



CCNP SECURITY - SVPN -350-730



Dynamic Multi-point VPN

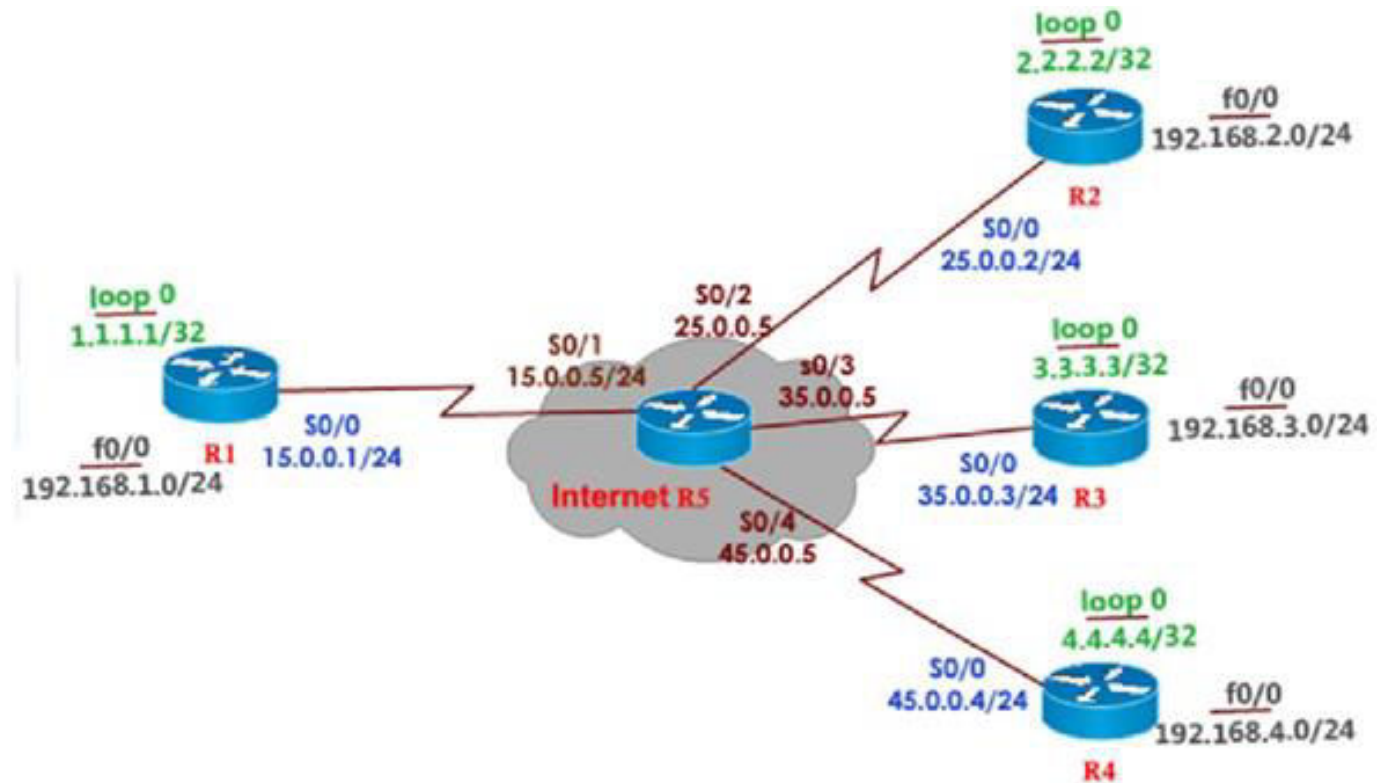
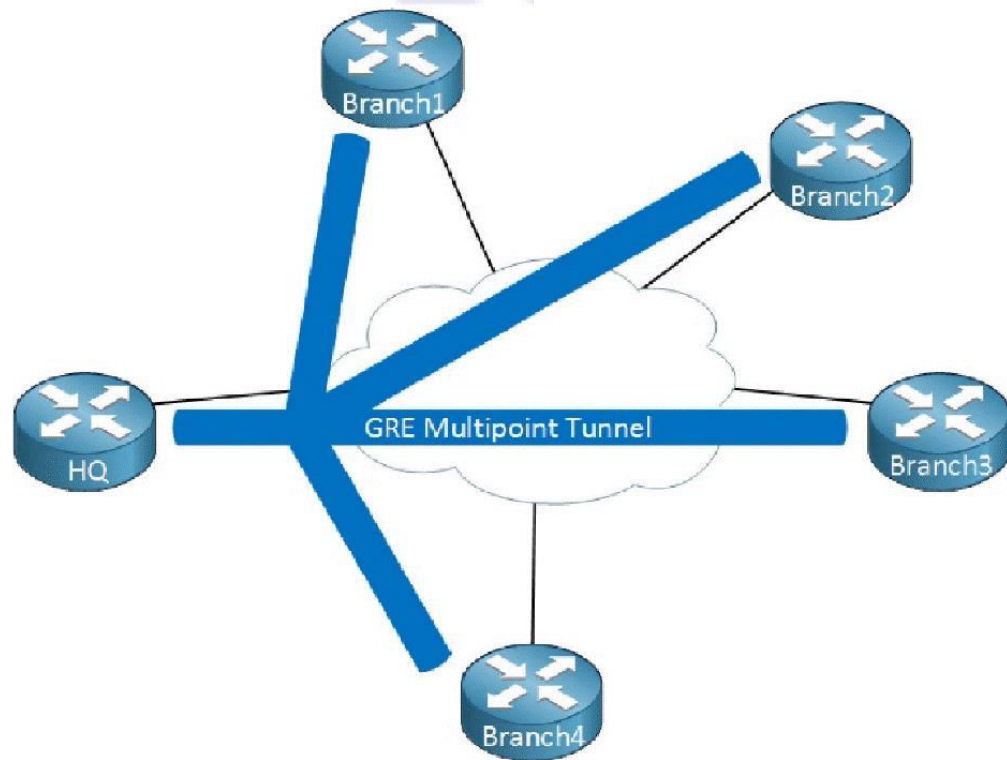
Dynamic Multi-point VPN

- **DMVPN- Protocols CCNP**
- **mGRE**
- **NHRP**
- **DMVPN Phases - 1 2 3**

Dynamic Multi Point VPN

- Introduced by Cisco in late 2000.
- Developed to address needs for automatically created VPN tunnels (Full mesh)
- Supports Full mesh Tunnels build dynamically (using mGRE)
- Peers discovers each other dynamically (using NHRP)
- Spokes can be dynamic IP on exit interfaces.
- Keeping costs low, minimizing configuration complexity and increasing flexibility & Scalability.

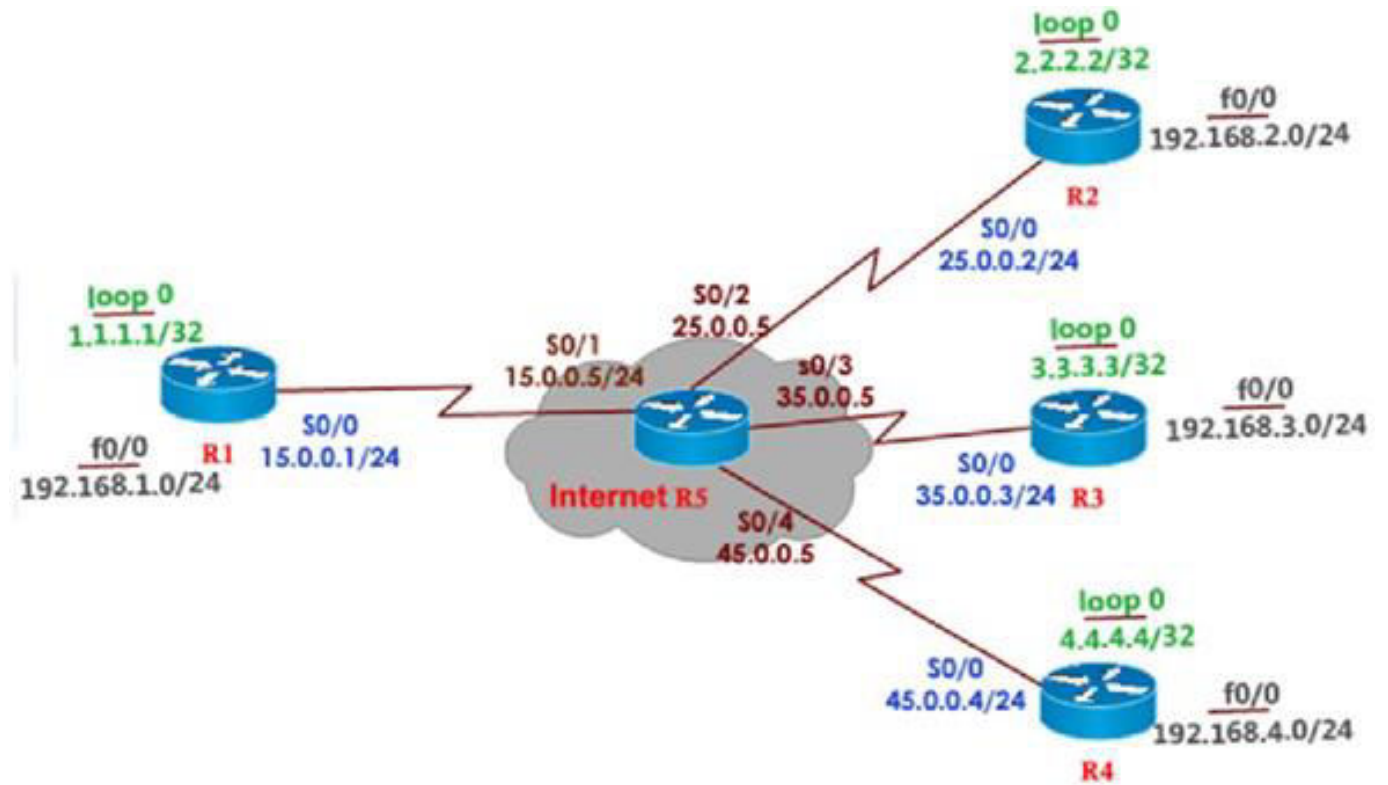
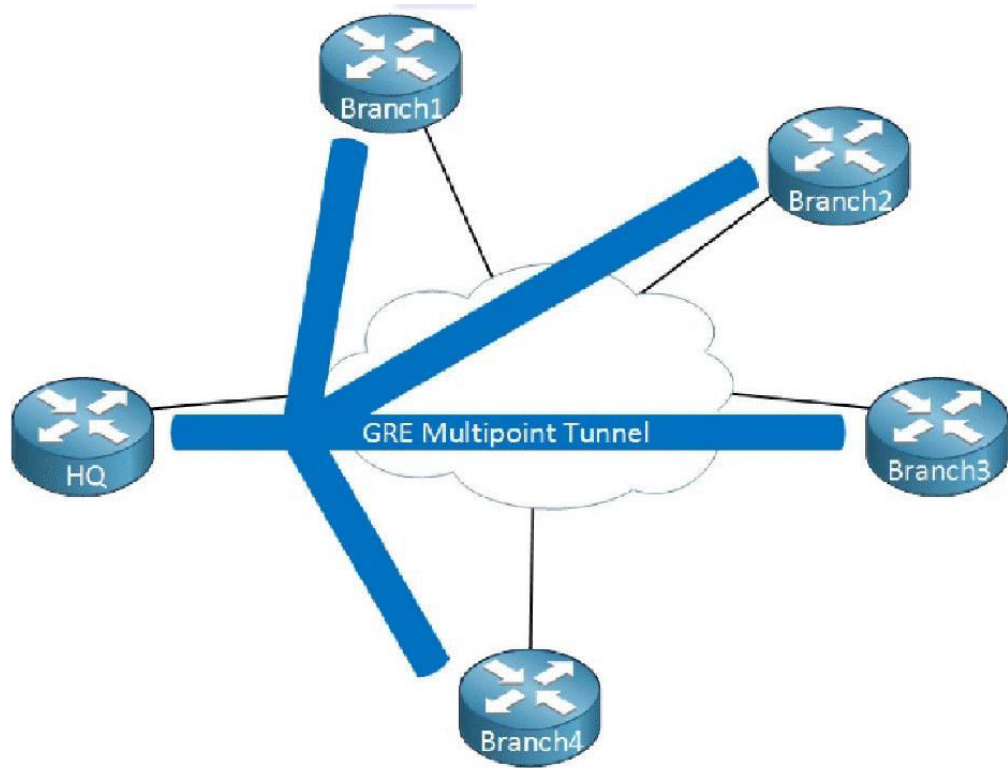
Dynamic Multi Point VPN



DMVPN - Protocols

- DMVPN is combination of the following technologies:
- Multipoint GRE (mGRE) - Allows P2M tunnels full mesh
- Next-Hop Resolution Protocol (NHRP) - Learn peer information dynamically
- Dynamic Routing Protocol (EIGRP, RIP, OSPF, BGP) - LAN to LAN Communication
- Dynamic IPsec encryption - to Secure tunnel traffic

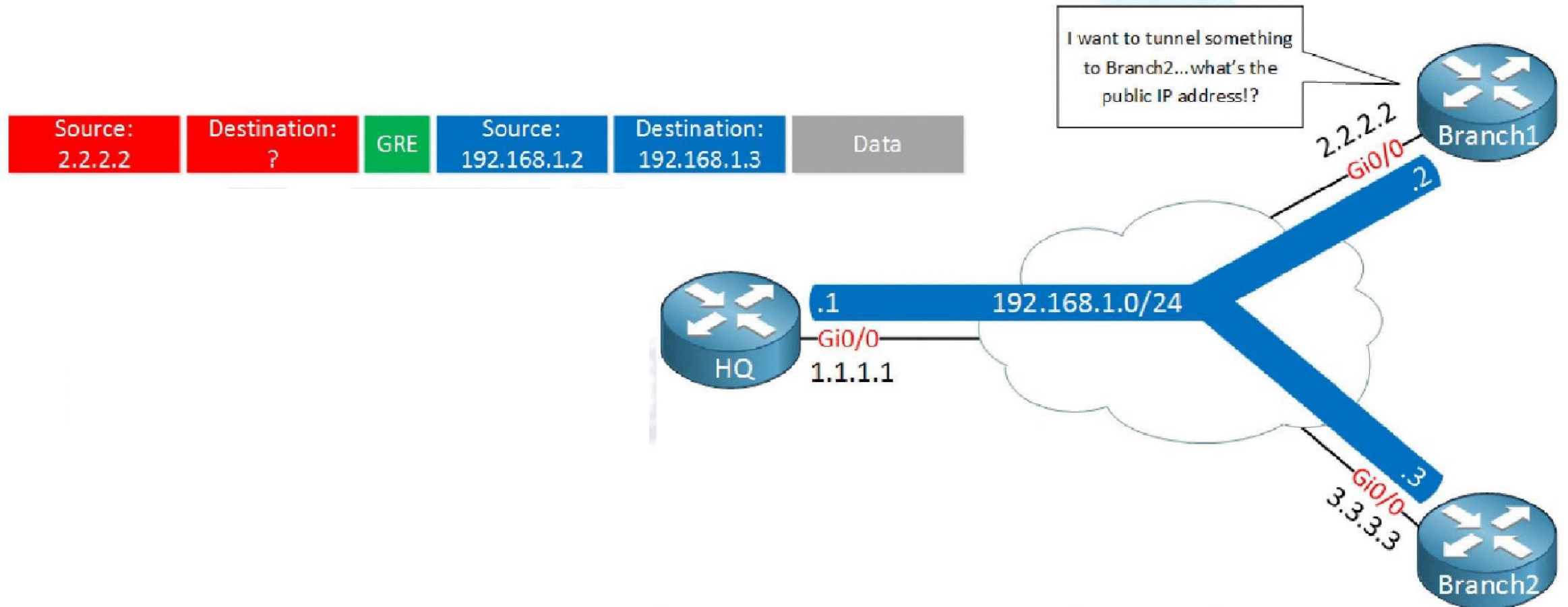
DMVPN - Protocols



DMVPN - NHRP protocol

- When two branch routers want to tunnel some traffic, how do they know what IP addresses to use?
- Resolves the tunnel IP with NBMA address (public IP)
- Similar to ARP on Ethernet networks (MAC - IP)
- Build a dynamic database stored on the hub with information about spokes' IP Addresses.

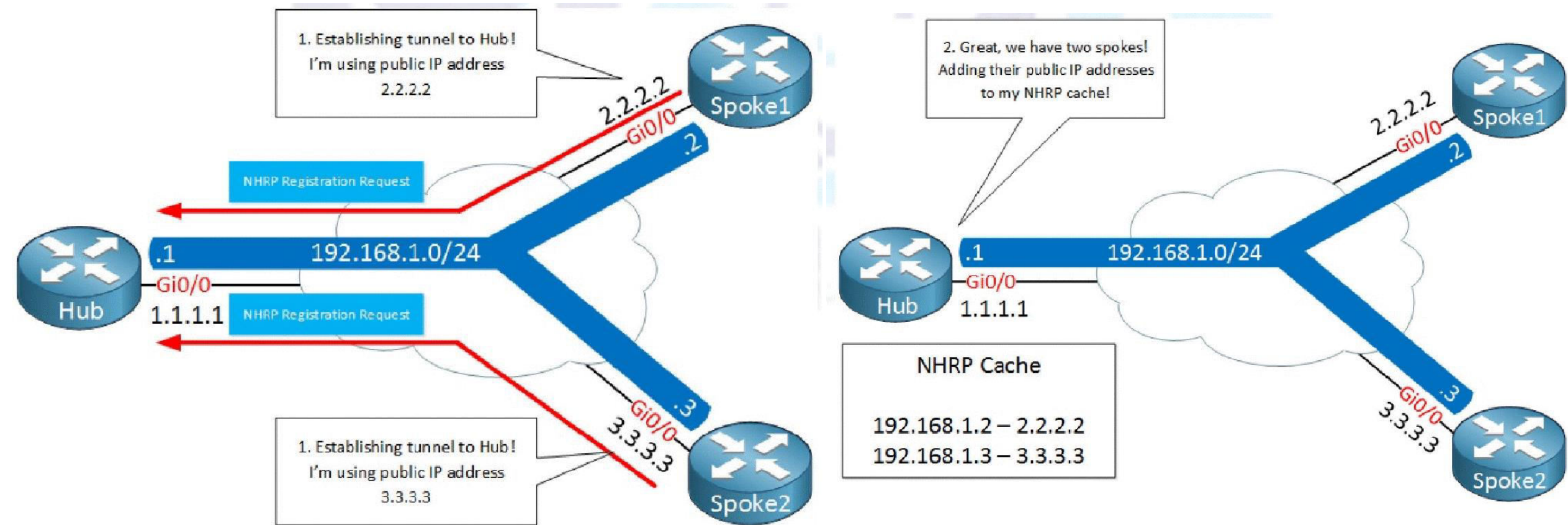
DMVPN - NHRP protocol



DMVPN - NHRP protocol

- Routers can be configured as
 - Next hop servers (NHS)
 - Next hop Clients (NHC)
- NHRP clients register themselves with the NHRP server and report their public IP address.
- The NHRP server keeps track of all public IP addresses in its cache.
- NHC send query to the NHS if they want to communicate with another NHC.
- NHS reply to queries made by NHC.

Generic Routing Encapsulation(GRE)



DMVPN - NHRP Messages

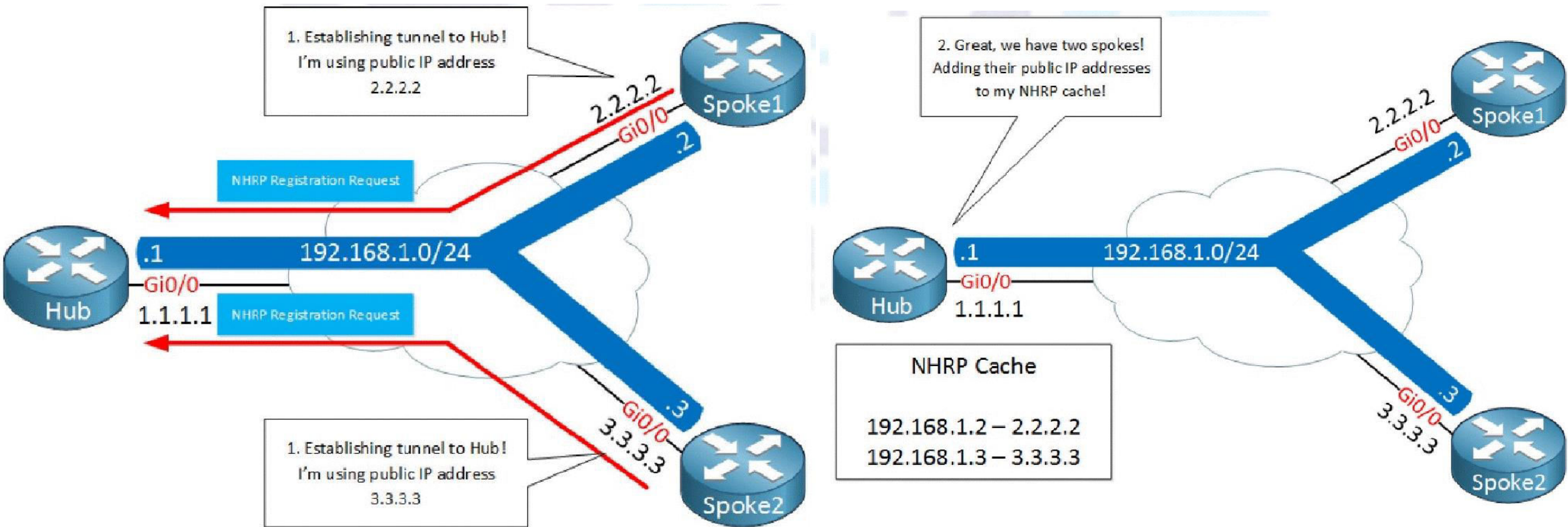
NHRP Registration Request

- Spoke register with NBMA and tunnel IP to NHS.
- Required to build spoke to Hub Tunnels

NHRP Registration Reply

- Hub Acknowledges the registration request.
- Required to tell spokes that tunnel with Hub was formed

DMVPN - NHRP Messages



DMVPN - NHRP Messages

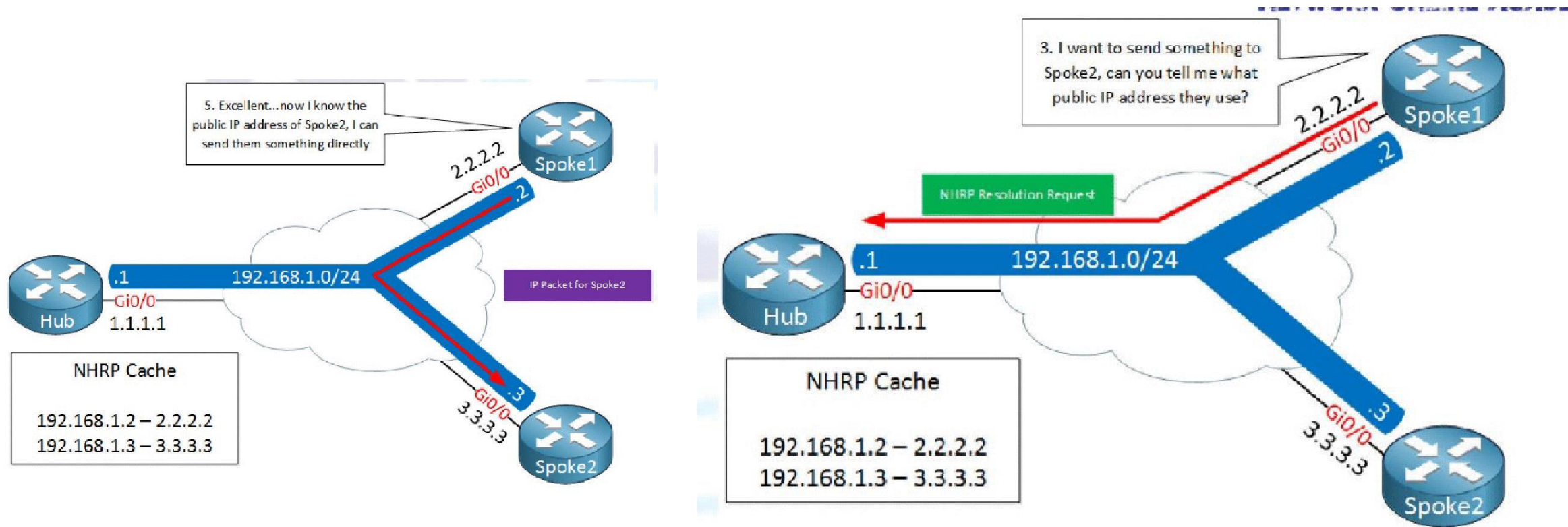
NHRP Resolution Request

- Spoke query for NBMA and tunnel IP of other spokes.
- Required to build spoke to spoke Tunnels.

