close $cli1(fd) $cli1(tty_id)} result
```

Put arr_einfo array elements into local variables

Open CLI and enable

Check for special flag route

Connect or Reject

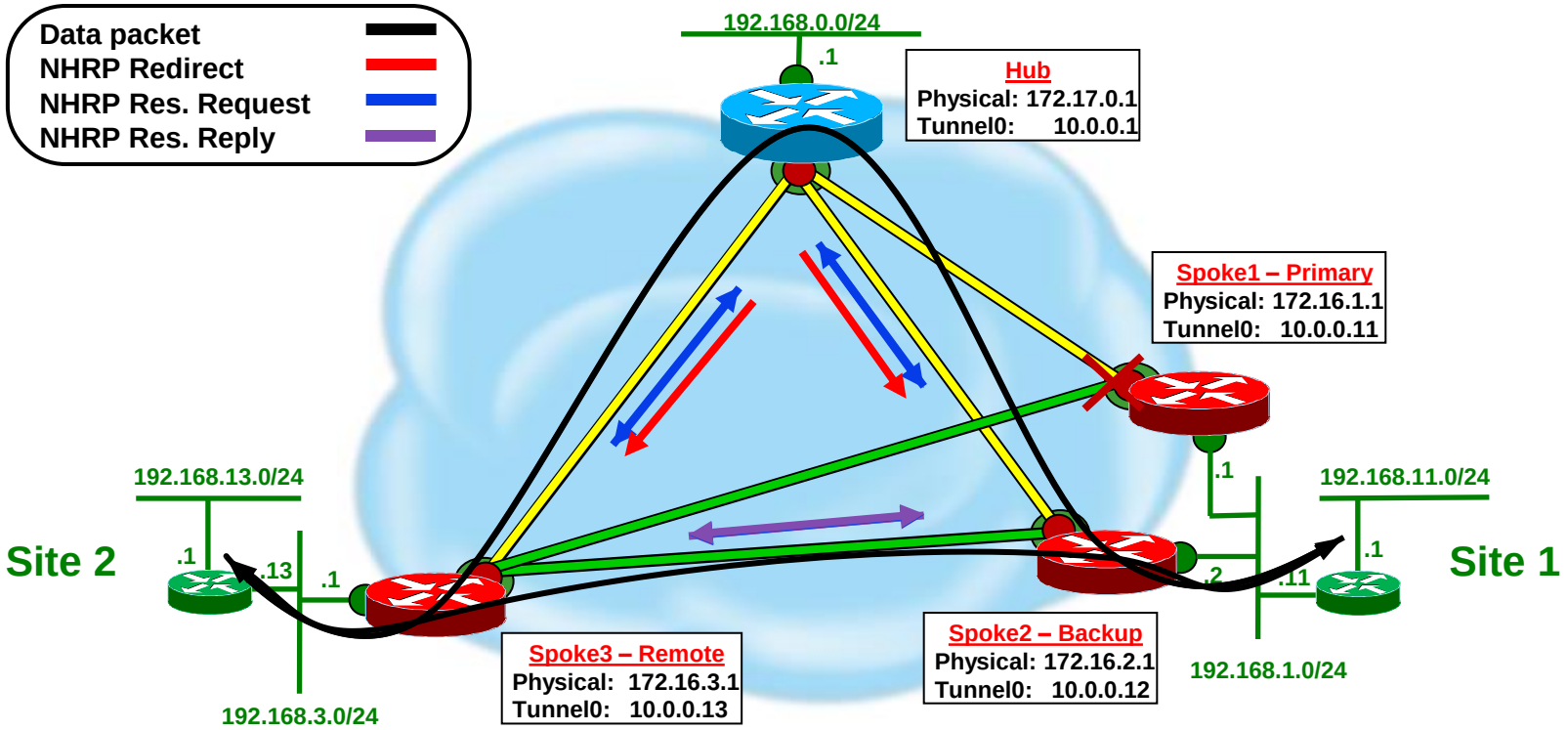
Close CLI and exit

DMVPN Redundant Spoke
How do we do this? (cont)

- If “flag” route is not in RIB on backup
  - NHRP resolution request is accepted by backup
    - Remote will build or refresh remote→backup tunnel

- If “flag” route is in RIB on backup
  - Remote’s NHRP resolution request is rejected by backup
    - Remote will time out remote→backup tunnel
    - Remote then sends data packets via Hub, and Hub will send back an NHRP Redirect
    - Remote will send NHRP Resolution Request via Hub and Hub will forward it to the Primary
    - Primary will send NHRP Resolution Reply to Remote via primary→remote tunnel

DMVPN Redundant Spoke
Backup Operation



DMVPN Redundant Spoke Backup Operation

- Spoke2 (backup) accepts NHRP resolution request