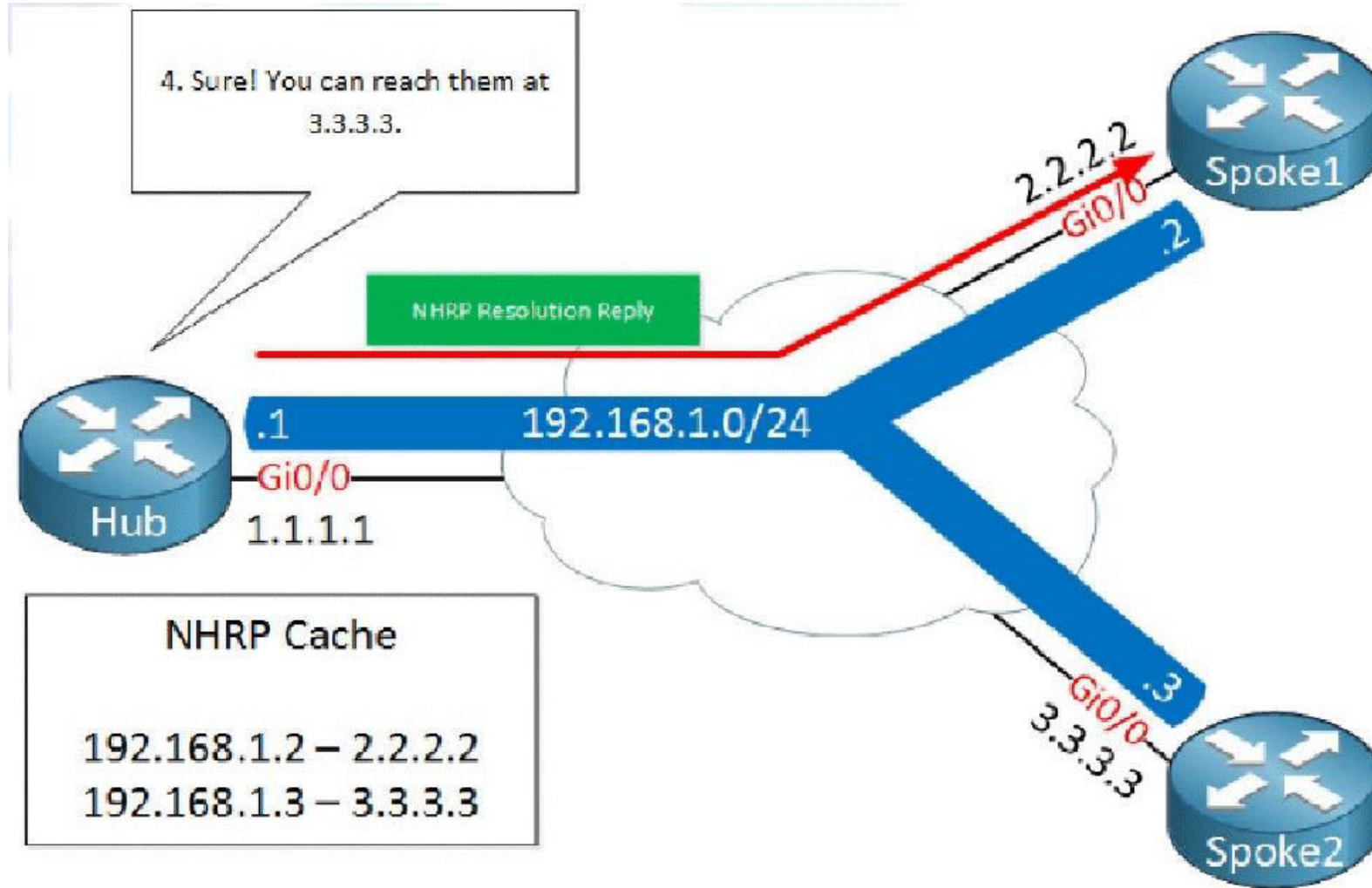
NHRP Resolution Reply

- Queried Device reply to the resolution request
- Required to build spoke to spoke Tunnels.

DMVPN - NHRP Messages



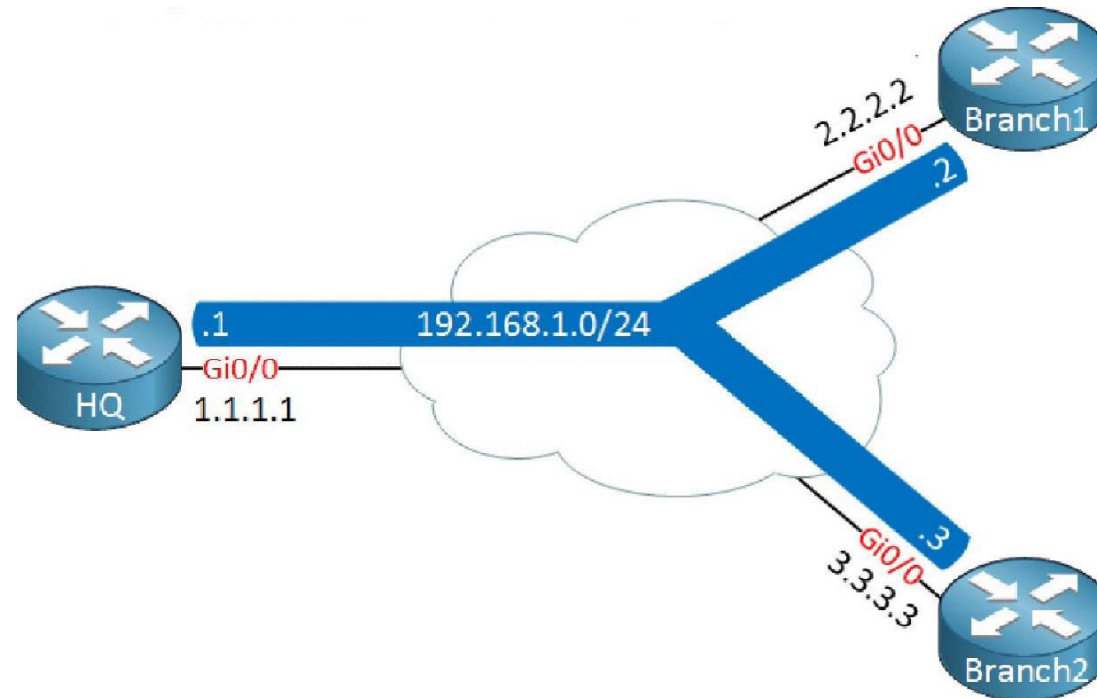
DMVPN - NHRP Messages



DMVPN - NHRP Messages

NHRP Redirect (used in Phase-3)

- Hub(NHS) answers spoke to spoke data plane packet through it.
- Used in DMVPN phase3 to build spoke to spoke tunnels.(needed if we have spoke to spoke traffic)

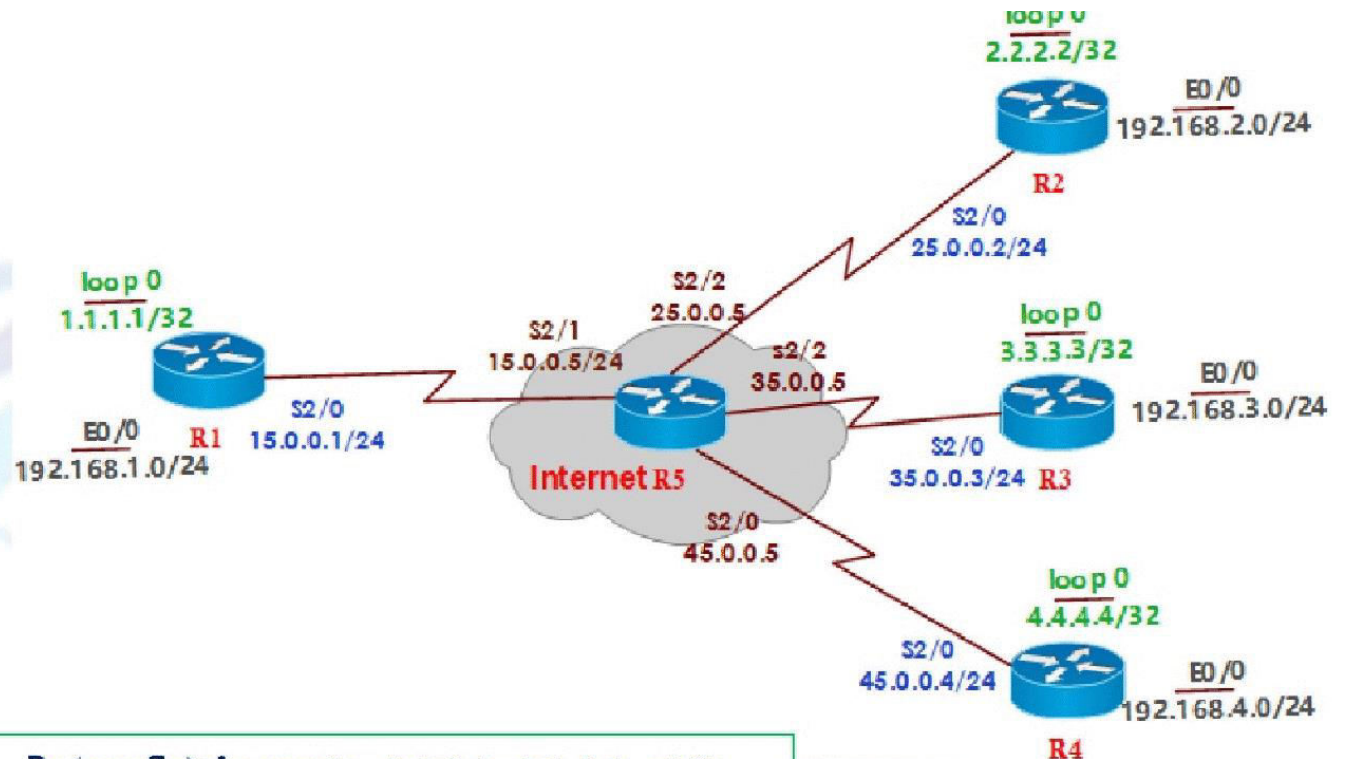


DMVPN - lab basic Setup

```
R1(config)# int s2/0
R1(config-if)# ip address 15.0.0.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit

R1(config)# int loop 0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# exit

R1(config)# int E0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
```



```
Rx(config)# ip route 0.0.0.0 0.0.0.0 s2/0
Rx(config)# exit
```

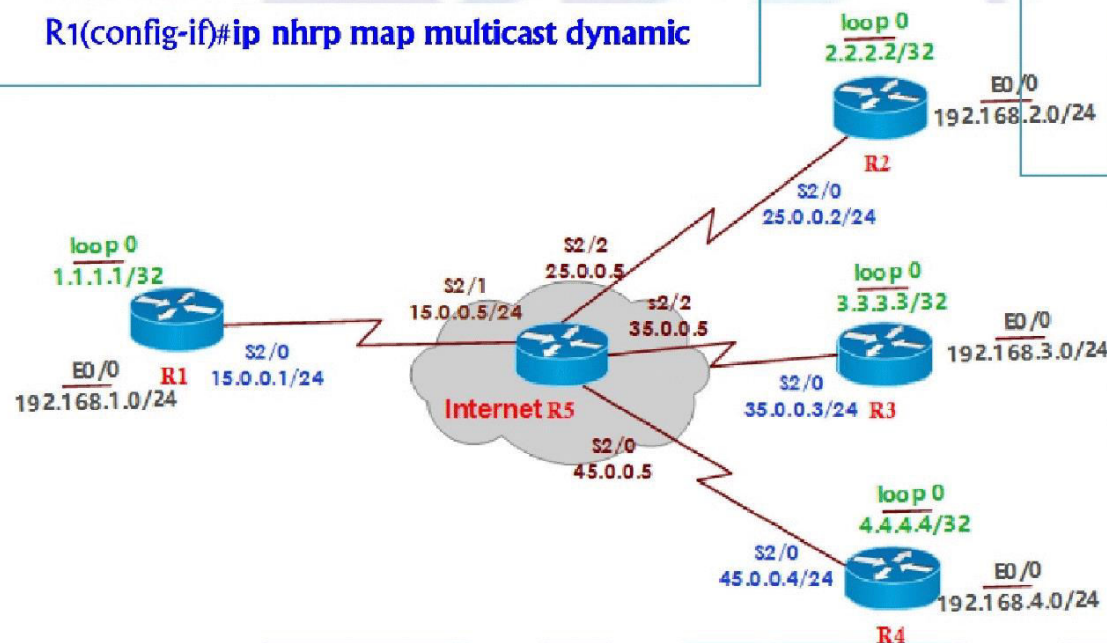
DMVPN- Configuration Example - Phase 2

HUB ROUTER (R1)

```
R1(config)# int tunnel 0
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# tunnel source s2/0
R1(config-if)# tunnel mode gre multipoint
R1(config-if)# ip nhrp network-id 1
R1(config-if)# ip nhrp map multicast dynamic
```

On SPOKES (R2/R3/R4)

```
Rx(config)# interface tunnel 0
Rx(config-if)# ip address 10.0.0.X 255.255.255.0
Rx(config-if)# tunnel source s2/0
Rx(config-if)# tunnel mode gre multipoint
Rx(config-if)# ip nhrp network-id 2
Rx(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
Rx(config-if)# ip nhrp nhs 10.0.0.1
Rx(config-if)# ip nhrp map multicast 15.0.0.1
```



On all Routers

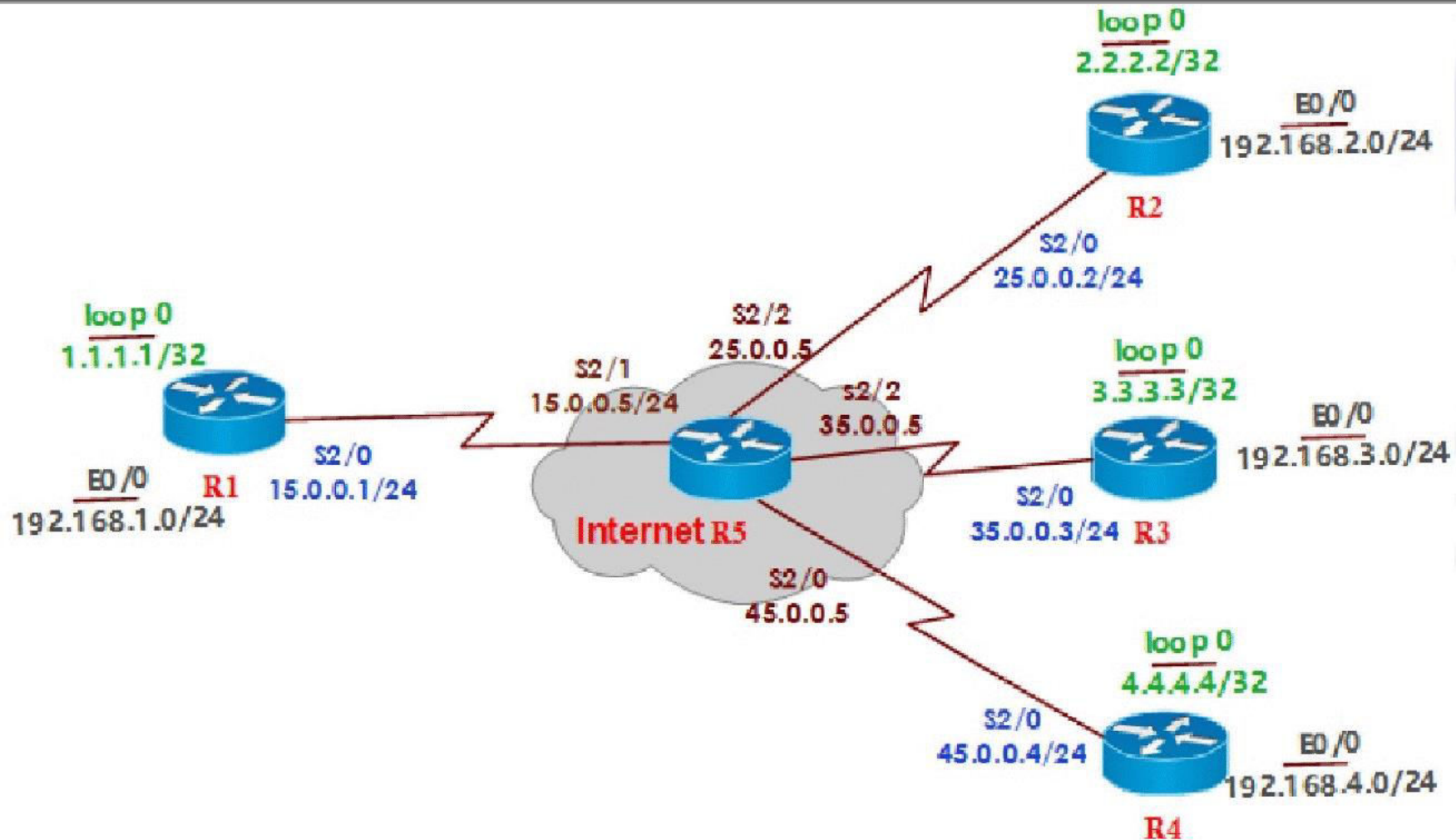
```
(config)# Router eigrp 100
(config-router)# No auto-summary
(config-router)# Network 192.168.X.0
(config-router)# Network 10.0.0.0
```

DMVPN- Configuration Example - Phase 2

Ri(config-if)#ip nhrp network-id 1

- Used to define the NHRP domain for an NHRP interface
- Differentiate between multiple NHRP domains or networks, when two or more NHRP domains
- NHRP network ID is used to help keep two NHRP networks (clouds) separate from each other when both are configured on the same router.
- significant only to the local router and it is not transmitted in NHRP packets to other NHRP nodes
- NHRP network ID configured on a router need not match the same NHRP network ID on another router where both of these routers are in the same NHRP domain.

DMVPN- Configuration Example - Phase 2

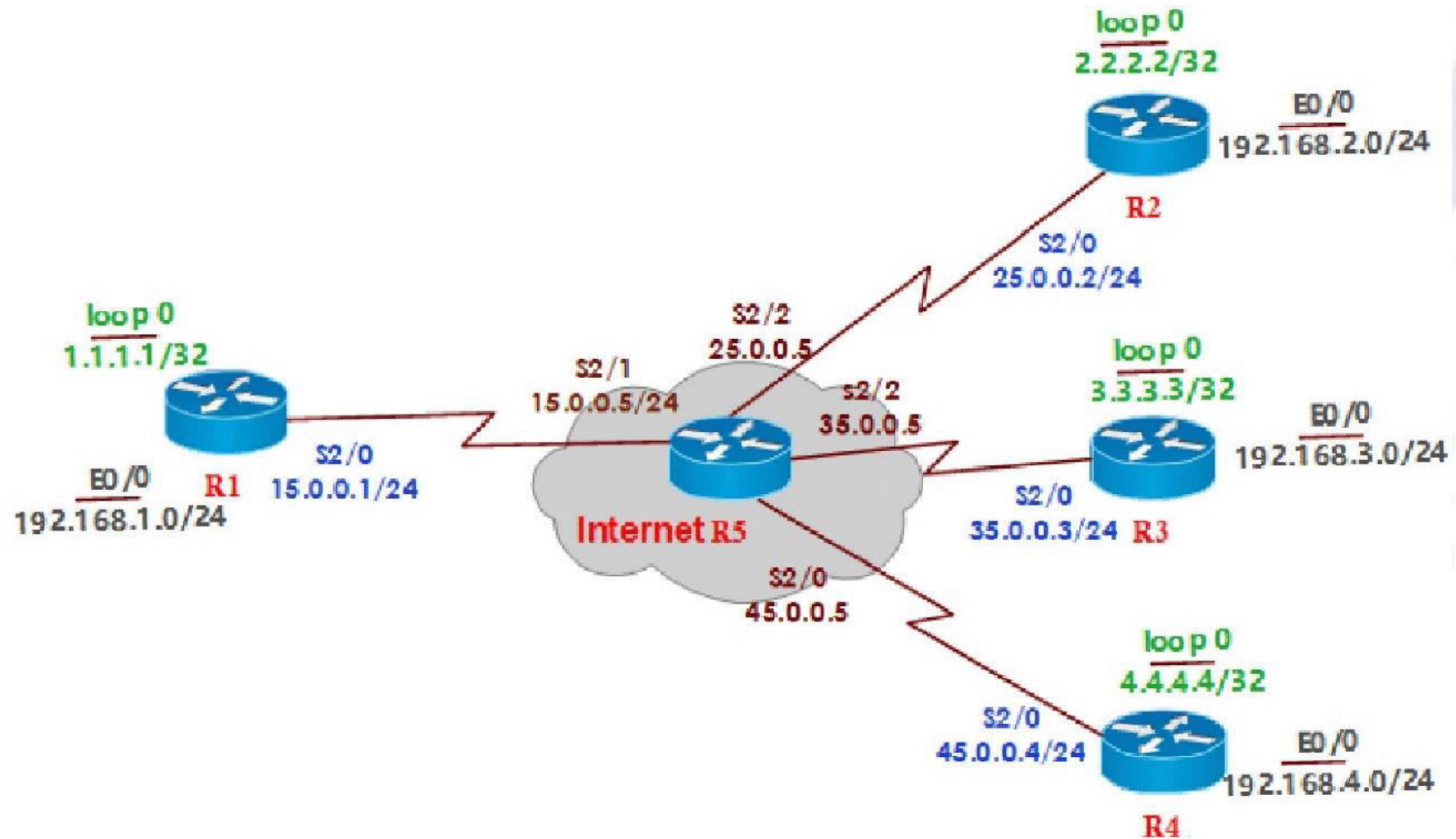


DMVPN- Configuration Example - Phase 2

RI(config-if)#ip nhrp map multicast dynamic

- The tunnel itself does not support multicast
- For this purpose, the **ip nhrp map multicast dynamic command on the hub is used to dynamically create mappings in the**
- NHRP multicast table for each spoke that registers with it.
 - **RI(config-if)#ip nhrp map multicast dynamic**
- You are telling the hub to create a multicast mapping for each spoke that registers with it.
- Usually required by routing protocols such as OSPF and EIGRP

DMVPN- Configuration Example - Phase 2



DMVPN- Configuration Example - Phase 2

- **RI(config-if) # ip nhrp map multicast 15.0.0,1**
- Ensures multicast traffic is sent only from spokes to the hub and not from spoke to spoke.
- All multicast traffic should be received by the hub, processed and then updates are sent out to the spokes.

DMVPN - Verification

R1#show ip nhrp

10.0.0.2/32 via 10.0.0.2, Tunnel0 created 00:14:28, expire 01:48:46

Type: dynamic, Flags: unique registered

NBMA address: 25.0.0.2

10.0.0.3/32 via 10.0.0.3, Tunnel0 created 00:11:42, expire 01:48:17

Type: dynamic, Flags: unique registered

NBMA address: 25.0.0.2

10.0.0.4/32 via 10.0.0.4, Tunnel0 created 00:10:47, expire 01:49:12

Type: dynamic, Flags: unique registered

NBMA address: 45.0.0.4

R4#show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete

N - NATed, L - Local, X - No Socket

Ent --> Number of NHRP entries with same NBMA peer

Tunnel0, Type:Spoke, NHRP Peers:3,

Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb

# Ent	Peer	NBMA Addr	Peer Tunnel	Add	State	UpDn	Tm Attrb
1	15.0.0.1	10.0.0.1	UP	00:02:46	S		
1	25.0.0.2	10.0.0.2	UP	never	D		
1	35.0.0.3	10.0.0.3	UP	never	D		

DMVPN - Verification

Rl#ping 10.0.0.2

- Type escape sequence to abort.
- Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
- !!!!!
- Success rate is 100 percent (5/5), round-trip min/avg/max = 44/114/212 ms

Ri#ping10.0.0.3

- Type escape sequence to abort.
- Sending 5,100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
- !!!!!
- Success rate is 100 percent (5/5), round-trip min/avg/max = 8/136/316 ms

DMVPN - Verification

RI#ping10.0.0.4

- Type escape sequence to abort
- Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
- !!!!
- Success rate is 100 percent (5/5), round-trip min/avg/max = 4/140/268 ms

Routing over DMVPN - Verification

R1#show ip nhrp

10.0.0.2/32 via 10.0.0.2, Tunnel0 created 00:14:28, expire 01:48:46

Type: dynamic, Flags: unique registered

NBMA address: 25.0.0.2

10.0.0.3/32 via 10.0.0.3, Tunnel0 created 00:11:42, expire 01:48:17

Type: dynamic, Flags: unique registered

NBMA address: 25.0.0.2

10.0.0.4/32 via 10.0.0.4, Tunnel0 created 00:10:47, expire 01:49:12

Type: dynamic, Flags: unique registered

NBMA address: 45.0.0.4

R4#show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete

N - NATed, L - Local, X - No Socket

Ent --> Number of NHRP entries with same NBMA peer

Tunnel0, Type:Spoke, NHRP Peers:3,

Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb

```
-----  
1 15.0.0.1 10.0.0.1 UP 00:02:46 S  
1 25.0.0.2 10.0.0.2 UP never D  
1 35.0.0.3 10.0.0.3 UP never D
```

Routing over DMVPN - Verification

Rl#ping 10.0.0.2

- Type escape sequence to abort.
- Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
- !!!
- Success rate is 100 percent (5/5), round-trip min/avg/max = 44/114/212 ms

Ri#ping10.0.0.3

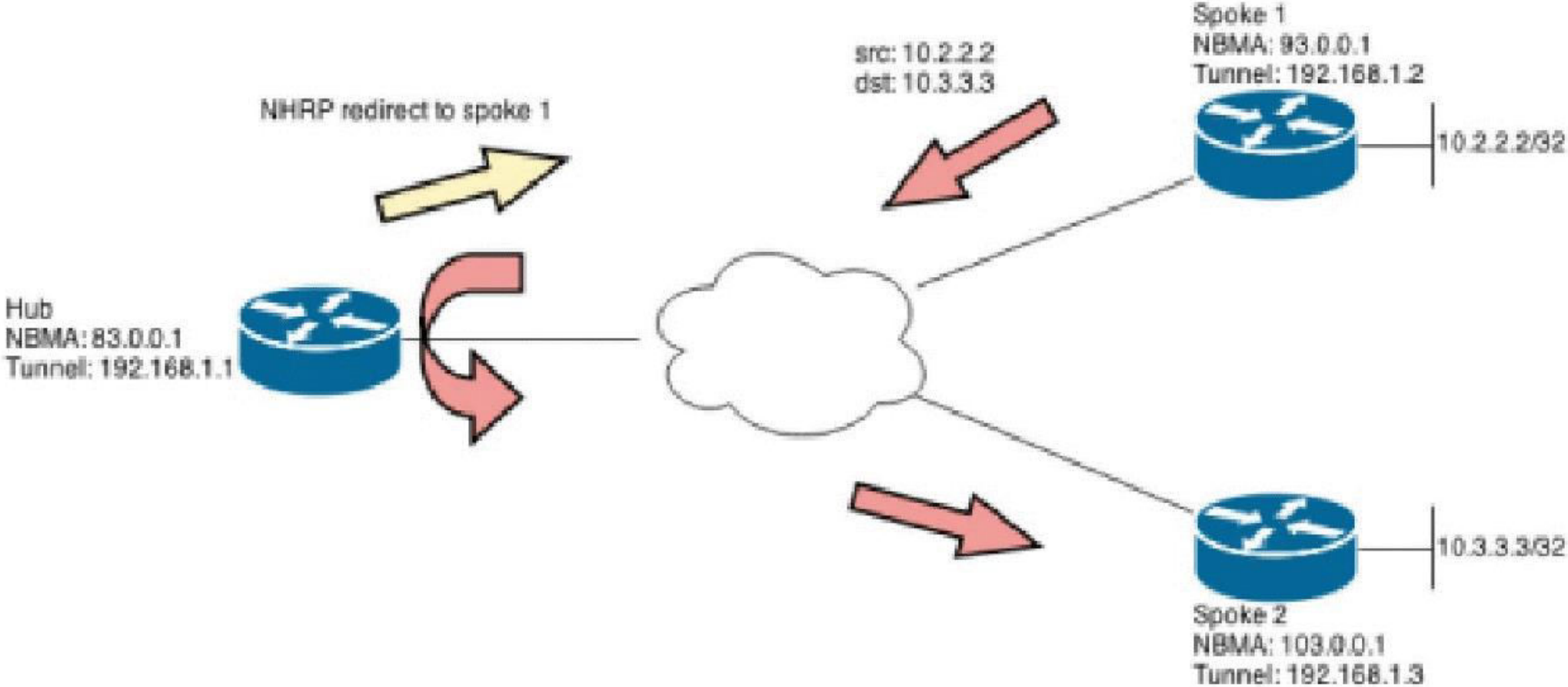
- Type escape sequence to abort.
- Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
- !!!!
- Success rate is TOO percent (5/5), round-trip min/avg/max = 8/136/316 ms

Routing over DMVPN - Verification

Ri#ping10.0.0.4

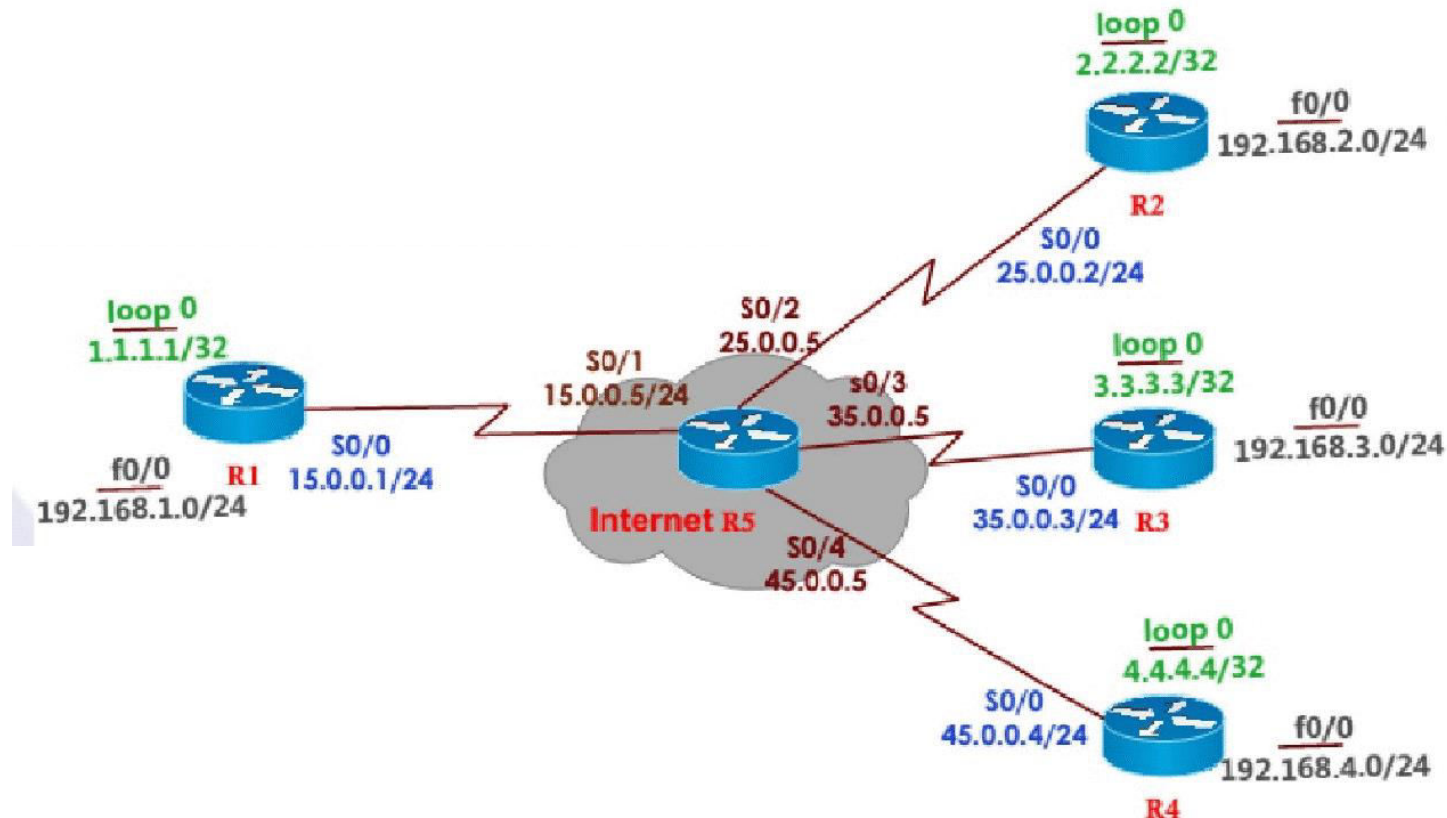
- Type escape sequence to abort.
- Sending 5, loo-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
- !!!!
- Success rate is 100 percent (5/5), round-trip min/avg/max = 4/140/268 ms

DMVPN Phase 3 - EIGRP Routing



DMVPN- Phase 3 - Configuration

- Configuration on tunnel interface is same we did in the Phase-2



DMVPN- Phase 3 - Configuration

HUB ROUTER (R1)

```
R1(config)# int tu 1234
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# tunnel source s0/0
R1(config-if)# tunnel mode gre multipoint
R1(config-if)# ip nhrp network-id 1
R1(config-if)# ip nhrp map multicast dynamic
```

On SPOKES (R2/R3/R4)

```
Rx(config)# interface tunnel 1234
Rx(config-if)# ip address 10.0.0.X 255.255.255.0
Rx(config-if)# tunnel source s0/0
Rx(config-if)# tunnel mode gre multipoint
Rx(config-if)# ip nhrp network-id 2
Rx(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
Rx(config-if)# ip nhrp nhs 10.0.0.1
Rx(config-if)# ip nhrp map multicast 15.0.0.1
```

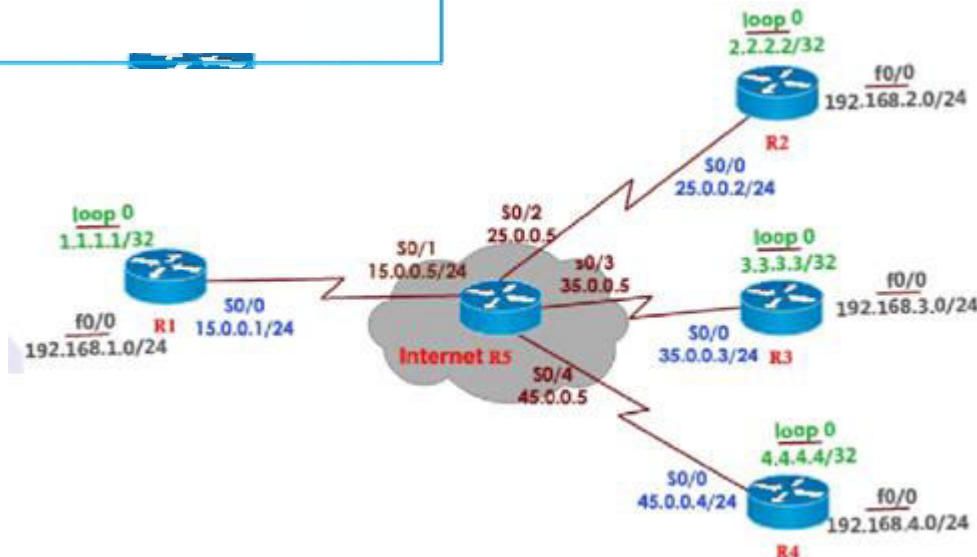
DMVPN Phase 3 - EIGRP ROUTING

ON HUB – R1

```
R1(config)# int tunnel 0
R1(config-if)#no ip split-horizon eigrp 100
R1(config-if)# ip next-hop-self eigrp 100
R1(config-if)# ip nhrp redirect
R1(config-if)# exit
```

on SPOKES (R2/R3/R4)

```
R2(config)# int tunnel 0
R2(config-if)# ip nhrp shortcut
R2(config-if)# exit
```



DMVPN Phase 3 - EIGRP ROUTING

- ▶ NHRP Redirect is a special NHRP message sent by the Hub to the spoke to tell the spoke that there is a better path to the remote spoke than through the Hub.
- ▶ All it does is enforces the spoke to trigger an NHRP resolution request to IP destination. The “ip nhrp redirect” command should be configured on the Hub only!
- ▶ Note that we do not need “no ip next-hop-self eigrp” command in the DMVPN Phase 3.

- ▶ The only difference on the spoke is that the spoke has NHRP Shortcut configured.
- ▶ This will work together with NHRP Redirect on the Hub to send a new Resolution Request NHRP message and overwrite CEF entry to use direct spoke to spoke tunnel instead of the Hub.
- ▶ This command should be configured on spokes only

DMVPN Phase 3 - EIGRP ROUTING

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

Type escape sequence to abort.

Tracing the route to 192.168.2.2

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.0.0.1 19 msec 20 msec 18 msec
```

```
 2 10.0.0.2 39 msec 36 msec 34 msec
```

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

Type escape sequence to abort.

Tracing the route to 192.168.2.2

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.0.0.2 19 msec 23 msec 21 msec
```

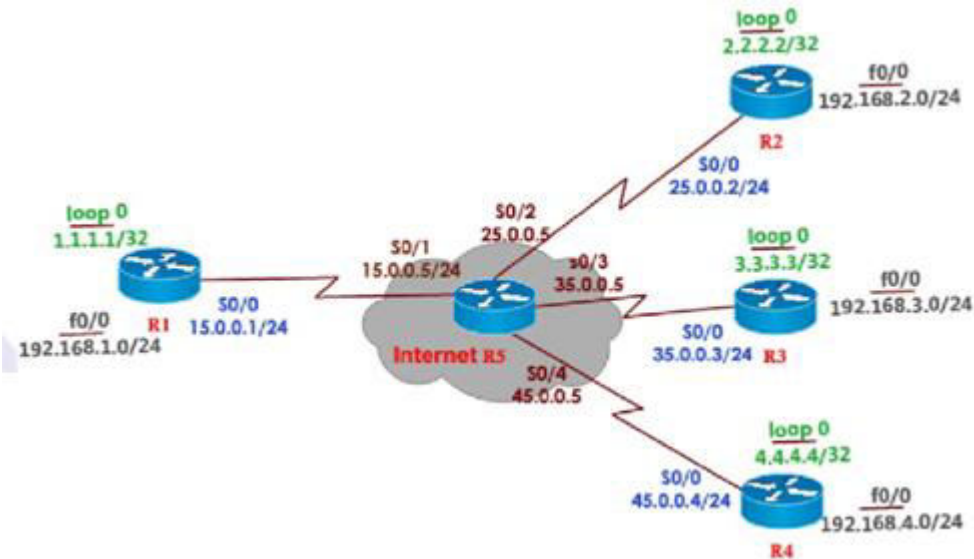
```
R4#show ip route eigrp
```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

```
D 192.168.1.0/24 [90/26905600] via 10.0.0.1, 00:00:18, Tunnel0
```

```
D 192.168.2.0/24 [90/28185600] via 10.0.0.1, 00:00:18, Tunnel0
```

```
D 192.168.3.0/24 [90/28185600] via 10.0.0.1, 00:00:18, Tunnel0
```



DMVPN Phase 3 - EIGRP ROUTING

- ▶ Next-hop to reach all other spokes routes Next hop is HUB ROUTER.
- ▶ This is because of “ip nhrp redirect” Command on Hub & “IP Nhrp shortcut” on spokes.
- ▶ They override the entries in the routing table

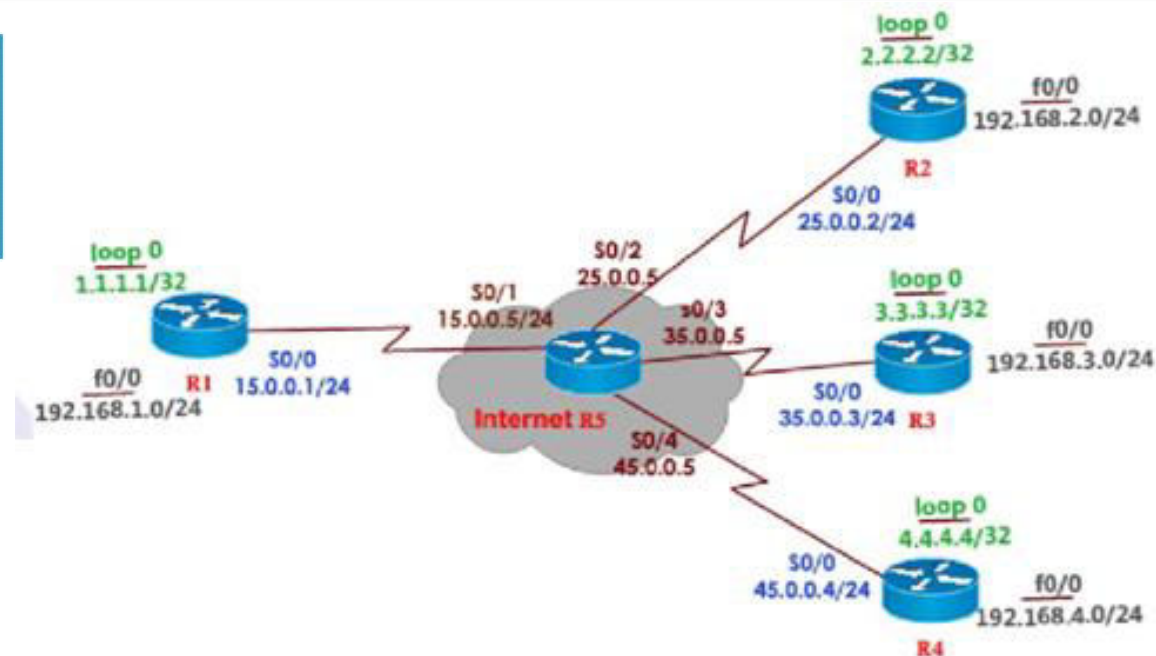
DMVPN Phase 3 - OSPF Routing

ON Hub & Spokes)

```
Rx(config)# int tunnel 0  
Rx(config-if)# ip ospf network point-to-multipoint
```

ON HUB – R1

```
R1(config)# int tunnel 0  
R1(config-if)# ip nhrp redirect  
R1(config-if)# exit
```



on SPOKES (R2/R3/R4)

```
R2(config)# int tunnel 0  
R2(config-if)# ip nhrp shortcut  
R2(config-if)# exit
```

DMVPN Phase 3 - OSPF Routing

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

Type escape sequence to abort.

Tracing the route to 192.168.2.2

VRF info: (vrf in name/id, vrf out name/id)

```
1 10.0.0.1 19 msec 20 msec 18 msec
```

```
2 10.0.0.2 39 msec 36 msec 34 msec
```

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

Type escape sequence to abort.

Tracing the route to 192.168.2.2

VRF info: (vrf in name/id, vrf out name/id)

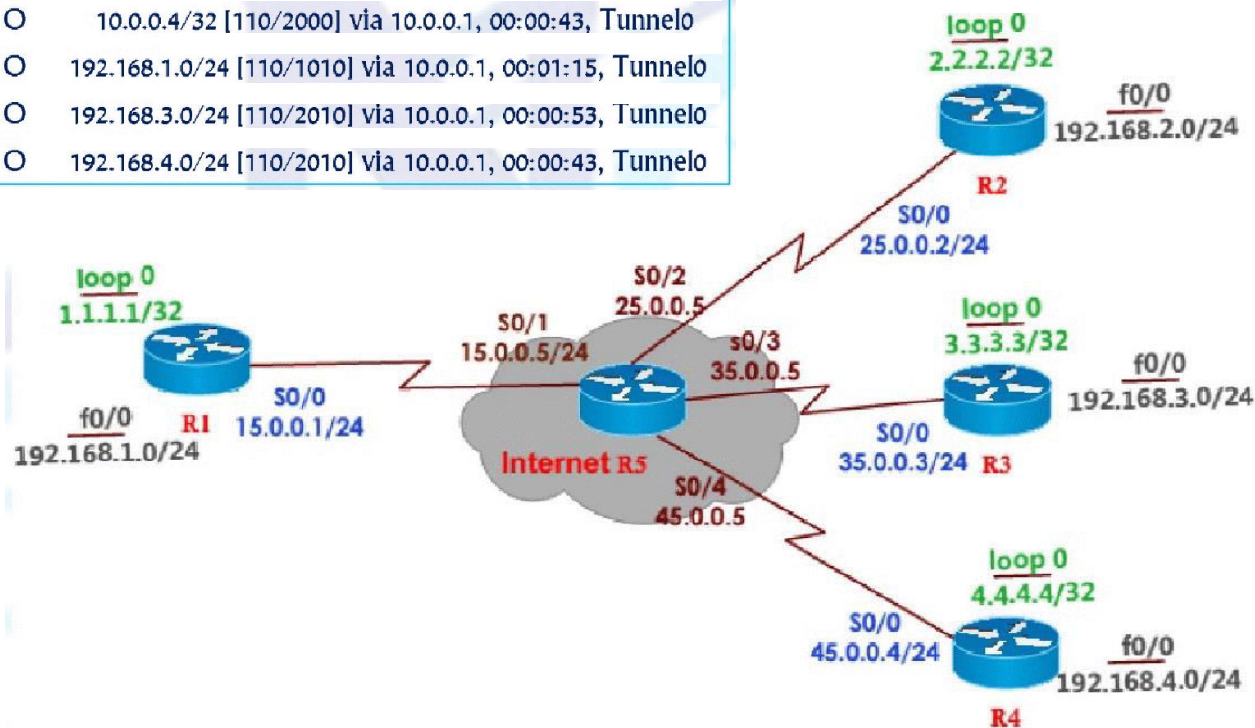
```
1 10.0.0.2 19 msec 23 msec 21 msec
```

```
R2#sh ip route ospf
```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

- 10.0.0.1/32 [110/1000] via 10.0.0.1, 00:01:15, Tunnel0
- 10.0.0.3/32 [110/2000] via 10.0.0.1, 00:00:53, Tunnel0
- 10.0.0.4/32 [110/2000] via 10.0.0.1, 00:00:43, Tunnel0
- 192.168.1.0/24 [110/1010] via 10.0.0.1, 00:01:15, Tunnel0
- 192.168.3.0/24 [110/2010] via 10.0.0.1, 00:00:53, Tunnel0
- 192.168.4.0/24 [110/2010] via 10.0.0.1, 00:00:43, Tunnel0



DMVPN Phase 3 - OSPF Routing

- ▶ Next-hop to reach all other spokes routes Next hop is HUB ROUTER.
- ▶ This is because of “ip nhrp redirect” Command on Hub & “IP Nhrp shortcut” on spokes.
- ▶ They override the entries in the routing table



DMPVN with IPSec (IKEv1)

DMVPN and IKEvi IPsec

Previous LAN-to-LAN IPsec used CryptoMaps

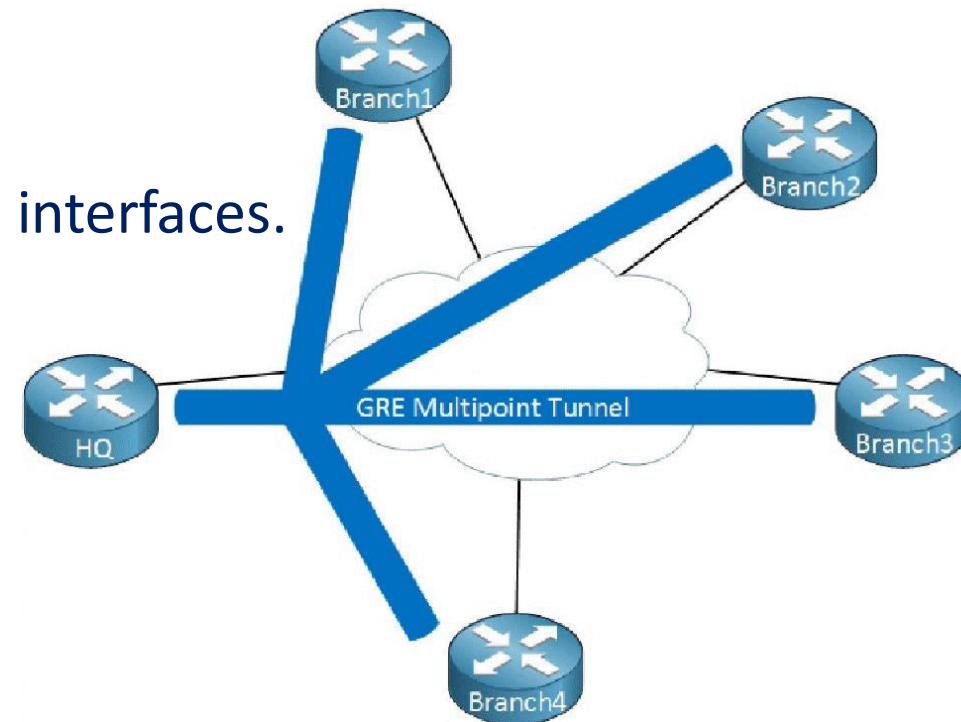
- Requires manual peer and proxy ACL definitions
- Not scalable.

IPsec over DMVPN

- DMVPN uses IPsec Crypto Profile applied on tunnel interfaces.
- Configuration identical to GRE with IPsec Profile.