```
%HA_EM-6-LOG: NHRP_Res-req.tcl: Executing NHRP EEM Script
%HA_EM-6-LOG: NHRP_Res-req.tcl: Event_id: 182, Int: Tunnel0, Src: 10.0.0.13, Dst: 192.168.11.1
%HA_EM-6-LOG: NHRP_Res-req.tcl: Local_attr: (T:Node, V:BACKUP); Remote_attr: (T:Node, V:PRIMARY)

%HA_EM-6-LOG: NHRP_Res-req.tcl:ip nhrp connect 182
```

- Spoke2 builds shortcut tunnel

```
Spoke1# show ip nhrp

Spoke1# show ip route nhrp
```

```
Spoke2# show ip nhrp 192.168.13.0

192.168.13.0/24 via 10.0.0.13
  Tunnel0 created 00:03:03, expire 00:01:56
  Type: dynamic, Flags: router rib
  NBMA address: 172.18.0.13

Spoke2# show ip route nhrp

H    192.168.13.0/24 [250/1] via 10.0.0.13, 00:02:25, Tunnel0
```

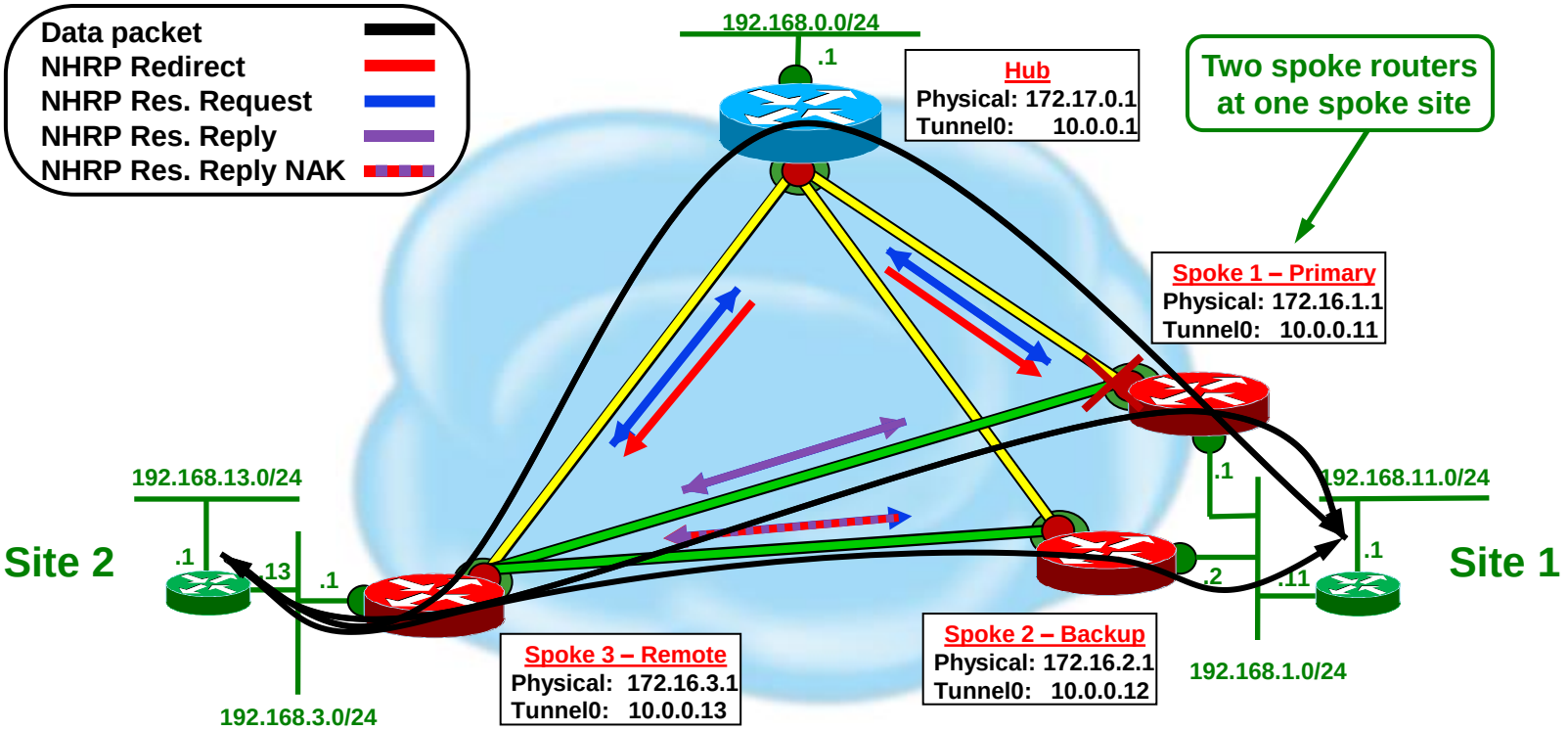
```
Spoke3# show ip nhrp 192.168.11.0

192.168.11.0/24 via 10.0.0.12
  Tunnel0 created 00:07:46, expire 00:01:34
  Type: dynamic, Flags: router rib
  NBMA address: 172.16.2.1

Spoke3# show ip route nhrp

H    192.168.11.0/24 [250/1] via 10.0.0.12, 00:07:54, Tunnel0
```

DMVPN Redundant Spoke Recovery Operation



DMVPN Redundant Spoke Recovery Operation

- Spoke2 (backup) rejects NHRP resolution request

```
%HA_EM-6-LOG: NHRP_Res-req.tcl: Executing NHRP EEM Script
%HA_EM-6-LOG: NHRP_Res-req.tcl: Event_id: 191, Int: Tunnel0, Src: 10.0.0.13, Dst: 192.168.11.1
%HA_EM-6-LOG: NHRP_Res-req.tcl: Local_attr: (T:Node, V:BACKUP); Remote_attr: (T:Node, V:PRIMARY)
%HA_EM-6-LOG: NHRP_Res-req.tcl: 192.168.254.199 [90/2854400] via 192.168.1.1,
%HA_EM-6-LOG: NHRP_Res-req.tcl: ip nhrp reject 191 10
```

- Spoke3 times out shortcut tunnel and reverts back to Spoke1

```
Spoke2# show ip nhrp 192.168.13.0

Spoke2# show ip route nhrp
```

```
Spoke1# show ip nhrp 192.168.13.0

192.168.13.0/24 via 10.0.0.13
  Tunnel0 created 00:13:08, expire 00:04:02
  Type: dynamic, Flags: router rib
  NBMA address: 172.18.0.13

Spoke1# show ip route nhrp

H    192.168.13.0/24 [250/1] via 10.0.0.13, 00:13:15, Tunnel0
```

```
Spoke3# show ip nhrp 192.168.11.0

192.168.11.0/24 via 10.0.0.11
  Tunnel0 created 00:03:18, expire 00:01:49
  Type: dynamic, Flags: router rib
  NBMA address: 172.16.1.1