IPsec Profile protects all traffic inside GREtunnel

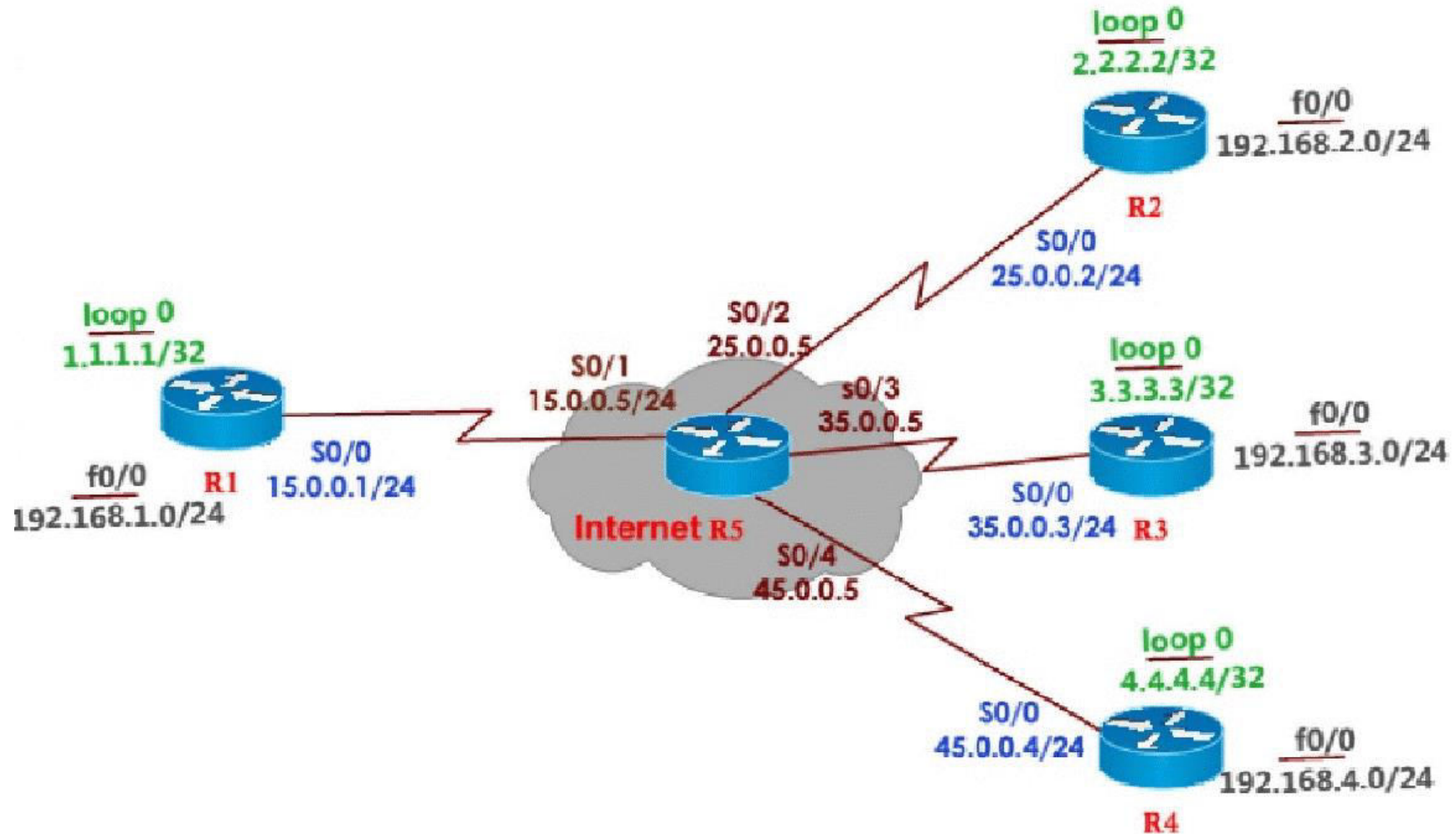
- Both control-plane and data-plane
- NHRP/IGP/BGP traffic is protected.



IPsec over DMVPN- Configuration steps

1. Configure matching IKE (ISAKMP) Policy attributes 2.2.2.2/32
2. Configure Pre-shared key used for Authenticating remote peers
3. Configure IP transform set
4. Create IPsec Profile & attach transform-set.
5. Apply IPsec Profile on tunnel interface.

IPsec over DMVPN- Configuration steps

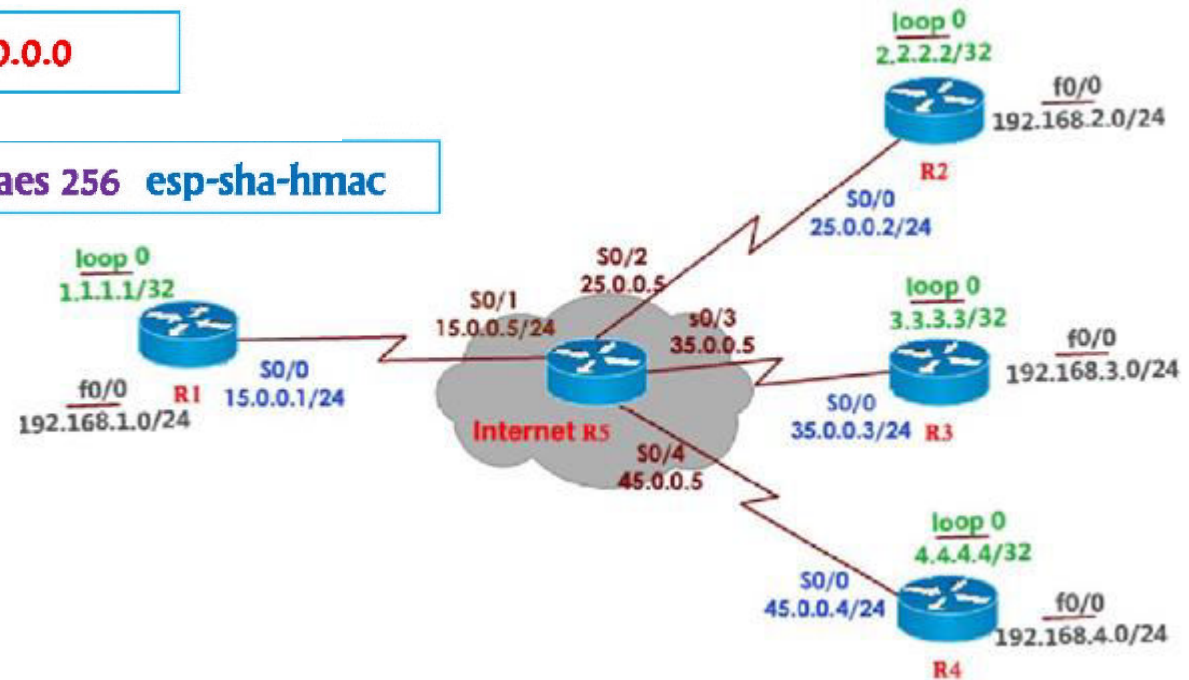


IPsec over DMVPN - Configuration

```
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#hash sha
R1(config-isakmp)#group 5
```

```
R1(config)# crypto isakmp key cisco123 address 0.0.0.0
```

```
R1(config)#crypto ipsec transform-set IP_SET esp-aes 256 esp-sha-hmac
```

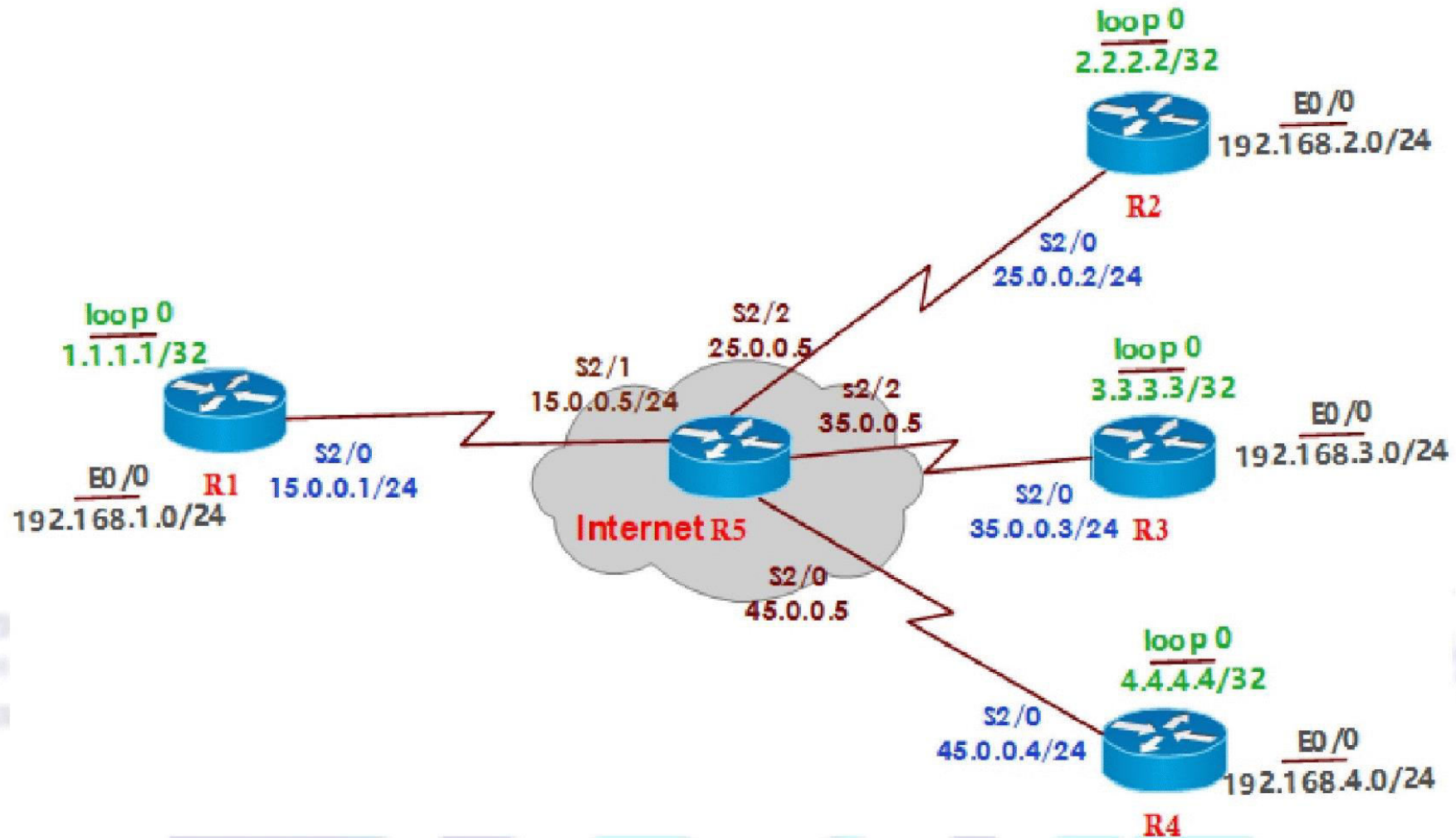


IPsec over DMVPN - Configuration

```
Rx(config)# crypto ipsec profile TN_PR  
Rx(ipsec-profile)# set transform-set IP_SET  
Rx(ipsec-profile)# exit
```

```
Rx(config)# interface tunnel 0  
Rx(config-if)# tunnel protection ipsec profile TN_PR  
Rx(config-if)# exit
```

IPsec over DMVPN - Configuration





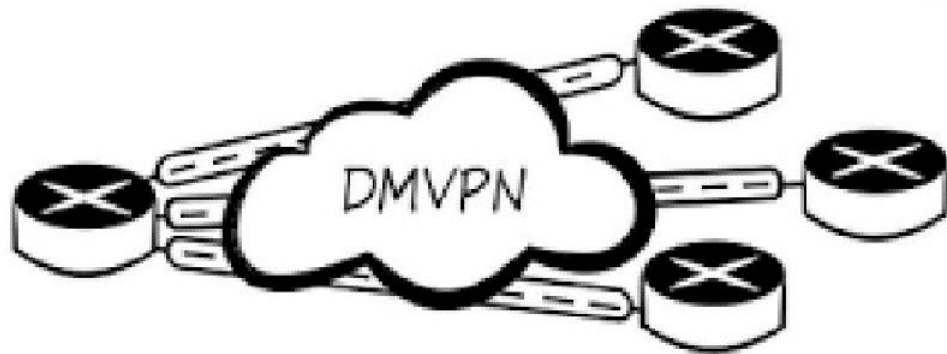
Dynamic Multi-point VPN

Dynamic Multi-point VPN

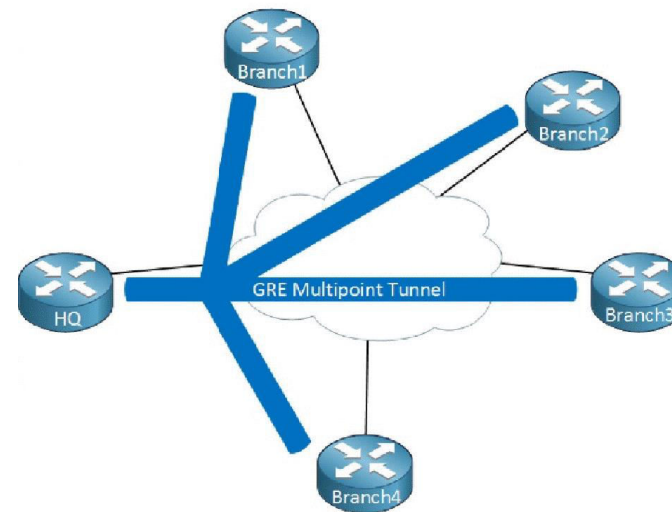
- **DMVPN- Protocols CCNP**
- **mGRE**
- **NHRP**
- **DMVPN Phases – 1 2 3**

DMVPN and IKEvi IPsec

- DMVPN is a "routing technique" that relies on multipoint GRE and NHRP (Without Encryption)
- Mostly we use DMVPN with the Internet as the underlay network, it might be wise to encrypt your tunnels.
- With IPsec Over DMVPN Tunnels, we can encrypt tunnel traffic between sites.



DMVPN + IPSEC



DMVPN and IKEvi IPsec

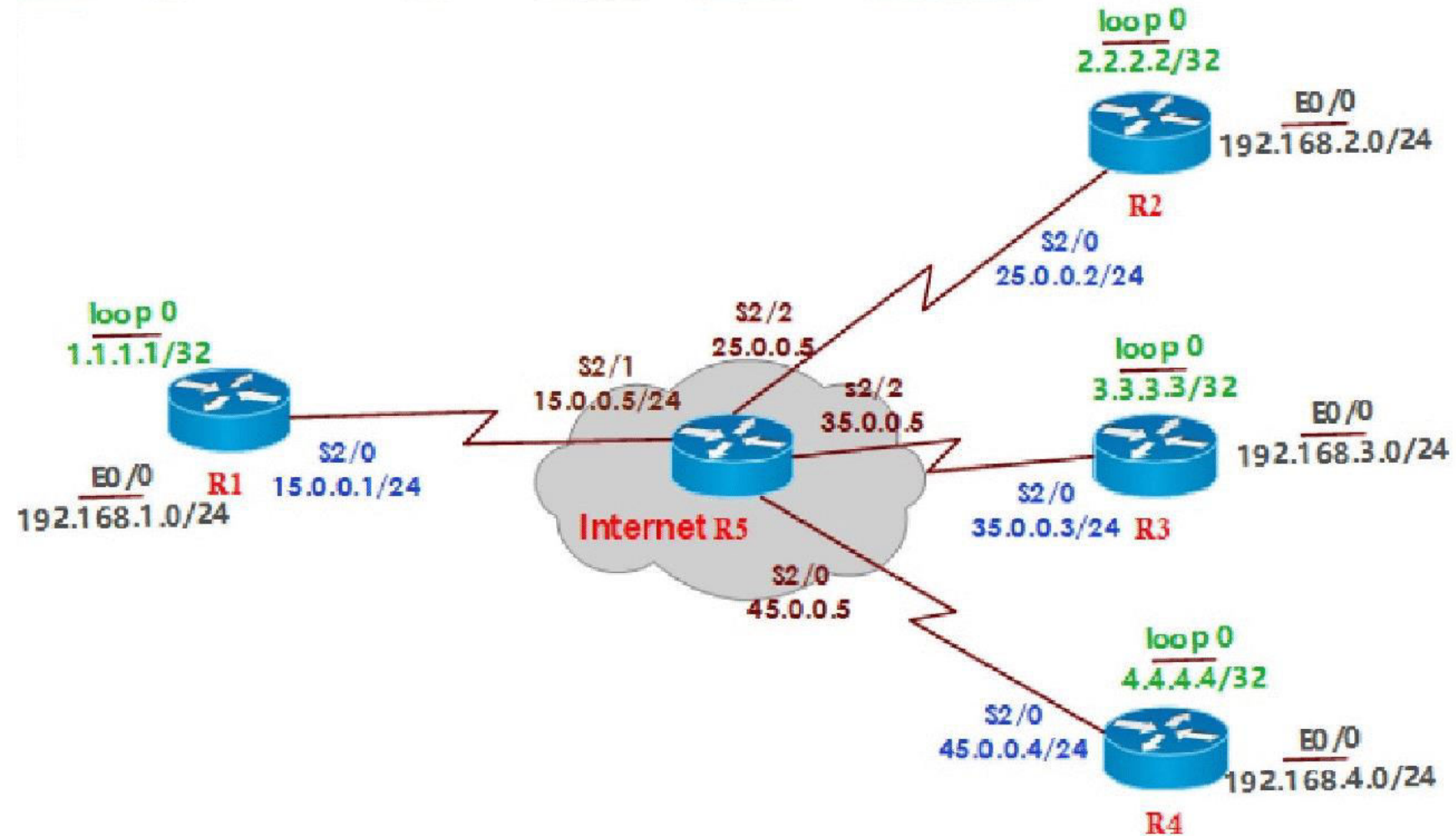
IPsec Profile protects ad traffic inside DMVPN GRE tunnels

- Both control-plane and data-plane
- NHRP/IGP/BGP traffic is protected

IPsec over DMVPN - Configuration steps

1. Configure matching IKE (ISAKMP) Policy attributes
2. Configure Pre-shared key used for Authenticating remote peers
3. Configure IP transform set.
4. Create IPsec Profile & attach transform-set.
5. Apply IPsec Profile on tunnel interface.

IPsec over DMVPN - Configuration steps



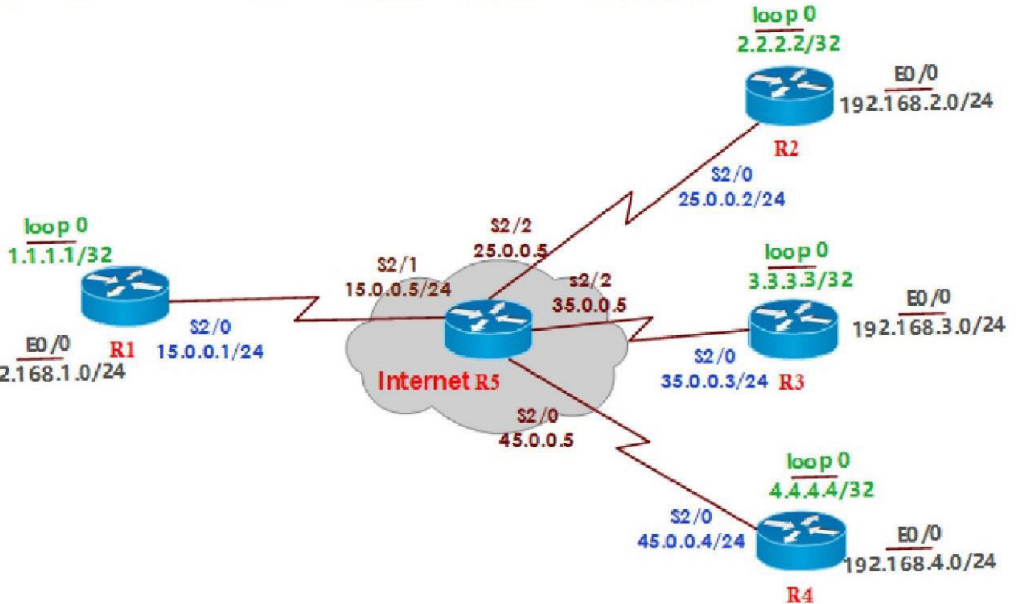
IPsec over DMVPN - Configuration

```
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#hash sha
R1(config-isakmp)#group 5
```

```
R1(config)# crypto isakmp key cisco123 address 0.0.0.0
```

```
R1(config)#crypto ipsec transform-set IP_SET esp-aes 256 esp-sha-hmac
```

```
Rx(config)# crypto ipsec profile TN_PR
Rx(ipsec-profile)# set transform-set IP_SET
Rx(ipsec-profile)# exit
```



```
Rx(config)# interface tunnel 0
Rx(config-if)# tunnel protection ipsec profile TN_PR
Rx(config-if)# exit
```

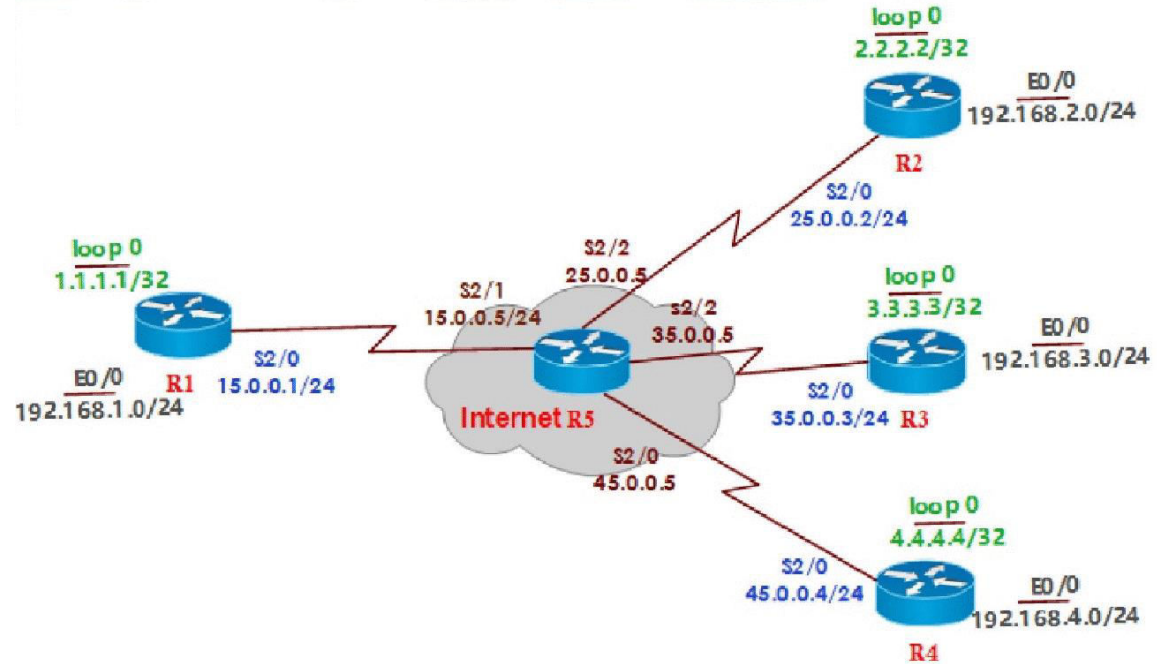
IPsec over DMVPN- Configuration

```
R4#show crypto isakmp sa
```

IPv4 Crypto ISAKMP SA

dst	src	state	conn-id	status
15.0.0.1	45.0.0.4	QM_IDLE	1001	ACTIVE
45.0.0.4	15.0.0.1	QM_IDLE	1003	ACTIVE
25.0.0.2	45.0.0.4	QM_IDLE	1002	ACTIVE
45.0.0.4	25.0.0.2	QM_IDLE	1004	ACTIVE
45.0.0.4	35.0.0.3	QM_IDLE	1005	ACTIVE
35.0.0.3	45.0.0.4	QM_IDLE	1006	ACTIVE

IPv6 Crypto ISAKMP SA



```
R4#show crypto ipsec sa
```

```
R1#show dmvpn
```

```
R1#show crypto isakmp sa
```

KE Phase 1 - Configuration

- ▶ Configure matching IKE (ISAKMP) Policy attributes
- ▶ Configure Pre-shared key used for Authenticating remote peers
- ▶ Configure IP transform set.

```
RI(config)#crypto isakmp policy 10
```

```
RI(config-isakmp)#authentication pre-share
```

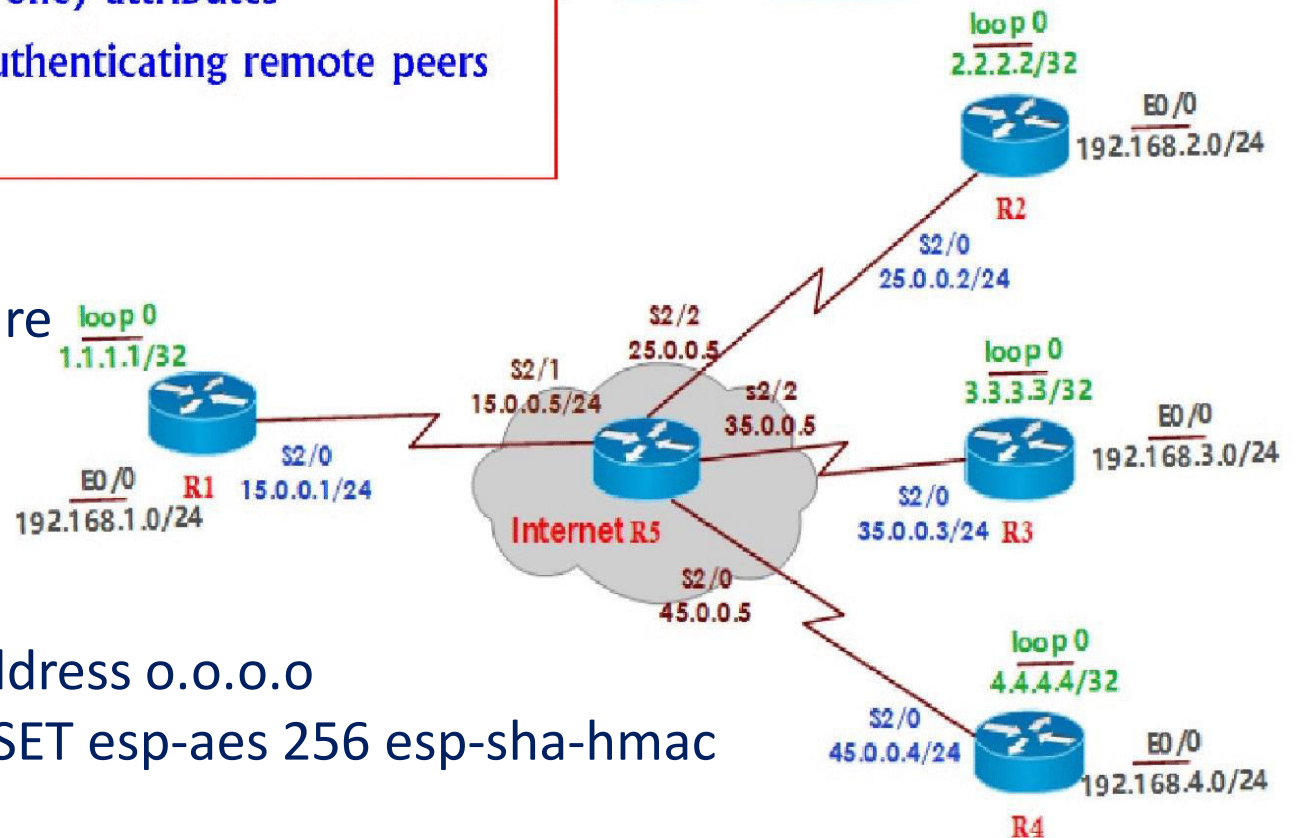
```
RI(config-isakmp)#encryption aes 256
```

```
RI(config-isakmp)#hash sha
```

```
RI(config-isakmp)#group 5
```

```
Ri(config)# crypto isakmp key ciscoi 23 address o.o.o.o
```

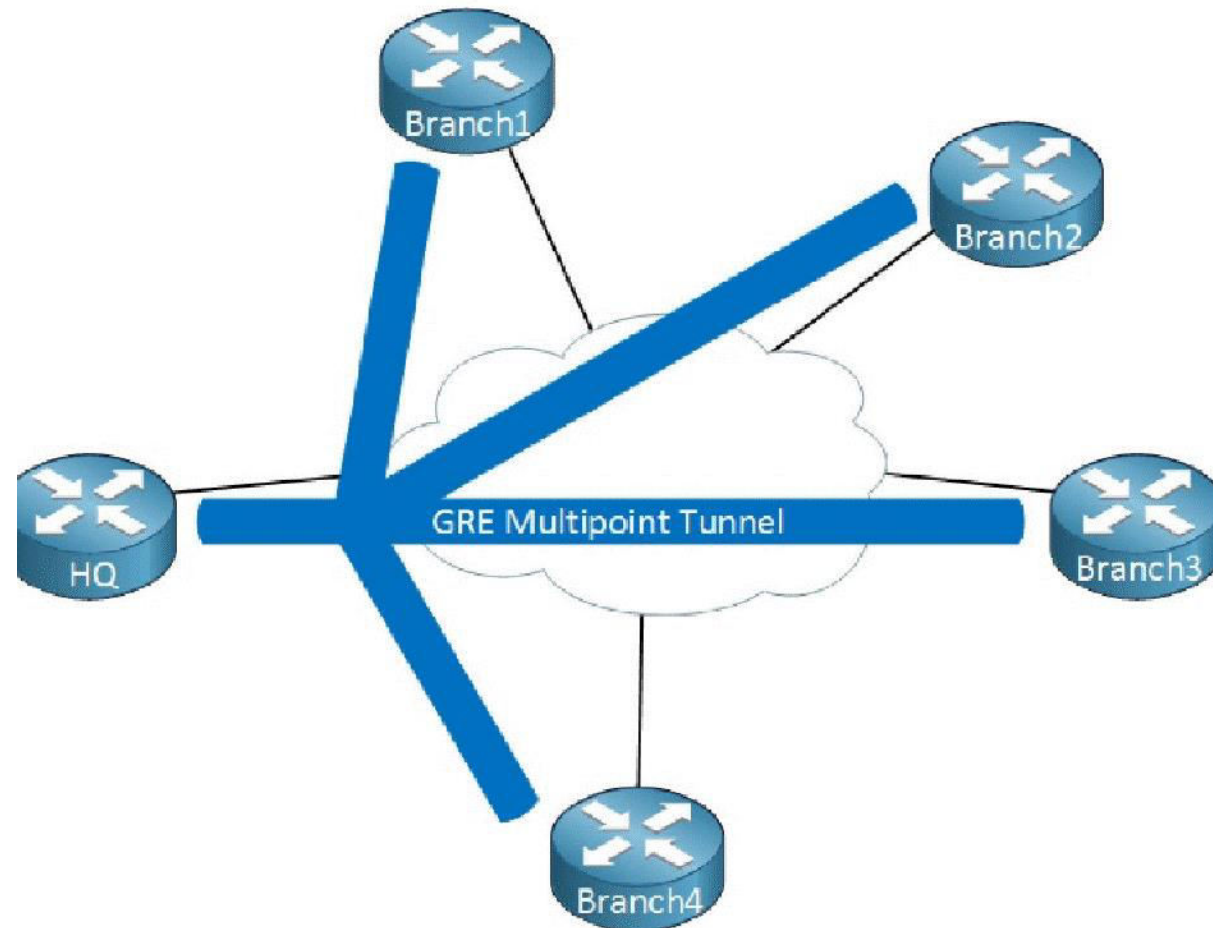
```
Ri(config)#crypto ipsec transform-set IP_SET esp-aes 256 esp-sha-hmac
```



DMVPN and Crypto IPsec Profiles

Order of operation

- NBMA routing has to work first
- IPsec is after
- GRE/NHRP is after
- Overlay IGP/BGP is after



DMVPN and IKEvi IPsec

Previous LAN-to-LAN IPsec used Crypto Maps

- Requires manual peer and proxy ACL definitions
- Not scalable.

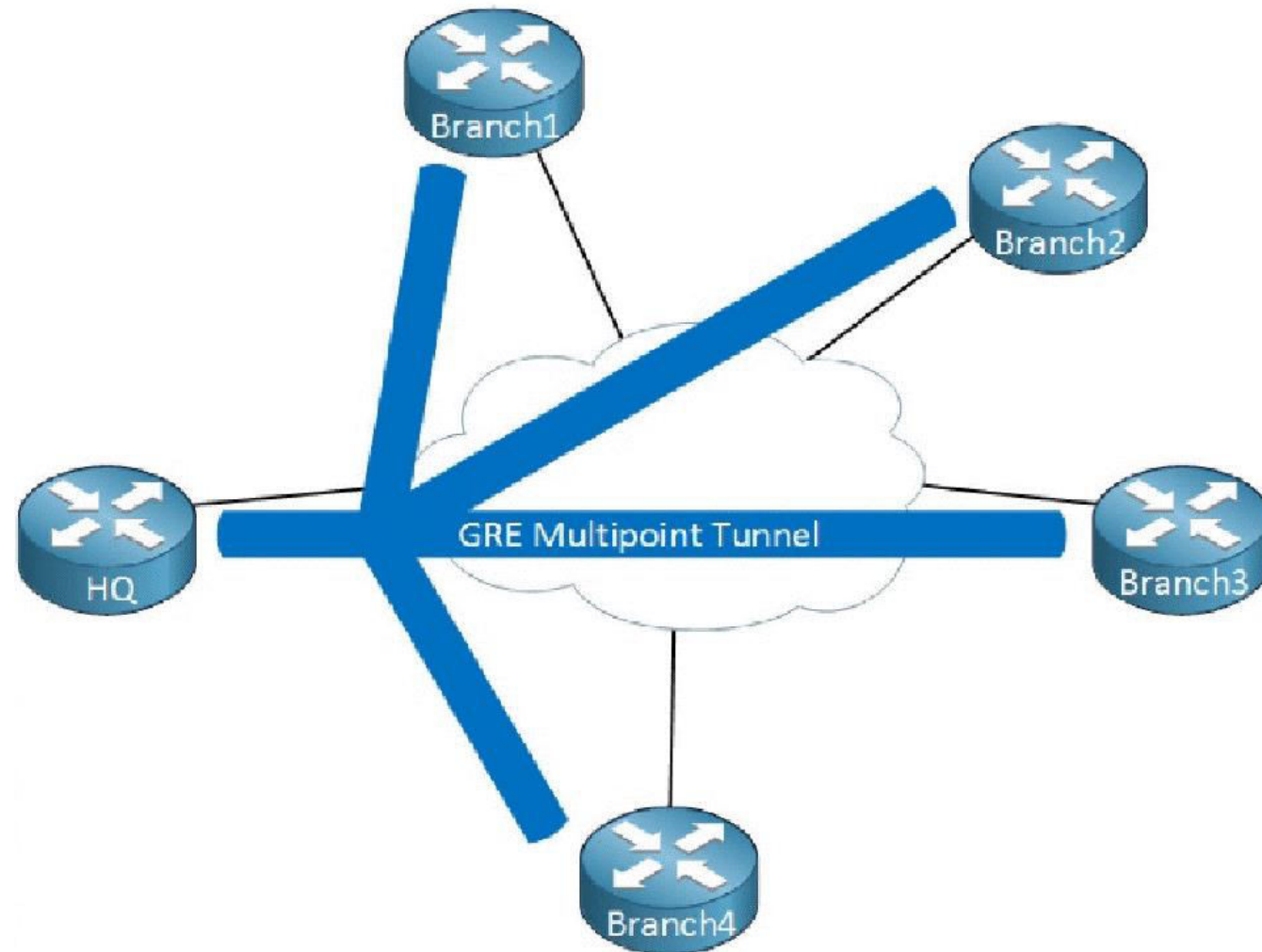
IPsec over DMVPN

- DMVPN uses IPsec Crypto Profile applied on tunnel interfaces.
- Configuration identical to GRE with IPsec Profile.

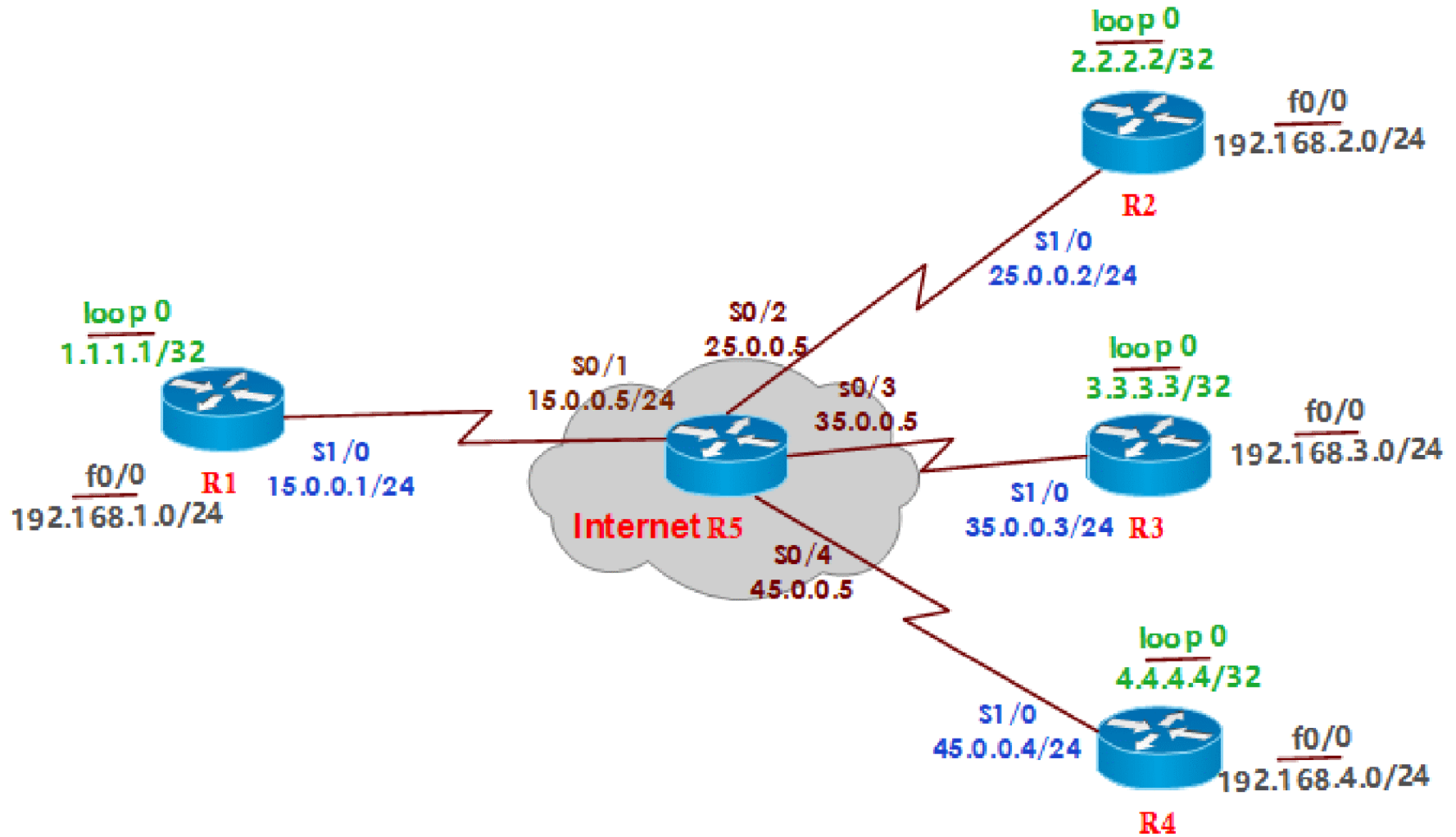
IPsec Profile protects all traffic inside GRE tunnel

- Both control-plane and data-plane
- NHRP/IGP/BGP traffic is protected

DMVPN and IKEvi IPsec



LAB: DMVPN Basic Example



LAB: DMVPN Basic Example

TASK: (Remove all Previous config and make sure that u have only reachability to 25.0.0.2 and 35.0.0.3 and 45.0.0.4 via static and default routing and no eigrp configured and no any tunnel also)

- Configure Hub-and-Spoke GRE tunnels between R1, R2, R3, R4 where R1 is acting as a Hub.
- Traffic originated from every Spoke's F0/0 interface should be transmitted directly to the other spokes.
- Use EIGRP dynamic routing protocol to let other spokes know about protected networks.
- Use IP addressing 10.0.0.x/24 and ensure that all tunnel end points should be able to reach each other.

LAB: DMVPNBasic Example

```
R1#ping 25.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 25.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/19/20 ms
R1#ping 35.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 35.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/20/24 ms
R1#ping 45.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 45.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 17/20/25 ms
R1#
```

LAB: DMVPN Basic Example

```
R1 (config)# int tunnel 0
```

```
R1 (config-if)# ip address 10.0.0.1 255.255.255.0
```

```
R1 (config-if)# tunnel source 1 5.0.0.1
```

```
R1 (config-if)# tunnel mode gre multipoint
```

```
R1 (config-if)# ip
```

LAB: DMVPN Basic Example

Note:

- R1 will be configured as hub.
- `ip nhrp network ID` enables NHRP on tunnel interface.

R1 (config-if)#ip nhrp network-id 1

LAB: DMVPN Basic Example

- Used to define the NHRP domain for an NHRP interface
- Differentiate between multiple NHRP domains or networks, when two or more NHRP domains
- NHRP network ID is used to help keep two NHRP networks (clouds) separate from each other when both are configured on the same router.
- significant only to the local router and it is not transmitted in NHRP packets to other NHRP nodes
- NHRP network ID configured on a router need not match the same NHRP network ID on another router where both of routers are in the same NHRP domain

LAB: DMVPN Basic Example

tunnel mode gre multipoint: Sets the GRE tunnel to behave as a multipoint tunnel.

```
R2(config-if)# interface tunnel 0
```

```
R2(config-if)# ip address 10.0.0.2 255.255.255.0
```

```
R2(config-if)# tunnel source s1 /0
```

```
R2(config-if)# tunnel Destination 15.0.0.1
```

```
R2(config-if)# tunnel mode gre multipoin
```

```
R2(config-if)# ip nhrp network-id 2
```

```
R2(config-if)# ip nhrp nhs 10.0.0.1
```

```
R2(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
```

```
R2(config-if)# exit
```

LAB: DMVPN Basic Example

```
R3(config)# interface tunnel 0
R3(config-if)#ip address 10.0.0.3 255.255.255.0
R3(config-if)#tunnel source s1 /0
R3(config-if)# tunnel Destination 15.0.0.1
R3(config-if)#tunnel mode gre multipoin
R3(config-if)#ip nhrp network-id 3
R3(config-if)# ip nhrp map 10.0.0.1 1 5.0.0.1
R3(config-if)# ip nhrp nhs 10.0.0.1
R3(config-if)# exit
```

LAB: DMVPN Basic Example

```
R4(config)# interface tunnel 0
R4(config-if)# ip address 10.0.0.4 255.255.255.0
R4(config-if)# tunnel source s1 /0
R4(config-if)# tunnel Destination 15.0.0.1
R4(config-if)# tunnel mode gre multipoint
R4(config-if)# ip nhrp network-id 4
R4(config-if)# ip nhrp map 10.0.0.1 1 5.0.0.1
R4(config-if)# ip nhrp nhs 10.0.0.1
R4(config-if)# exit
```

LAB: DMVPN Basic Example - with EIGRP Routing

```
R1(config)# router eigrp 100
R1(config-router)# no auto-summary
R1(config-router)# network 10.0.0.0
R1(config-router)# network 192.168.1.0
R1(config-router)# exit
R2(config)# router eigrp 100
R2(config-router)# no auto-summary
R2(config-router)# network 10.0.0.0
R2(config-router)# network 192.168.2.0
R2(config-router)# exit
```

LAB: DMVPN Basic Example - with EIGRP Routing

```
R3(config)# router eigrp 100
```

```
R3(config-router)# no auto-summary
```

```
R3(config-router)# network 10.0.0.0
```

```
R3(config-router)# network 192.168.3.0
```

```
R3(config-router)# exit
```

```
R4(config)# router eigrp 100
```

```
R4(config-router)# no auto-summary
```

```
R4(config-router)# network 10.0.0.0
```

```
R4(config-router)# network 192.168.4.0
```

```
R4(config-router)# exit
```

LAB: DMVPNBasic Example

```
Loopback0          4.4.4.4          YES NVRAM  up          up
Tunnel0            10.0.0.4         YES manual up          up
Tunnel14          unassigned       YES unset  up          down
Tunnel141         10.0.14.4        YES NVRAM  up          up
R4(config)#
R4(config)#
```

```
R4#sh ip nhrp
10.0.0.1/32 via 10.0.0.1
  Tunnel0 created 00:00:53, never expire
  Type: static, Flags: used
  NBMA address: 15.0.0.1
R4#
R4#
```

LAB: DMVPN Basic Example

```
R1#sh ip nhrp
10.0.0.2/32 via 10.0.0.2
  Tunnel0 created 00:01:53, expire 01:58:06
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 25.0.0.2
10.0.0.3/32 via 10.0.0.3
  Tunnel0 created 00:01:31, expire 01:58:28
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 35.0.0.3
10.0.0.4/32 via 10.0.0.4
  Tunnel0 created 00:01:11, expire 01:58:48
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 45.0.0.4
R1#
```

LAB: DMVPNBasic Example

```
R1#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/21/25 ms
R1#ping 10.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 17/18/21 ms
R1#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 18/18/20 ms
```

LAB: DMVPNBasic Example

```
R4#traceroute 10.0.0.2
Type escape sequence to abort.
Tracing the route to 10.0.0.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.1 26 msec 16 msec 25 msec
  2 10.0.0.2 20 msec 23 msec 19 msec
R4#traceroute 10.0.0.3
Type escape sequence to abort.
Tracing the route to 10.0.0.3
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.1 18 msec 19 msec 23 msec
  2 10.0.0.3 20 msec 15 msec 19 msec
R4#traceroute 10.0.0.2
Type escape sequence to abort.
Tracing the route to 10.0.0.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.2 21 msec 21 msec 15 msec
R4#traceroute 10.0.0.3
Type escape sequence to abort.
Tracing the route to 10.0.0.3
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.3 24 msec 16 msec 21 msec
R4#
```

LAB: DMVPNBasic Example

```
R4#sh ip nhrp detail
10.0.0.1/32 via 10.0.0.1
  Tunnel0 created 00:17:36, never expire
  Type: static, Flags: used
  NBMA address: 15.0.0.1
10.0.0.2/32 via 10.0.0.2
  Tunnel0 created 00:00:43, expire 01:59:16
  Type: dynamic, Flags: router used nhop
  NBMA address: 25.0.0.2
10.0.0.3/32 via 10.0.0.3
  Tunnel0 created 00:00:41, expire 01:59:18
  Type: dynamic, Flags: router used nhop
  NBMA address: 35.0.0.3
R4#
```

LAB: DMVPN Basic Example

```
R4#show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface: Tunnel0, IPv4 NHRP Details
Type:Spoke, NHRP Peers:3,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
-----
  1 15.0.0.1          10.0.0.1    UP 00:03:46    S
  1 25.0.0.2          10.0.0.2    UP 00:01:23    D
  1 35.0.0.3          10.0.0.3    UP 00:01:22    D

R4#
```

LAB: DMVPN Basic Example

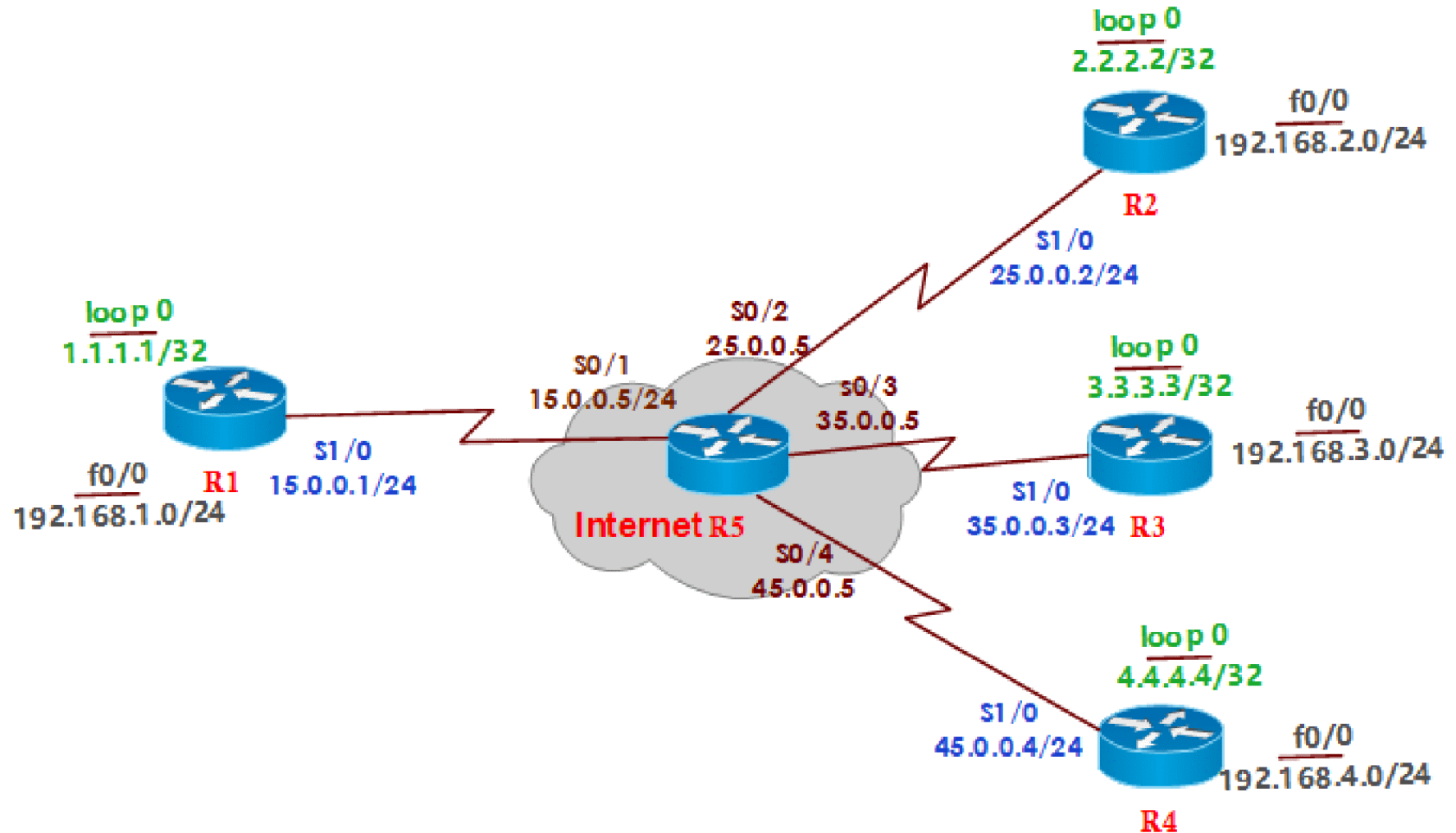
```
R1#show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface: Tunnel0, IPv4 NHRP Details
Type:Hub, NHRP Peers:3,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
-----
   1  25.0.0.2          10.0.0.2   UP 00:04:57   D
   1  35.0.0.3          10.0.0.3   UP 00:04:35   D
   1  45.0.0.4          10.0.0.4   UP 00:04:14   D

R1#
```

LAB: DMVPN Basic Example - with EIGRP Routing



LAB: DMVPN Basic Example - with EIGRP Routing

TASK:

- Continue with same DMVPN tunnel configurations based on the previous lab
- Configure EIGRP 100 on all routers to provide LAN to LAN Connectivity between Hub and Spokes.
- we need a routing protocol over the tunnel.
- Remember, this protocol will be used to carry the info about networks behind the Spokes (or Hub). Be careful when configuring it as there is a chance to get into “ recursive loop” .
- This means we shouldn't use the same dynamic routing protocol instance for prefixes available over the tunnel and to achieve underlying connectivity between Hub and Spokes.