Spoke3# show ip route nhrp

H    192.168.11.0/24 [250/1] via 10.0.0.11, 00:03:16, Tunnel0
```

DMVPN Redundant Spoke
Summary

- DMVPN Smart-spoke with EEM Tcl Script
  - Used to make more complex decisions when building/refreshing tunnels
  - Could do much more with it
    - Only build primary←→primary and backup←→backup tunnels
    - Use to instantiate dynamic spoke-spoke QoS policy
    - More?

- Can also use regular EEM scripts to do other functions
  - Reset tunnel source address to go out over backup path
  - Install static NHRP mappings dynamically for GRE only spokes
  - Turn on debugging right after an event occurs
  - Many other functions

DMVPN in an SDN world

- The dynamic nature of DMVPN fits
  - Dynamically creating tunnels/paths as needed
  - Smart distributed control plane

- With features like Smart-spoke and EEM
  - Can do more SDN like functions

- Combine with central policy control, **BUT**
  - Need a smart distributed network to implement central policies
  - Network **must** keep functioning **when** access to central policy control is lost

DMVPN Recent and Future Features

DMVPN Recent and Future Features

- Recent
  - ASR (3.8S)
    - Dual Stack (IPv4 & IPv6) over DMVPN IPv6
    - IPv6 IPsec in VRF
    - Suite-B Data Plan (ASR1002-X/ESP100 only)
  - CSR 1000v DMVPN initial support
- Roadmap
  - Per-tunnel QoS IPv6 over DMVPN on Hub
  - TrustSec over DMVPN
  - 2547oDMVPN true spoke-spoke support
  - IPsec anti-replay window 1024 (ASR)
- Possible
  - Native Multicast over DMVPN
  - BFD for mGRE tunnels

Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Cisco Daily Challenge points for each session evaluation you complete.
- Complete your session evaluation online now through either the mobile app or internet kiosk stations.

Maximize your Cisco Live experience with your free Cisco Live 365 account. Download session PDFs, view sessions on-demand and participate in live activities throughout the year. Click the Enter Cisco Live 365 button in your Cisco Live portal to log in.

Note: This slide is now a Layout choice


CISCO
TM

Extras




Recent and New Features

Extras
Recent and New Features

- IKEv2 with DMVPN
- Tunnel Health Monitoring
- Backup and FQDN NHS
- DHCP over DMVPN
- DMVPN IPv6 Transport
- Routing protocol
- Per-tunnel QoS

IKEv2 with DMPVN

- DMVPN can work 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
```

Tunnel Health Monitoring
Interface State – 15.0(1)M

- 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

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

```
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
```

DHCP over DMVPN – 15.1(3)T

- Issue
  - Must pre-configure tunnel interface IP Address and Subnet on Spokes
- Solution
  - Use DHCP to allocate Spoke's Tunnel IP Address/Subnet
    - ip address dhcp
    - ip dhcp client broadcast-flag clear
  - Hub is DHCP Relay Agent
    - Global
      - ip dhcp support tunnel unicast
    - Tunnel Interface
      - ip helper-address *ip-dhcp-server*
  - Functionality
    - DHCP request broadcast to all NHSs, replies unicast back to Spoke
    - Sticky until tunnel interface goes down

DHCP and FQDN NHS Example:

Spoke:

```
interface Tunnel0
 ip dhcp client broadcast-flag clear
 ip address dhcp
 …
 ip nhrp network-id 100000
 …
 ip nhrp nhs dynamic nbma Hub1-NBMA multicast
 …
 ip nhrp shortcut
 tunnel source Serial1/0
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
```

Hub:

```
ip dhcp support tunnel unicast
!
interface Tunnel0
 ip address 10.0.0.1 255.255.255.0
 ip helper-address 192.168.0.3
 …
 ip nhrp map multicast dynamic
 ip nhrp network-id 100000
 ip nhrp redirect
 tunnel source Serial2/0
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
```

DHCP:

```
22:52:32.658: DHCP: Starting DHCP discover on Tunnel0
22:52:32.658: DHCP: SDiscover attempt # 1 for entry:
22:52:32.658:    Hostname: Spoke1, B'cast on Tunnel0 interface from 0.0.0.0

22:52:32.738: DHCP: Offer Message, Offered Address: 10.0.0.13
22:52:32.738: DHCP: Lease secs: 86400, Renewal secs: 43200, Rebind secs: 75600

22:52:32.738: DHCP: SRequest attempt # 1 for entry:
22:52:32.738:    Temp IP addr: 10.0.0.13  for peer on Interface: Tunnel0
22:52:32.738:    Temp  sub net mask: 255.255.255.0
22:52:32.738:    Hostname: Spoke1, B'cast on Tunnel0 interface from 0.0.0.0

22:52:32.818: DHCP: Ack Message Offered Address: 10.0.0.13
22:52:32.818: DHCP: Lease secs: 86400, Renewal secs: 43200, Rebind secs: 75600
22:52:32.818: DHCP: Host Name Option: Spoke1.cisco-test.com
```