LAB: DMVPN Basic Example - with EIGRP Routing

```
R1(config)# router eigrp 100
R1(config-router)# no auto-summary
R1(config-router)# network 10.0.0.0
R1(config-router)# network 192.168.1.0
R1(config-router)# exit
R2(config)# router eigrp 100
R2(config-router)# no auto-summary
R2(config-router)# network 10.0.0.0
R2(config-router)# network 192.168.2.0
R2(config-router)# exit
```

LAB: DMVPN Basic Example - with EIGRP Routing

```
R3(config)# router eigrp 100
```

```
R3(config-router)# no auto-summary
```

```
R3(config-router)# network 10.0.0.0
```

```
R3(config-router)# network 192.168.3.0
```

```
R3(config-router)# exit
```

```
R4(config)# router eigrp 100
```

```
R4(config-router)# no auto-summary
```

```
R4(config-router)# network 10.0.0.0
```

```
R4(config-router)# network 192.168.4.0
```

```
R4(config-router)# exit
```

LAB: DMVPN Basic Example - with EIGRP Routing

```
R1#show ip ei nei
EIGRP-IPv4 Neighbors for AS(100)
R1#
```

LAB: DMVPN Basic Example - with EIGRP Routing

IP nhrp map multicast X.X.X.X

- This command also enables routing protocols to work over the mGRE.
- multicast will not be sent across the MGRE tunnel by default in DMVPN.
- Ip nhrp map multicast X.X.X.X command enables forwarding of multicast traffic across the tunnel to dynamic spokes (required by most routing protocols).
- Normally on the Hub Routers we can configure IP NHRP MAP MULTICAST DYNAMIC to allow NHRP to automatically add routers to the multicast NHRP mappings so a static mapping is not required any more for each of the spokes

LAB: DMVPN Basic Example - with EIGRP Routing

```
R1 (config)# int tunnel 0
```

```
R1 (config-if)# ip nhrp map multicast dynamic
```

```
OR
```

```
R1 (config-if)# ip nhrp map multicast 25.0.0.2
```

```
R1 (config-if)# ip nhrp map multicast 35.0.0.3
```

```
R1 (config-if)# ip nhrp map multicast 45.0.0.4
```

```
R1 (config-if)#exit
```

```
R2 (config)# int tunnel 0
```

```
R2 (config-if)# ip nhrp map multicast 15.0.0.1
```

```
R2 (config-if)# ip nhrp map multicast 35.0.0.3
```

```
R2 (config-if)# ip nhrp map multicast 45.0.0.4
```

LAB: DMVPN Basic Example - with EIGRP Routing

```
R3(config)# int tunnel 0
```

```
R3(config-if)# ip nhrp map multicast 25.0.0.2
```

```
R3(config-if)# ip nhrp map multicast 35.0.0.3
```

```
R3(config-if)# ip nhrp map multicast 45.0.0.4
```

```
R3(config-if)#exit
```

```
R4(config)# int tunnel 0
```

```
R4(config-if)# ip nhrp map multicast 25.0.0.2
```

```
R4(config-if)# ip nhrp map multicast 35.0.0.3
```

```
R4(config-if)# ip nhrp map multicast 45.0.0.4
```

```
R4(config-if)#exit
```

LAB: DMVPN Basic Example - with EIGRP Routing

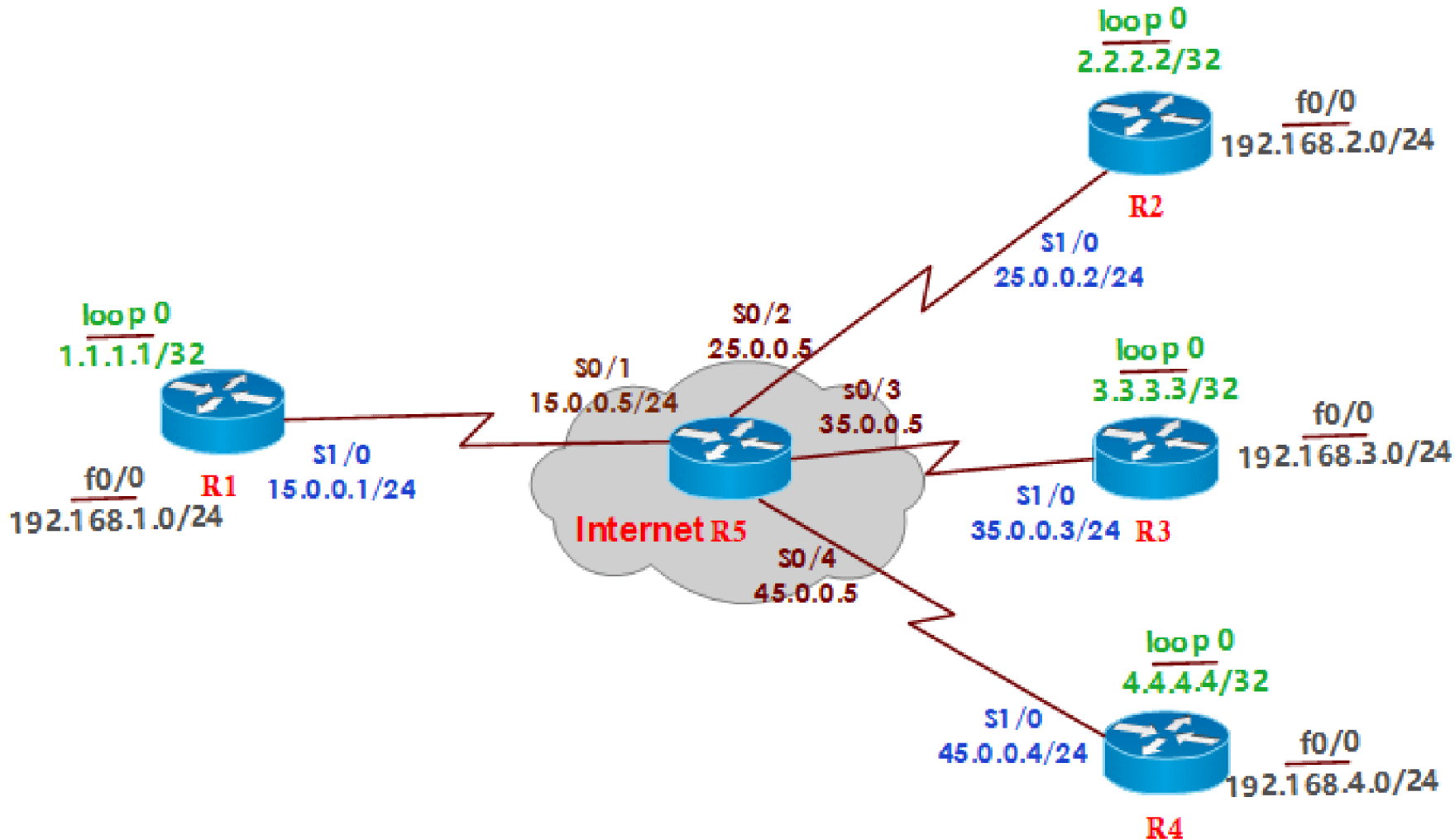
```
R3# sh ip ei nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address          Interface          Hold Uptime      SRTT   RTO   Q   Seq
                               (sec)           (ms)          Cnt  Num
0   10.0.0.1          Tu0               14 00:00:23     1   5000  1   0
2   10.0.0.4          Tu0               14 00:07:10    39   1470  0   13
1   10.0.0.2          Tu0               12 00:08:11    22   1470  0   16
R3#
```

```
R2# sh ip ei nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address          Interface          Hold Uptime      SRTT   RTO   Q   Seq
                               (sec)           (ms)          Cnt  Num
2   10.0.0.4          Tu0               10 00:06:31    802   4812  0   13
1   10.0.0.3          Tu0               14 00:07:01    768   4608  0   13
0   10.0.0.1          Tu0               11 00:07:59     24   1470  0   18
R2#
```

LAB: DMVPN Basic Example - with EIGRP Routing

```
R4#sh ip ei nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address          Interface          Hold Uptime      SRTT   RTO   Q   Seq
                               (sec)           (ms)          Cnt  Num
0   10.0.0.1          Tu0               12 00:01:02      1   5000  1   0
2   10.0.0.3          Tu0               14 00:08:16     19   1434  0   13
1   10.0.0.2          Tu0               10 00:08:50     24   1434  0   16
R4#
```

LAB: DMVPNphase1



LAB: DMVPNphase1

TASK:

- Configure Hub-and-Spoke GRE tunnels between R1, R2 R3, R4 where R1 is acting as a Hub.
- Traffic originated from every Spoke's F0/0 interface should be transmitted directly to the other spokes.
- Use EIGRP dynamic routing protocol to let other spokes know about protected networks.
- Use Ip addressing 10.0.0.x/24 and ensure that all tunnel end points should be able to reach each other

LAB: DMVPNphase1

```
R1#ping 25.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 25.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 18/18/19 ms
R1#ping 35.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 35.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/19/23 ms
R1#ping 45.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 45.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 17/19/20 ms
R1#
```

LAB: DMVPNphase1

```
R1 (config)#int tunnel 0
```

```
R1 (config-if)#ip address 10.0.0.1 255.255.255.0
```

```
R1 (config-if)#tunnel source s1 /0
```

```
R1 (config-if)#tunnel mode gre multipoint
```

```
R1 (config-if)#ip nhrp network-id 1
```

LAB: DMVPNphase1

```
R2(config)# int tunnel 0
R2(config-if)# ip add 10.0.0.2 255.255.255.0
R2(config-if)# tunnel source sl /0
R2(config-if)# tunnel Destination 15.0.0.1
R2(config-if)# ip nhrp network-id 2
R2(config-if)# ip nhrp nhs 10.0.0.1
R2(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
```

LAB: DMVPNphase1

```
R3(config)# int tunnel 0
R3(config-if)# ip add 10.0.0.3 255.255.255.0
R3(config-if)# tunnel source s1 /0
R3(config-if)# tunnel Destination 15.0.0.1
R3(config-if)# ip nhrp network-id 3
R3(config-if)# ip nhrp nhs 10.0.0.1
R3(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
R3(config-if)# exit
```

LAB: DMVPNphase1

```
R4(config)# int tunnel 0
R4(config-if)# ip add 10.0.0.4 255.255.255.0
R4(config-if)# tunnel source s1 /0
R4(config-if)# tunnel Destination 15.0.0.1
R4(config-if)# ip nhrp network-id 4
R4(config-if)# ip nhrp nhs 10.0.0.1
R4(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
R4(config-if)# exit
```

LAB: DMVPNphase1

```
R4#sh ip nhrp
10.0.0.1/32 via 10.0.0.1
  Tunnel0 created 00:00:09, never expire
  Type: static, Flags:
  NBMA address: 15.0.0.1
R4#show dm
R4#show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface: Tunnel0, IPv4 NHRP Details
Type:Spoke, NHRP Peers:1,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
-----
      1 15.0.0.1           10.0.0.1  UP 00:00:24  S

R4#
```

LAB: DMVPNphase1

```
R1#show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface: Tunnel0, IPv4 NHRP Details
Type:Hub, NHRP Peers:3,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
-----
  1 25.0.0.2          10.0.0.2    UP 00:00:23    D
  1 35.0.0.3          10.0.0.3    UP 00:00:18    D
  1 45.0.0.4          10.0.0.4    UP 00:00:02    D

R1#
```

LAB: DMVPNphase1

```
R1#show ip nhrp detail
10.0.0.2/32 via 10.0.0.2
  Tunnel0 created 00:00:50, expire 01:59:09
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 25.0.0.2
10.0.0.3/32 via 10.0.0.3
  Tunnel0 created 00:00:45, expire 01:59:14
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 35.0.0.3
10.0.0.4/32 via 10.0.0.4
  Tunnel0 created 00:00:29, expire 01:59:30
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 45.0.0.4
R1#
R1#
```

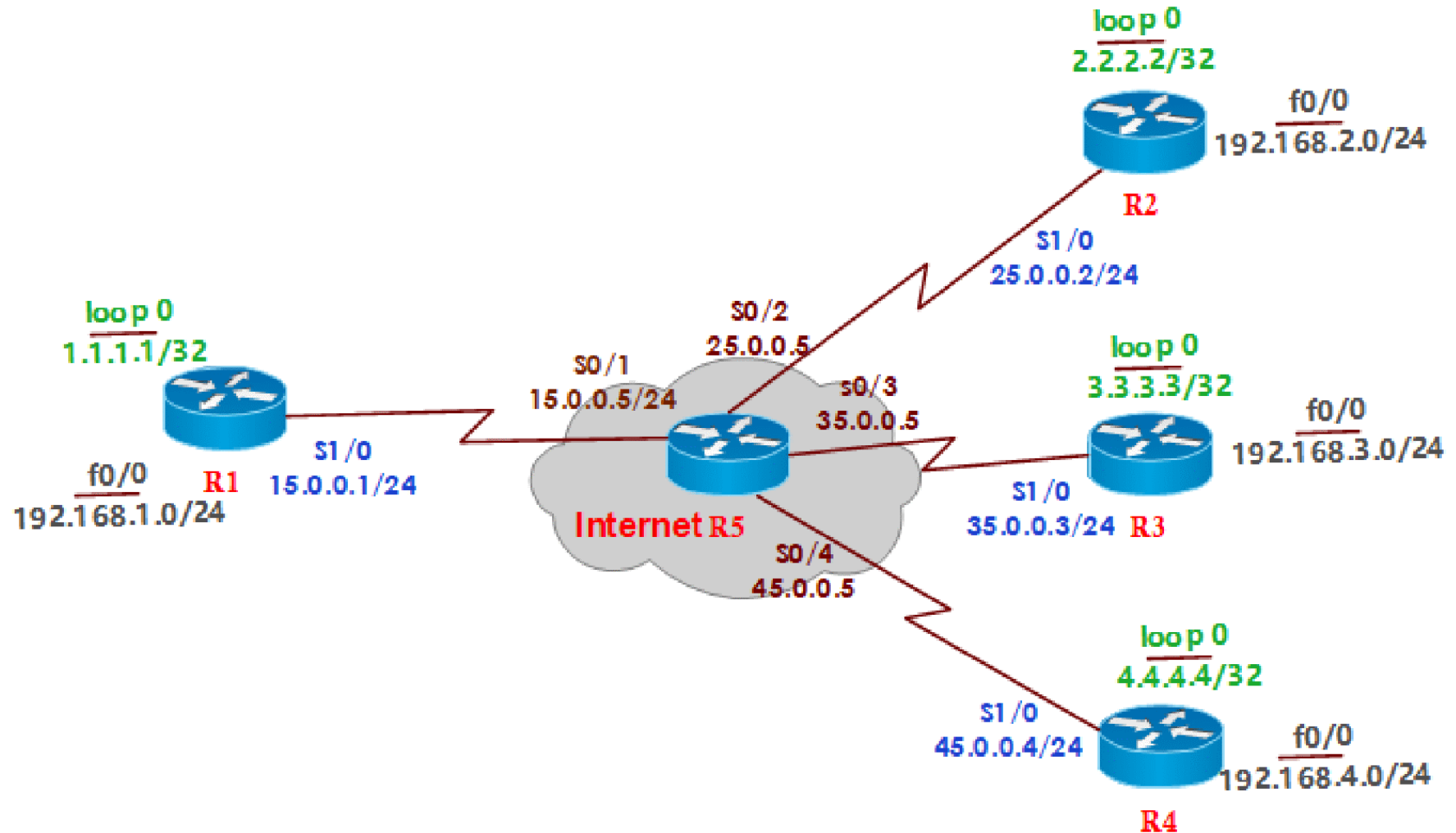
LAB: DMVPNphase1

```
R1#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/20/23 ms
R1#ping 10.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/19/20 ms
R1#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/19/21 ms
R1#
```

LAB: DMVPNphase1

```
R2#ping 10.0.0.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 35/38/43 ms
R2#ping 10.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/37/40 ms
R2#tr
R2#traceroute 10.0.0.4
Type escape sequence to abort.
Tracing the route to 10.0.0.4
VRF info: (vrf in name/id, vrf out name/id)
 1 10.0.0.1 23 msec 20 msec 20 msec
 2 10.0.0.4 40 msec 38 msec 37 msec
R2#
```

LAB:DMVPNPHASE1- With EIGRP Routing



LAB:DMVPNPHASE1- With EIGRP Routing

TASK: Configure EIGRP 100 on all routers to provide End to end reachability between LAN to LAN.

```
RI (config)# router eigrp 100
```

```
R1(config-router)# no auto-summary
```

```
R1(config-router)# network 192.168.1.0
```

```
R1(config-router)# network 10.0.0.0
```

```
R1(config-router)# exit
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R2(config)#router eigrp 100
```

```
R2(config-router)#no auto-summary
```

```
R2(config-router)# network 10.0.0.0
```

```
R2(config-router)# network 192.168.2.0
```

```
R2(config-router)# exit
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R3(config)#router eigrp 100
```

```
R3(config-router)# no auto-summary
```

```
R3(config-router)# network 10.0.0.0
```

```
R3(config-router)# network 192.168.3.0
```

```
R3(config-router)# exit
```

```
R4(config)#router eigrp 100
```

```
R4(config-router)#no auto
```

```
R4(config-router)# network 10.0.0.0
```

```
R4(config-router)# network 192.168.4.0
```

```
R4(config-router)# exit
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R4#sh ip ei nei
EIGRP-IPv4 Neighbors for AS(100)
R4#
```

LAB:DMVPNPHASE1- With EIGRP Routing

The ip nhrp map multicast dynamic command enables the forwarding of multicast traffic across the tunnel to dynamic spokes.

- This is usually required by routing protocols such as OSPF and EIGRP.
- In most cases, DMVPN is accompanied by a routing protocol to send and receive dynamic updates about the private networks.
- The ip nhrp map multicast dynamic command is not required if we are using Static NHRP mappings

LAB:DMVPNPHASE1- With EIGRP Routing

On HUB ROUTERS R1

```
R1 (config-if)# int tunnel 0
```

```
R1(config-if)# ip nhrp map multicast Dynamic
```

```
R1 (config-if)# exit
```

On SPOKES (R2/R3/R4)

```
Rx(config-if)#int tunnel 0
```

```
Rx(config-if)# ip nhrp map multicast 1 5.0.0.1
```

```
Rx(config-if)# exit
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R1#sh ip ei nei
EIGRP-IPv4 Neighbors for AS(100)
H   Address          Interface           Hold Uptime      SRTT   RTO   Q   Seq
                               (sec)           (ms)          Cnt  Num
2   10.0.0.4          Tu0                12 00:00:16      1 5000  1  0
1   10.0.0.3          Tu0                12 00:00:30      1 5000  1  0
0   10.0.0.2          Tu0                14 00:00:41      1 5000  1  0
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R1#sh ip route eigrp
```

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
D 192.168.2.0/24 [90/26882560] via 10.0.0.2,00:05:14,Tunnel0
```

```
D 192.168.3.0/24 [90/26882560] via 10.0.0.3,00:05:06,Tunnel0
```

```
D 192.168.4.0/24 [90/26882560] via 10.0.0.4,00:05:06,Tunnel0
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R1#sh ip route eigrp
```

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
D 192.168.2.0/24 [90/26882560] via 10.0.0.2,00:05:14,Tunnel0
```

```
D 192.168.3.0/24 [90/26882560] via 10.0.0.3,00:05:06,Tunnel0
```

```
D 192.168.4.0/24 [90/26882560] via 10.0.0.4,00:05:06,Tunnel0
```

```
R4#sh ip route eigrp
```

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
D 192.168.1.0/24 [90/26882560] via 10.0.0.1,00:04:38,Tunnel0
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R2#sh ip route eigrp
```

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
D 192.168.1.0/24 [90/26882560] via 10.0.0.1,00:05:34,Tunnel0
```

LAB:DMVPNPHASE1- With EIGRP Routing

- Note: If there is no neighborship coming, try shutting down and use no shutdown tunnel interface.

Here R2 advertises R1 the routes and R1 will not be advertising back on the same interface to R3/R4 because of split horizon rule the split horizon rule prohibits a router from advertising a route through an interface that the router itself uses to reach the destination.

- In order to disable the split horizon behavior, use the no ip split-horizon eigrp as-number interface command

LAB:DMVPNPHASE1- With EIGRP Routing

Some important points to remember about EIGRP split horizon are:

1. Split horizon behavior is turned on by default
2. When you change the EIGRP split horizon setting on an interface, it resets all adjacencies with EIGRP neighbors reachable over that interface.
3. Split horizon should only be disabled on a hub site in a hub-and-spoke network.
4. Disabling split horizon on the spokes radically increases EIGRP memory consumption on the hub router, as well as the amount of traffic generated on the spoke routers.
5. The EIGRP split horizon behavior is not controlled or influenced by the ip split-horizon command.
6. This is because split horizon is preventing R2 from distributing the routes from a spoke router back to the other spoke routers. This can easily be fixed by disabling split horizon on the tunnel interface on the hub router

LAB:DMVPNPHASE1- With EIGRP Routing

```
R1 (config)#int tunnel 0
```

```
R1 (config-if)#no ip split-horizon eigrp 100
```

```
R1 (config-if)#exit
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R4#sh ip route eigrp
```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

```
D 192.168.1.0/24 [90/26882560] via 10.0.0.1,00:06:41,TunnelO
```

```
D 192.168.2.0/24 [90/28162560] via 10.0.0.1,00:00:14, TunnelO
```

```
D 192.168.3.0/24 [90/28162560] via 10.0.0.1,00:00:14, TunnelO
```

```
R2#sh ip route eigrp
```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

```
D 192.168.1.0/24 [90/26882560] via 10.0.0.1,00:06:58,TunnelO
```

```
D 192.168.3.0/24 [90/28162560] via 10.0.0.1,00:00:22,TunnelO
```

```
D 192.168.4.0/24 [90/28162560] via 10.0.0.1,00:00:22,TunnelO
```

LAB:DMVPNPHASE1- With EIGRP Routing

```
R2#traceroute 192.168.4.4 source 192.168.2.2
```

Type escape sequence to abort.

Tracing the route to 192.168.4.4

VRF info: (vrf in name/id, vrf out name/id)

```
1 10.0.0.1 100 msec 132 msec 144 msec
```

```
2 10.0.0.4 156 msec 176 msec 148 msec
```

```
R2#traceroute 192.168.3.3 source 192.168.2.2
```

Type escape sequence to abort.

Tracing the route to 192.168.3.3

VRF info: (vrf in name/id,vrf out name/id)

```
1 10.0.0.1 120 msec 128 msec 88 msec
```

```
2 10.0.0.3 184 msec 184 msec 168 msec
```

LAB:DMVPNPHASE1- With EIGRP Routing

TASK: Reconfigure IP Splithorizaon rule and ensure that spoke to spoke have reachability (use Summarzation)

R1

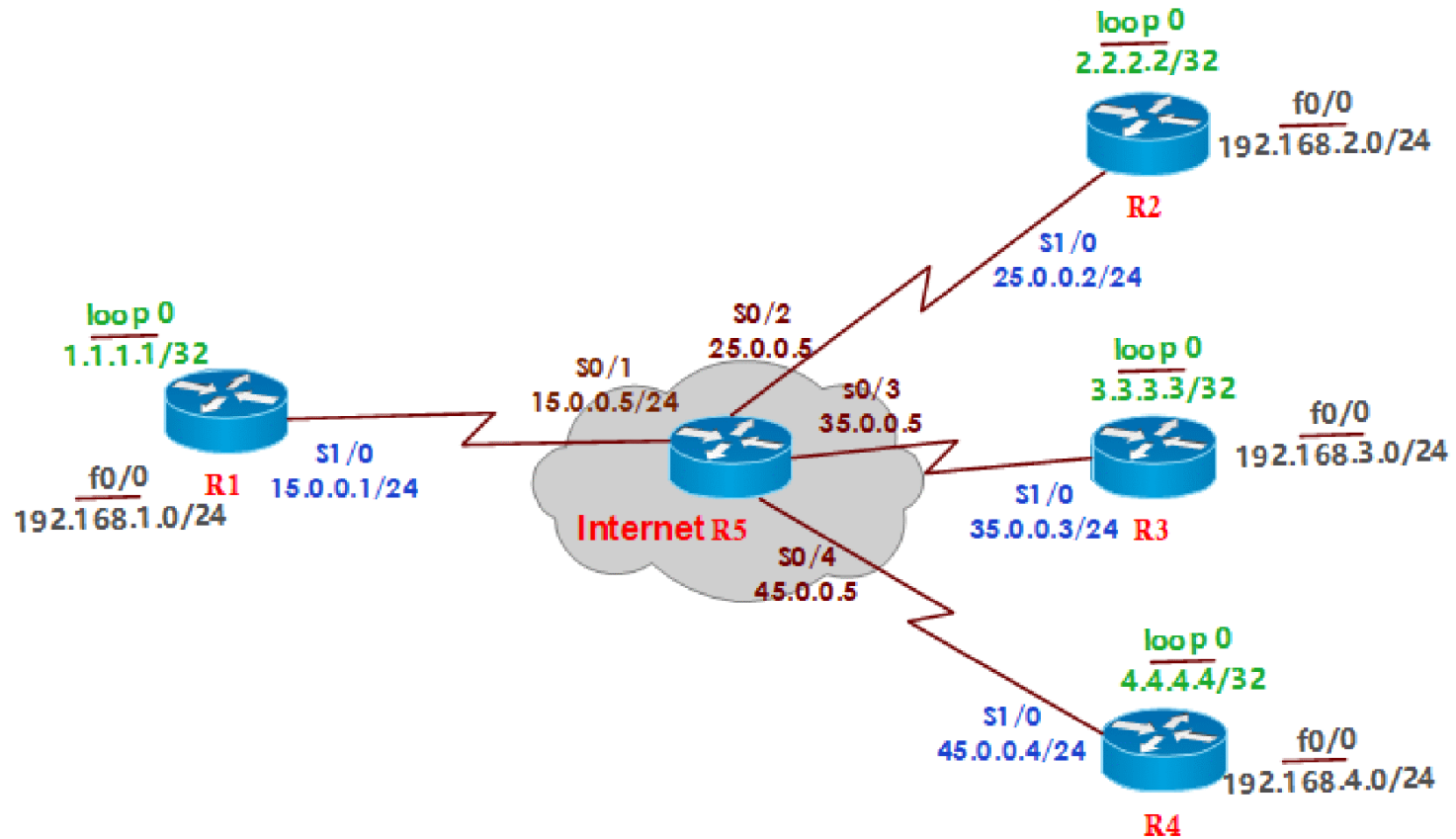
Int tunnel 0

Ip split-horizon eigrp 100

Ip summary-address eigrp 100 0.0.0.0 0 .0.0.0

DMVPN - Phases 1.2.3

- When Cisco did major changes to DMVPN - introduces in terms of Phases (1, 2, 3)
- In every phase added new features, more scalability & different behavior.



DMVPN - Phase 1

Spoke-to-spoke GRE tunnels cannot be build

- mGRE interface on the hub
- Point-to-point GRE interfaces on the spokes
- All data-plane traffic between spokes routed via the hub

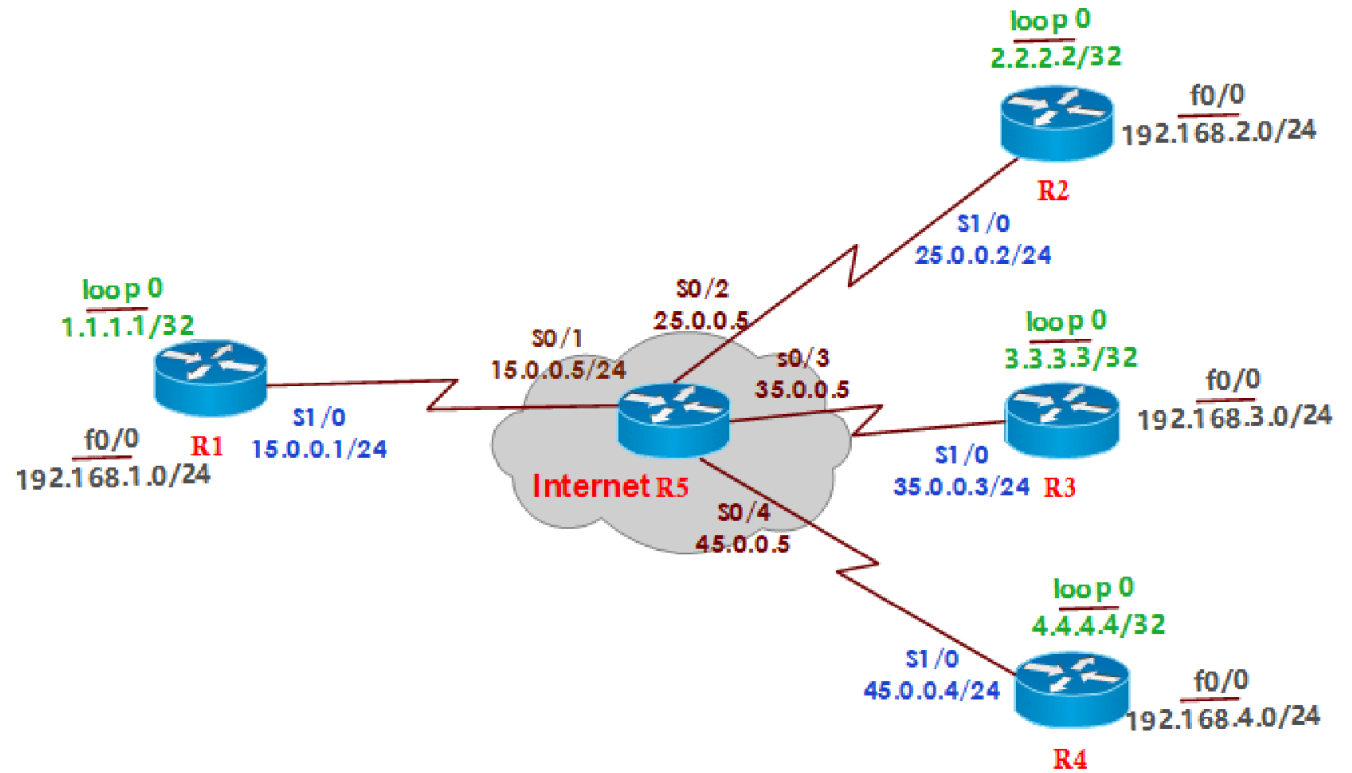
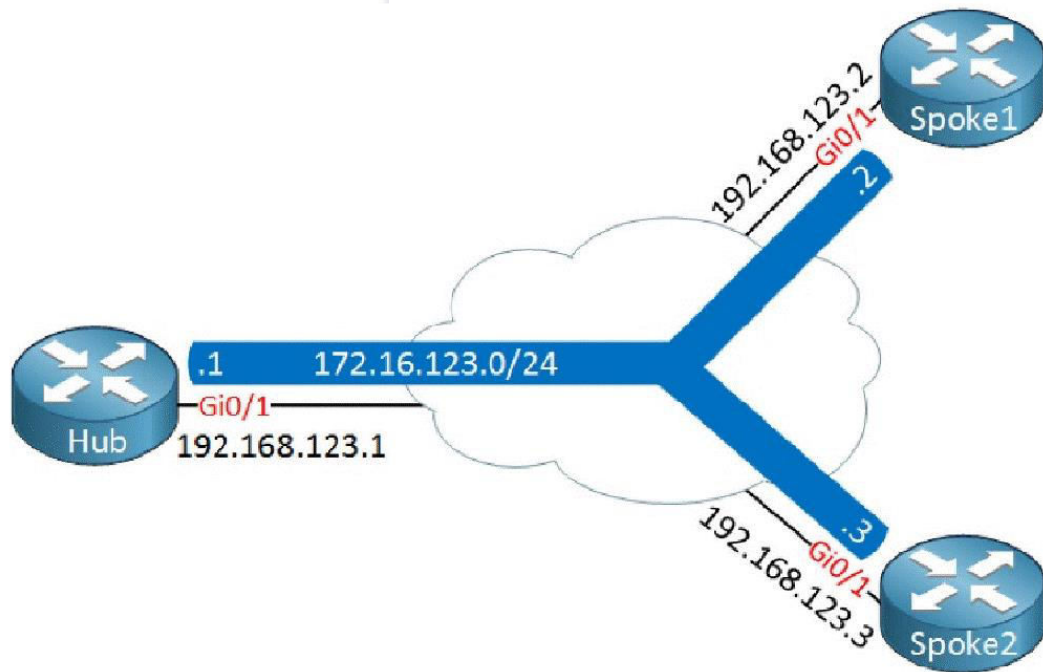
NHRP Role

- Builds the static spoke-to-hub GRE tunnel
- Builds the dynamic hub-to-spoke GRE tunnel

Overlay Routing

- On spokes all routes have a next-hop the hub
- Summarization/default routing at hub is allowed (Limit Routes on spokes)

DMVPN - Phase 1



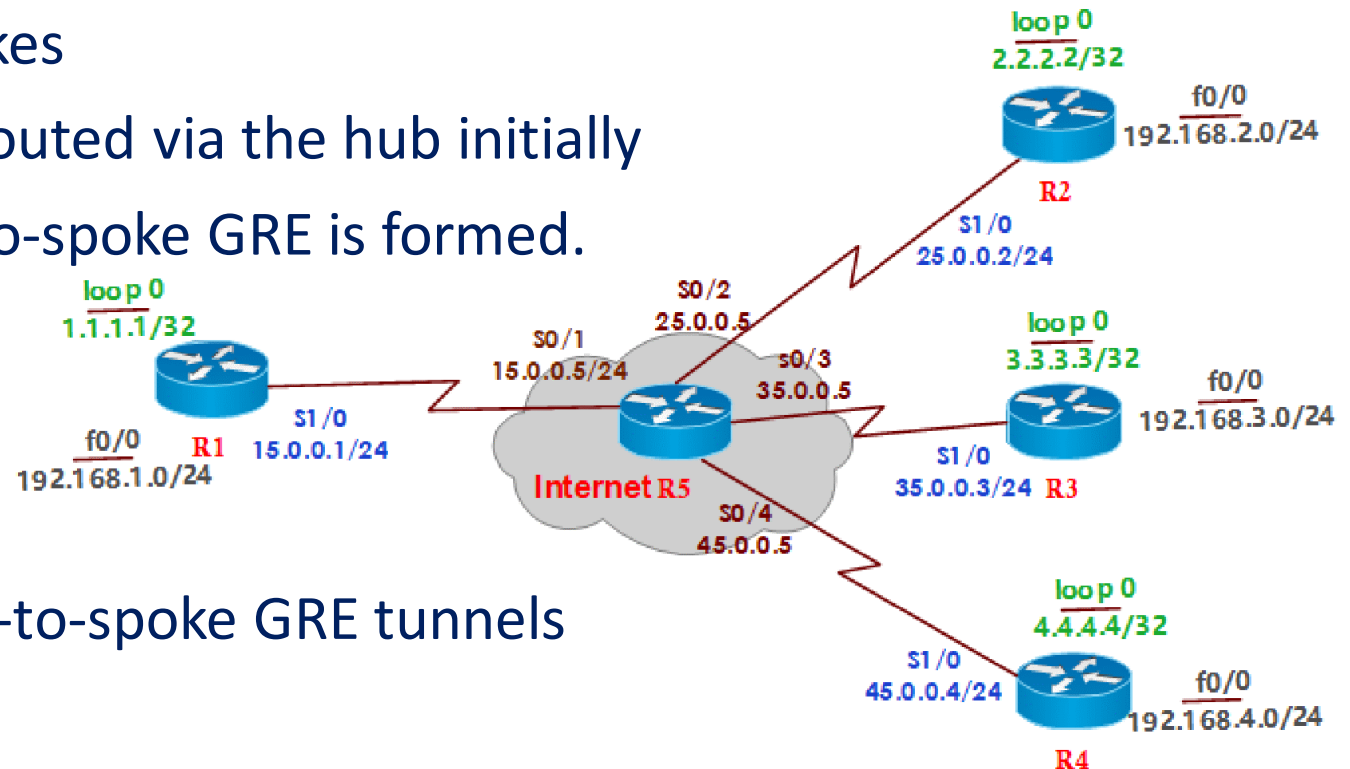
DMVPN - Phase 2

Spoke-to-spoke GRE tunnels can be build

- mGRE interface on the hub and spokes
- Data-plane traffic between spokes routed via the hub initially
- Re-routed spoke-to-spoke if spoke-to-spoke GRE is formed.

NHRP Role

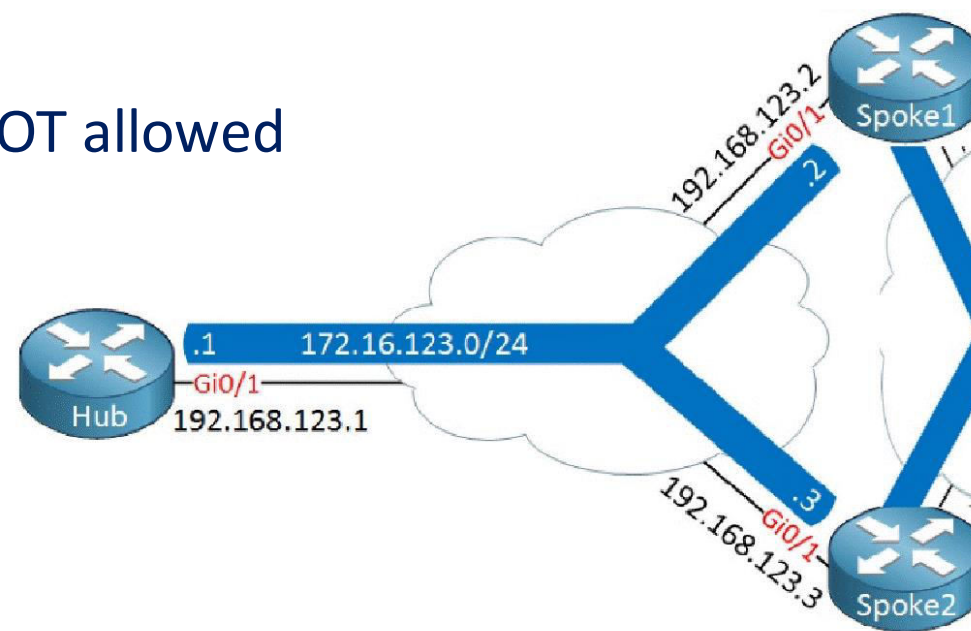
- Same like in Phase 1
- Additionally it builds dynamic spoke-to-spoke GRE tunnels



DMVPN - Phase 2

Overlay Routing

- The hub preserves the next-hop
- On spokes routes have a next-hop of the hub for hub prefixes
- On spokes routes have a next-hop of remote spokes for remote spokes prefixes
- Summarization/default routing at hub is NOT allowed



DMVPN Phase 1 - Tunnel Configuration

HUB ROUTER (R1)

```
R1(config)# int tunnel 0
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# tunnel source s2/0
R1(config-if)# tunnel mode gre multipoint
R1(config-if)# ip nhrp network-id 1
R1(config-if)# ip nhrp map multicast dynamic
R1(config-if)# no IP Split-horizon eigrp 100
```

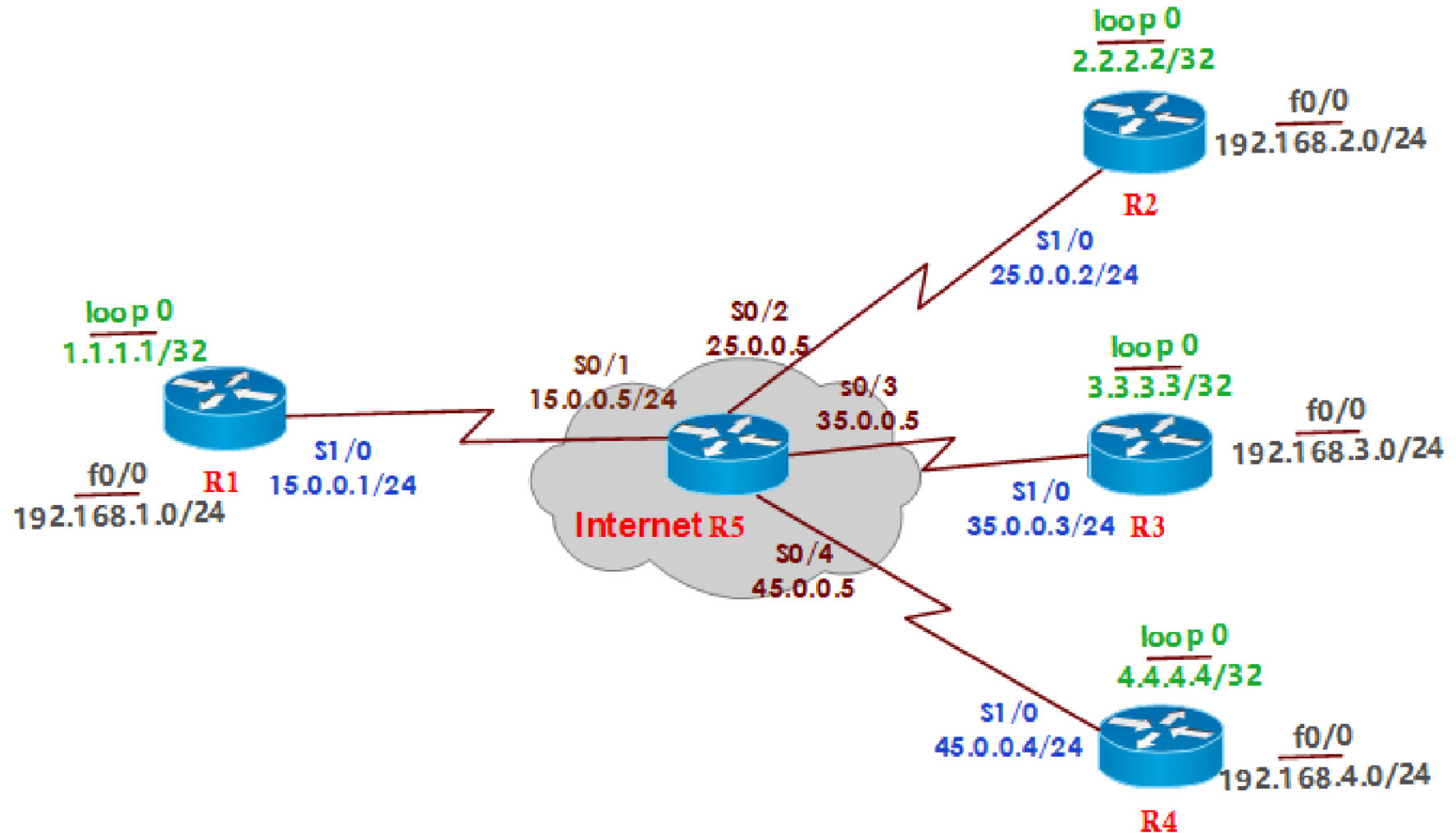
On all Routers

```
(config)# Router eigrp 100
(config-router) # No auto-summary
(config-router) # Network 192.168.X.0
(config-router) # Network 10.0.0.0
```

On SPOKES (R2/R3/R4)

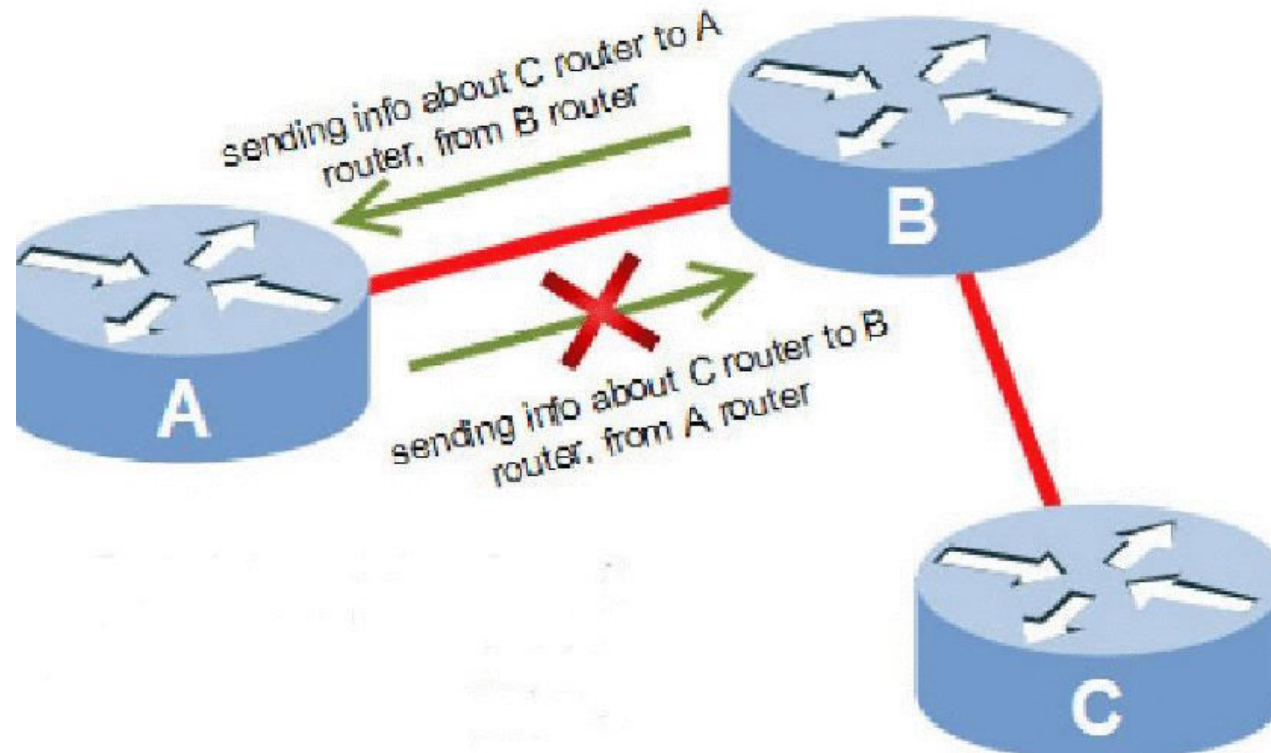
```
Rx(config)# interface tunnel 0
Rx(config-if)# ip address 10.0.0.X 255.255.255.0
Rx(config-if)# tunnel source s2/0
Rx(config-if)# tunnel Destination 15.0.0.1
Rx(config-if)# ip nhrp network-id 2
Rx(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
Rx(config-if)# ip nhrp nhs 10.0.0.1
R1(config-if)# ip nhrp map multicast 15.0.0.1
```

DMVPN Phase 1 - Tunnel Configuration



IGP - Split Horizon Rule

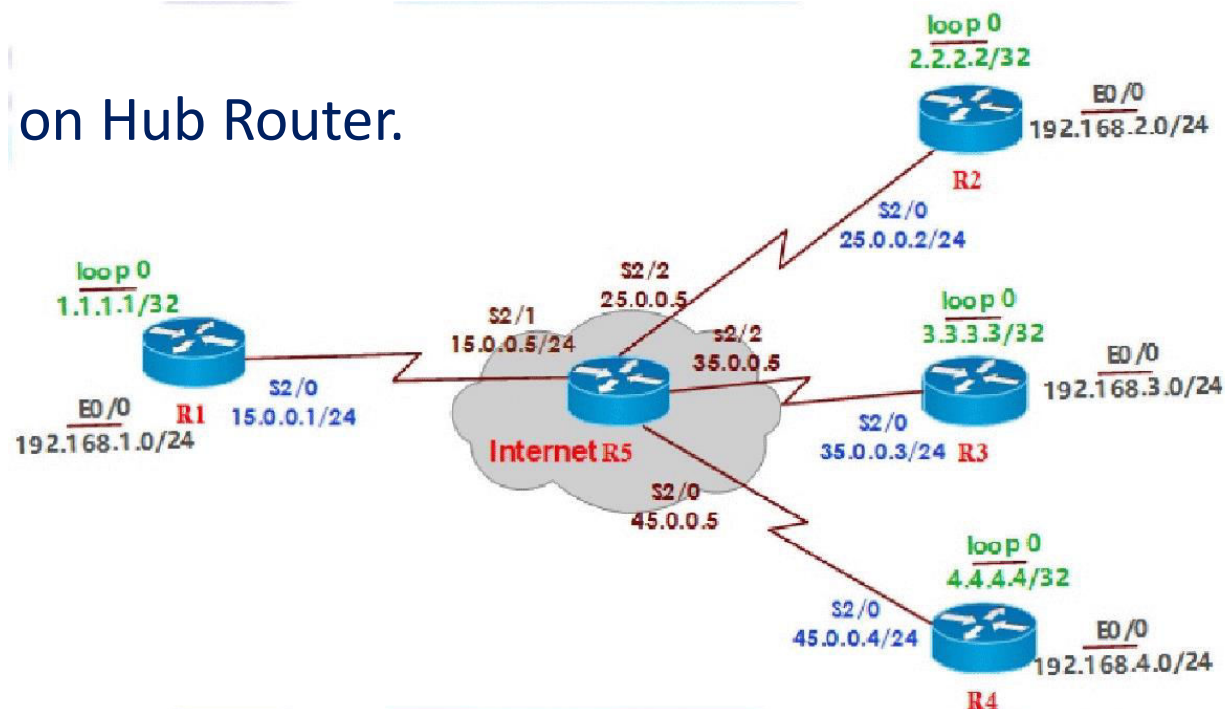
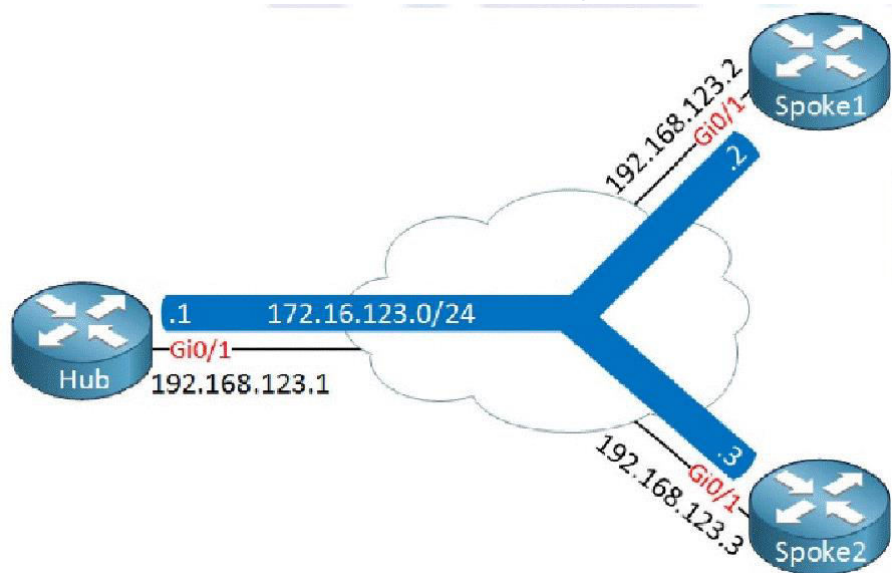
- Method of preventing routing loops in distance-vector routing protocols
- Prohibiting a router from advertising a route back onto the interface from which it was learned.