DHCP and FQDN NHS Example: (cont)

NHRP:

```
22:52:32.242: NHRP: Resolved FQDN Hub1-NBMA to 172.17.0.1
22:52:32.242: NHRP: Supressing registration requests (Tunnel0) has invalid address
 . . .
22:52:32.818: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 104
22:52:32.818:      src NBMA: 172.16.1.1, src proto: 10.0.0.13, dst proto: 10.0.0.13
22:52:32.818:    NAT address Extension(9):  client NBMA: 172.17.0.1, client protocol: 10.0.0.13
 . . .
22:52:32.870: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 124
22:52:32.870:      src NBMA: 172.16.1.1, src proto: 10.0.0.13, dst proto: 10.0.0.1
22:52:32.870:    Responder Address Extension(3):  client NBMA: 172.17.0.1, client protocol: 10.0.0.1
22:52:32.870:    NAT address Extension(9):  client NBMA: 172.17.0.1, client protocol: 10.0.0.1

22:52:32.870: NHRP: Tu0: Creating nhs  mapping  for 10.0.0.1/32 NBMA: 172.17.0.1
22:52:32.870: NHRP: Tunnel0: Cache add for target 10.0.0.1/32 next-hop 10.0.0.1, 172.17.0.1

22:52:32.870: NHRP: Adding Tunnel Endpoints (VPN: 10.0.0.1, NBMA: 172.17.0.1)
```

Tunnel:

```
22:52:29.618: %LINK-3-UPDOWN: Interface Tunnel0, changed state to up
22:52:29.622: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
 . . .
22:52:32.870: Tunnel0: Linking endpoint 10.0.0.1/172.17.0.1
22:52:32.870: FIBtunnel: Tu0:TED: Adding adj for 10.0.0.1, conn_id 0
22:52:32.870: FIBtunnel: Tu0: stacking IP 10.0.0.1 to Default:172.17.0.1
 . . .
22:52:32.902: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.0.0.1 (Tunnel0) is up: new adjacency
```

DMVPN over IPv6 Transport – 15.2(1)T

- IPv6 and IPv4 packets over DMVPN IPv6 tunnels
  - Introduced in IOS 15.2(1)T, 15.3(1)S
  - IPv6 infrastructure network
  - IPv6 and/or IPv4 data packets over same IPv6 GRE tunnel
  - NHRP modifies Routing Table
- Can run both DMVPN IPv4 and IPv6
  - Separate DMVPNs (mGRE tunnel)
  - DMVPN IPv4 ←→ IPv6 spoke to spoke via hub
- Configuration
  - Standard IPv6 configuration on Outside (WAN) interface
  - Small change on mGRE tunnel
  - Must use IKEv2 for IPsec encryption
- Split-tunneling
  - Enterprise versus ISP assigned IPv6 addresses at spoke
  - No NAT66

DMVPN over IPv6 Transport
Configuration

Hub

```
crypto ikev2 keyring DMVPN
  peer DMVPNv6
    address ::/0
    pre-shared-key cisco123v6
crypto ikev2 profile DMVPN
  match identity remote address ::/0
  authentication local pre-share
  authentication remote pre-share
  keyring DMVPN
crypto ipsec profile DMVPN
  set transform-set DMVPN
  set ikev2-profile DMVPN
…
interface Tunnel0
  ip address 10.0.0.1 255.255.255.0
  …
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  …
  ipv6 address 2001:DB8:0:100::1/64
  …
  ipv6 nhrp map multicast dynamic
  ipv6 nhrp network-id 100006
  …
  tunnel source Serial2/0
  tunnel mode gre multipoint ipv6
  tunnel protection ipsec profile DMVPN
!
interface Serial2/0
  ip address 172.17.0.1 255.255.255.252
  ipv6 address 2001:DB8:0:FFFF:1::1/126
!
ipv6 route ::/0 Serial2/0
```

Spoke

```
crypto ikev2 keyring DMVPN
  peer DMVPNv6
    address ::/0
    pre-shared-key cisco123v6
crypto ikev2 profile DMVPN
  match identity remote address ::/0
  authentication local pre-share
  authentication remote pre-share
  keyring DMVPN
  dpd keepalive 30 5 on-demand
crypto ipsec profile DMVPN
  set transform-set DMVPN
  set ikev2-profile DMVPN
…
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 2001:DB8:0:FFFF:1::1 multicast
  …
  ipv6 address 2001:DB8:0:100::B/64
  …
  ipv6 nhrp network-id 100006
  ipv6 nhrp nhs 2001:DB8:0:100::1 nbma 2001:DB8:0:FFFF:1::1 multicast
  …
  tunnel source Serial1/0
  tunnel mode gre multipoint ipv6
  tunnel protection ipsec profile DMVPN
!
interface Serial1/0
  ip address 172.16.1.1 255.255.255.252
  ipv6 address 2001:DB8:0:FFFF:0:1:0:1/126
!
ipv6 route ::/0 Serial1/0
```

DMVPN over IPv6 Transport
Data Structures

```
Hub1# show ip nhrp
10.0.0.11/32 via 10.0.0.11
   Tunnel0 created 22:26:55, expire 00:03:37
   Type: dynamic, Flags: unique registered used
   NBMA address: 2001:DB8:0:FFFF:0:1:0:1
```

```
Hub1# show ipv6 nhrp
2001:DB8:0:100::B/128 via 2001:DB8:0:100::B
   Tunnel0 created 22:27:52, expire 00:03:39
   Type: dynamic, Flags: unique registered
   NBMA address: 2001:DB8:0:FFFF:0:1:0:1
FE80::A8BB:CCFF:FE00:C800/128 via 2001:DB8:0:100::B
   Tunnel0 created 22:27:52, expire 00:03:39
   Type: dynamic, Flags: unique registered
   NBMA address: 2001:DB8:0:FFFF:0:1:0:1
```

```
Hub1# show crypto session
Interface: Tunnel0;  Session status: UP-ACTIVE
Peer: 2001:DB8:0:FFFF:0:1:0:1 port 500
  IKEv2 SA: local 2001:DB8:0:FFFF:1::1/500
        remote 2001:DB8:0:FFFF:0:1:0:1/500 Active
  IPSEC FLOW: permit 47 host 2001:DB8:0:FFFF:1::1 host 2001:DB8:0:FFFF:0:1:0:1
      Active SAs: 2, origin: crypto map
```

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
    - Also use BGP Dynamic Neighbors to reduce configuration on hub

```
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
    ...
```

Per-tunnel QoS – 12.4(22)T

- QoS per tunnel (spoke) on hub
  - Dynamically selected Hierarchical (parent/child) QoS Policy
    - **Spoke:** Configure NHRP group name
    - **Hub:** NHRP group name mapped to QoS template policy
  - Spokes with same NHRP group name are mapped to individual instances of the same QoS template policy
- 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
  - Can't have separate aggregate QoS policy on physical
- CPU intensive; reduces hub scaling by about 50%

Per-tunnel QoS Configurations

Hub

```
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 (cont)

```
interface Tunnel0
   ip address 10.0.0.1 255.255.255.0
   …
   ip nhrp map group typeA service-policy output typeA_parent
   ip nhrp map group typeB service-policy output typeB_parent
   …
   ip nhrp redirect
   no ip split-horizon eigrp 100
   ip summary-address eigrp 100 192.168.0.0 255.255.192.0 5
   …
```

Spoke1

```
interface Tunnel0
   ip address 10.0.0.11 255.255.255.0
   …
   ip nhrp group typeA
   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
   …
```

Spoke2

```
interface Tunnel0
   ip address 10.0.0.12 255.255.255.0
   …
   ip nhrp group typeB
   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
   …
```

Spoke3

```
interface Tunnel0
   ip address 10.0.0.13 255.255.255.0
   …
   ip nhrp group typeA
   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
   …
```

Per-tunnel QoS
QoS Output

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
```