IGP - Split Horizon Rule

Split Horizon - DMVPN

- Spokes must receive specific routes for all remote spoke subnets.
- By default Split Horizon in IGP prevent Spokes to learn routes from other spokes in Phase-1
- Solution - Disable Split Horizon Rule on Hub Router.



DMVPN Phase -1 - Disable Split Horizon

R2#sh ip route eigrp

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

D 192.168.1.0/24 [90/26882560] via 10.0.0.1, 00:05:34, Tunnel0

EIGRP / RIPv2 ROUTING (HUB)

R1(config)#int tunnel 0

R1(config-if)#no ip split-horizon eigrp 100

R1(config-if)#exit

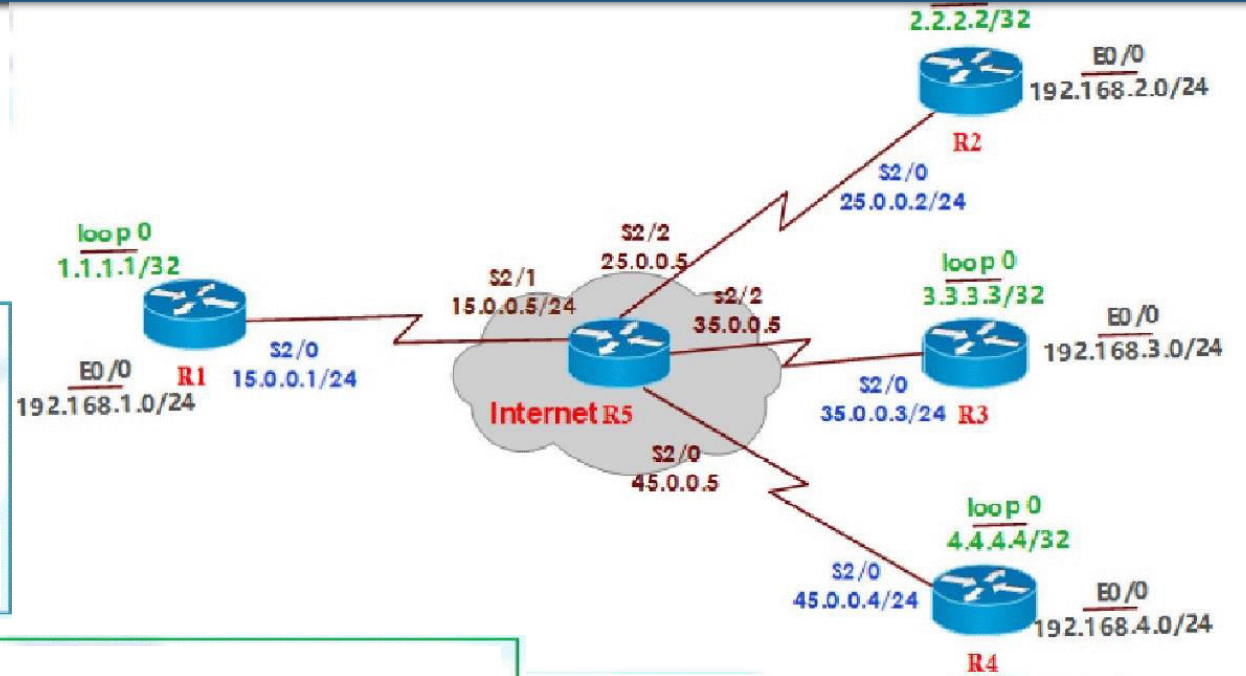
R2#sh ip route eigrp

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

D 192.168.1.0/24 [90/26882560] via 10.0.0.1, 00:06:58, Tunnel0

D 192.168.3.0/24 [90/28162560] via 10.0.0.1, 00:00:22, Tunnel0

D 192.168.4.0/24 [90/28162560] via 10.0.0.1, 00:00:22, Tunnel0



DMVPN Phase -1 - Disable Split Horizon

```
R2#traceroute 192.168.4.4 source 192.168.2.2
```

```
Type escape sequence to abort.
```

```
Tracing the route to 192.168.4.4
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
 1 10.0.0.1 108 msec 152 msec 128 msec
```

```
 2 10.0.0.4 156 msec 172 msec 172 msec
```

DMVPN Phase -1 - with OSPF Routing

- By default OSPF treats tunnel interface as Point to point (even if we configure Multipoint GRE).
- hence you will be so many console messages on routers saying neighbor ship established and then goes down repeatedly

```
*Mar 1 00:10:07.535: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on Tunnel 0 from INIT to DOWN, Neighbor Down: Adjacency forced to reset
```

```
R1#sh ip ospf int tunnel 0
```

```
Tunnel0 is up, line protocol is up
```

```
Internet Address 10.0.0.1/24, Area 0, Attached via Network Statement
```

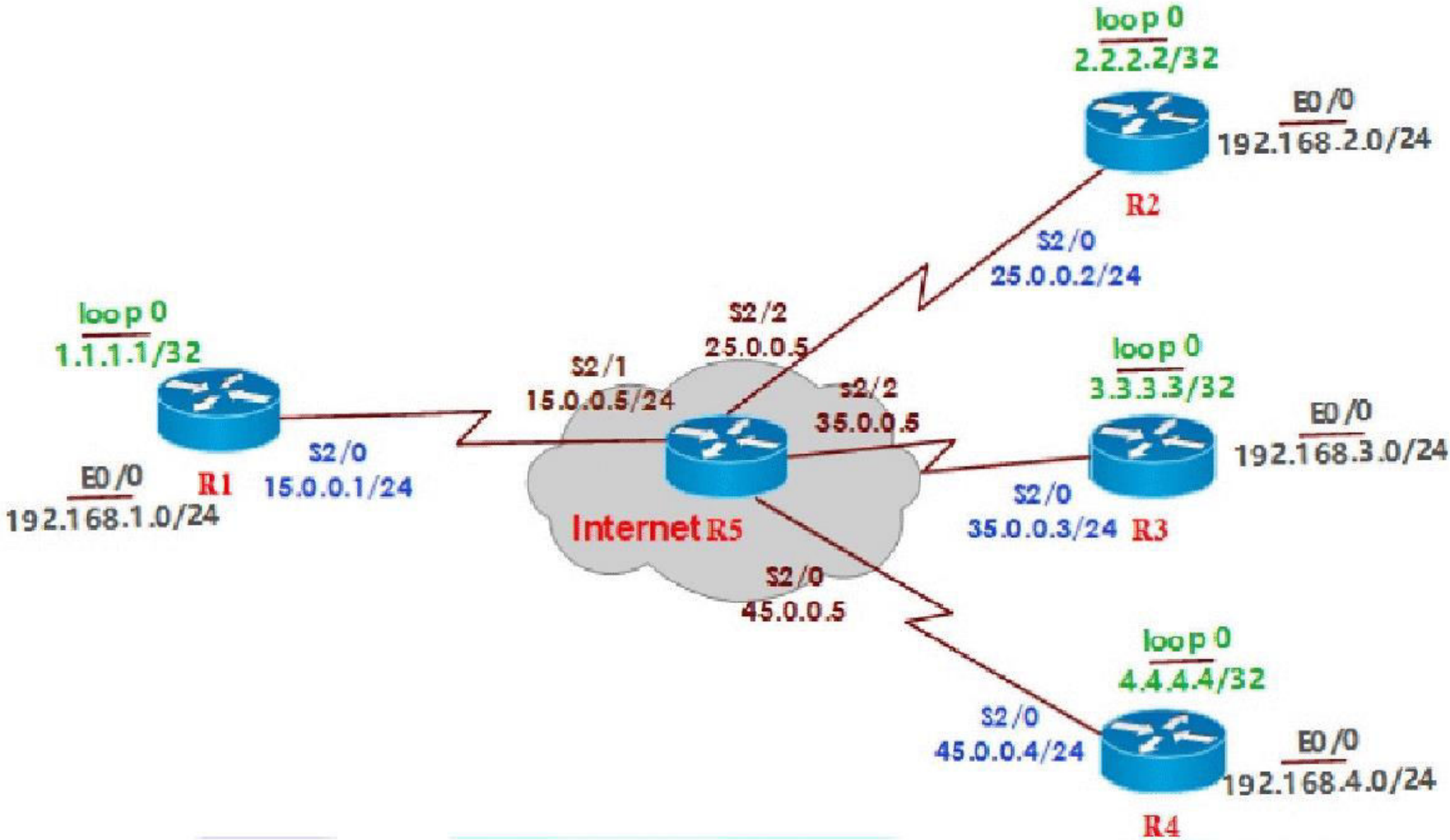
```
Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 1000
```

```
Topology-MTID Cost Disabled Shutdown Topology Name
```

```
0 1000 no no Base
```

```
Transmit Delay is 1 sec, State POINT_TO_POINT
```

DMVPN Phase -1 - with OSPF Routing



DMVPN Phase -1 - with OSPF Routing

- To fix this, we need to change the network type on all routers to OSPF Point to Multipoint.
- OSPF Neighbor will not come up, until it matches the network type.(technically Spokes are P2P)

OSPF ROUTING (Hub & Spokes)

```
Rx(config)# int tunnel 0
```

```
Rx(config-if)#ip ospf network point-to-multipoint
```

```
Rx(config-if)#end
```

DMVPN -Phase 2 - Tunnel Configuration

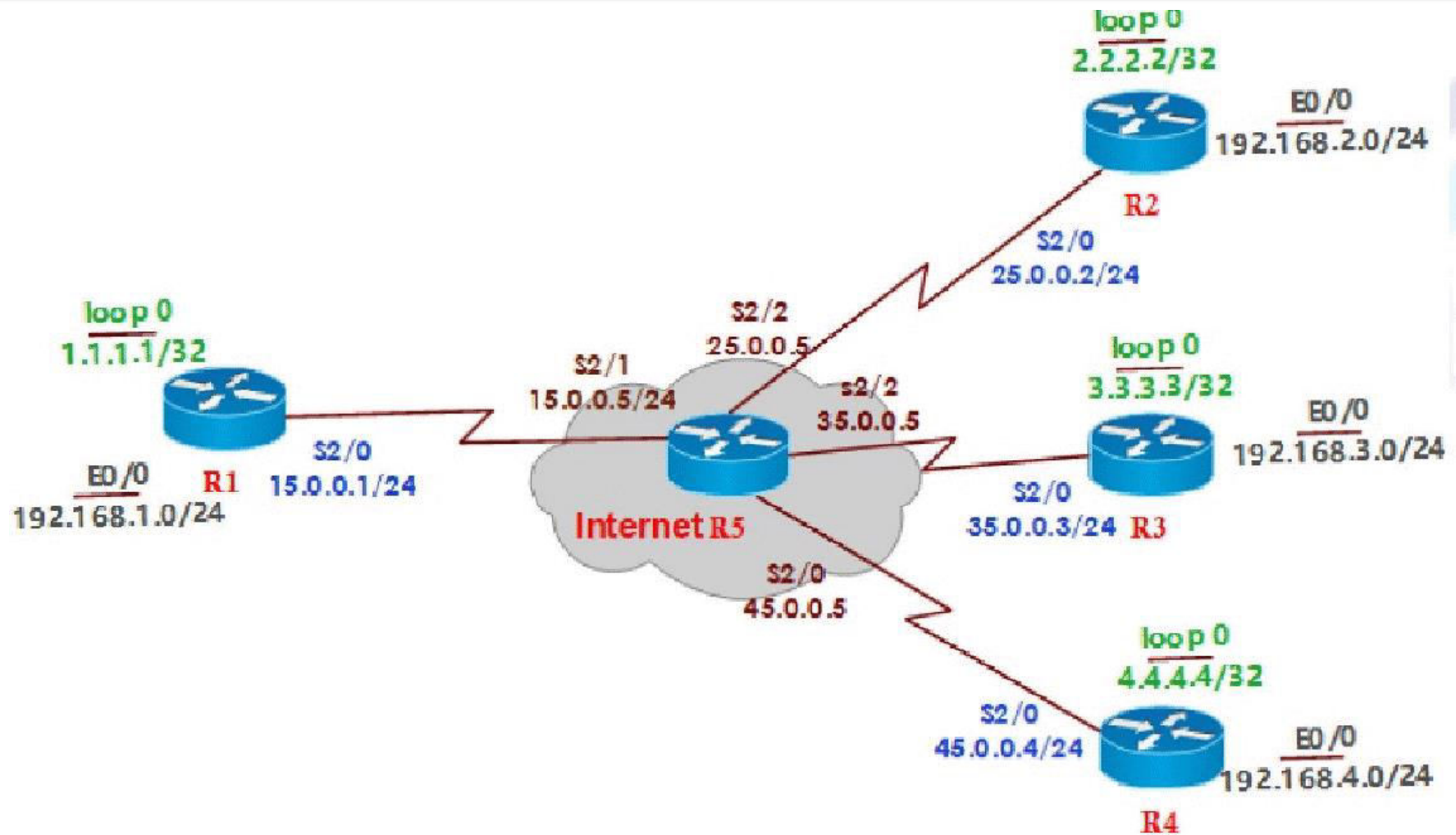
HUB ROUTER (R1)

```
R1(config)# int tu 0
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# tunnel source s2/0
R1(config-if)# tunnel mode gre multipoint
R1(config-if)# ip nhrp network-id 1
R1(config-if)# ip nhrp map multicast dynamic
```

On SPOKES (R2/R3/R4)

```
Rx(config)# interface tunnel 0
Rx(config-if)# ip address 10.0.0.X 255.255.255.0
Rx(config-if)# tunnel source s2/0
Rx(config-if)# tunnel mode gre multipoint
Rx(config-if)# ip nhrp network-id 2
Rx(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
Rx(config-if)# ip nhrp nhs 10.0.0.1
Rx(config-if)# ip nhrp map multicast Dynamic
```

DMVPN -Phase 2 - Tunnel Configuration



DMVPN Phase 2 - OSPF Routing

- By default OSPF treats tunnel interface as Point to point (even if we configure Multipoint GRE).
- Hence you will be so many console messages on routers saying Neighborship established and then goes down repeatedly.

```
*Mar 1 00:10:07.535: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on Tunnel 0 from INIT to DOWN, Neighbor Down: Adjacency forced to reset
```

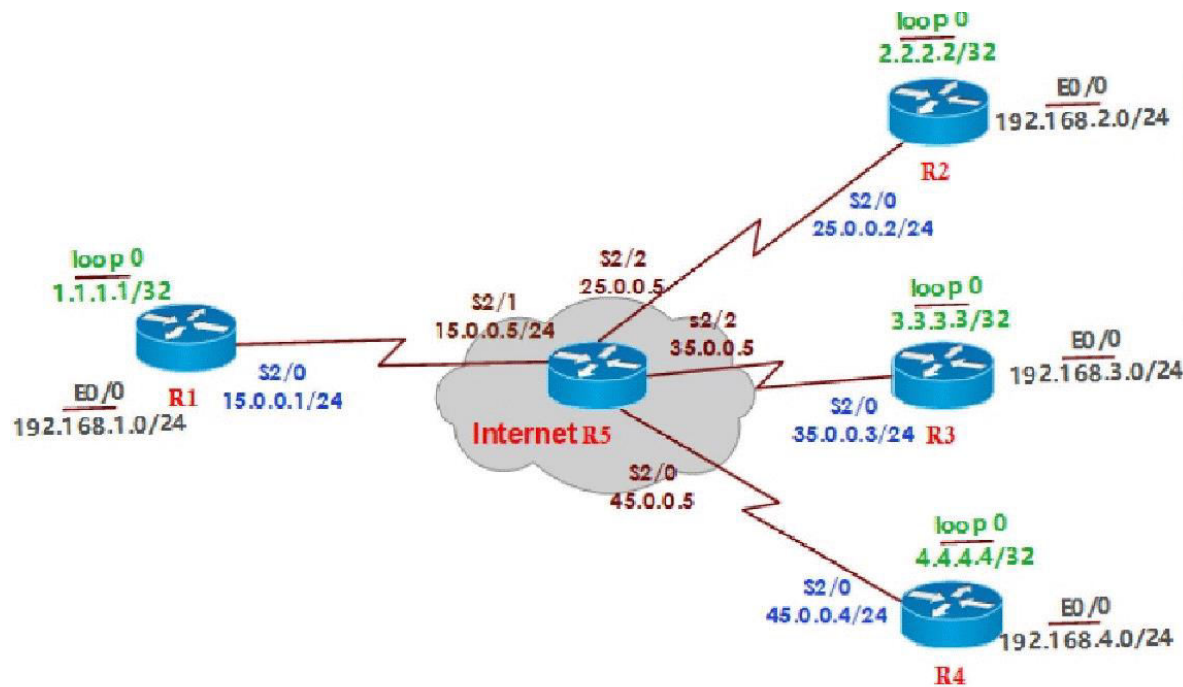
OSPF ROUTING (Hub & Spokes)

```
Rx(config)# int tunnel 0  
Rx(config-if)#ip ospf network Broadcast  
Rx(config-if)#end
```

OSPF ROUTING (on all spokes)

```
Rx(config)# int tunnel 0  
Rx(config-if)#ip ospf priority 0  
Rx(config-if)#end
```

DMVPN Phase 2 - OSPF Routing



To fix this, we need to change the network type to OSPF Point to Multipoint or broadcast networks on all routers.

- Also ensure that R1- HUB should become DR and **R2/R3/R4- Spokes should be DRother..**

DMVPN Phase 2 - EIGRP ROUTING

EIGRP ROUTING (HUB)

```
R1(config)#int tunnel 0
```

```
R1(config-if)#no ip split-horizon eigrp 100
```

```
R1(config-if)#exit
```

```
R2#sh ip route eigrp
```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

```
D 192.168.1.0/24 [90/26905600] via 10.0.0.1, 00:03:04, Tunnel0
```

```
D 192.168.3.0/24 [90/28185600] via 10.0.0.1, 00:00:38, Tunnel0
```

```
D 192.168.4.0/24 [90/28185600] via 10.0.0.1, 00:00:38, Tunnel0
```

```
R2#traceroute 192.168.3.3 source 192.168.2.2
```

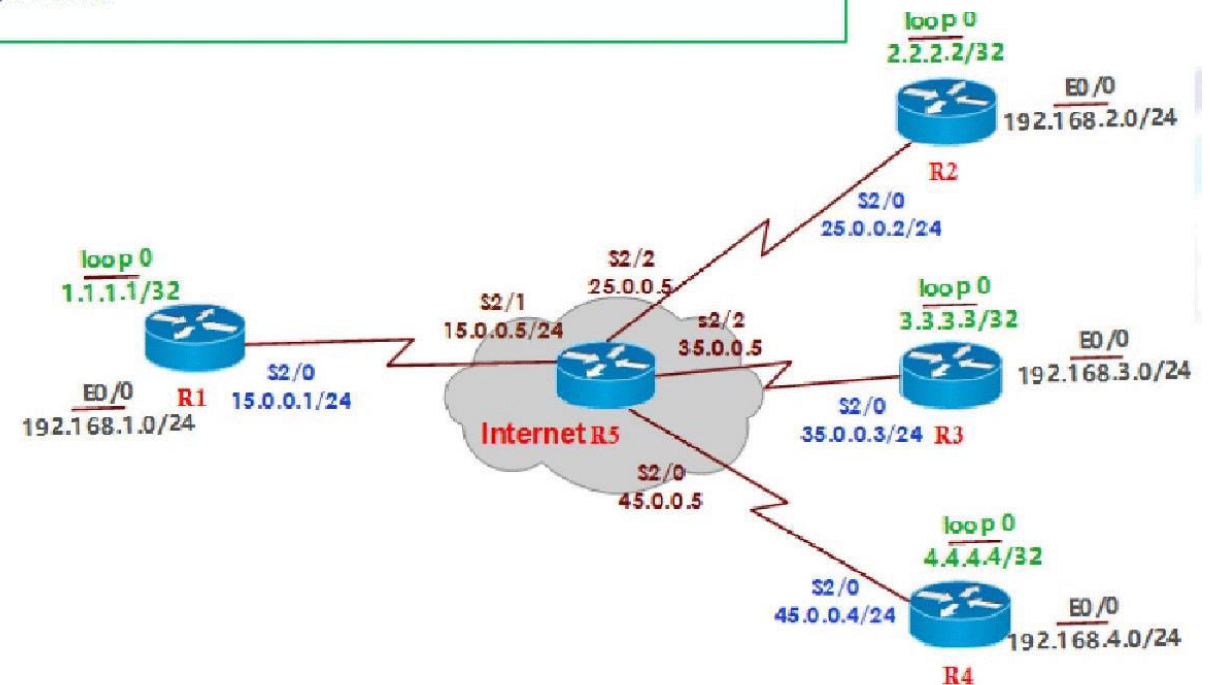
Type escape sequence to abort.

Tracing the route to 192.168.3.3

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.0.0.1 20 msec 20 msec 19 msec
```

```
 2 10.0.0.3 34 msec 40 msec 39 msec
```



DMVPN Phase 2 - EIGRP ROUTING

- By default for all routes learned on spokes from other spokes has next-hop of R1.
- As spokes advertises routes to R1(hub) R1, changes next-hop and send back to spokes.
- That's why all the traffic between spoke to spoke LAN is going via Hub(here we are using phase-2 and all traffic between spokes need to go directly after the first packet.

DMVPN Phase 2 - EIGRP ROUTING

- By default for all routes learned on spokes from other spokes has next-hop of R1.
- As spokes advertises routes to R1(hub) R1, changes next-hop and send back to spokes.
- That's why all the traffic between spoke to spoke LAN is going via Hub(here we are using phase-2 and all traffic between spokes need to go directly after the first packet.
- To change the behavior we can tell EIGRP Hub router not to change the next-hop and advertise with same next-hop to spokes.

```
R2#show ip route eigrp
```

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

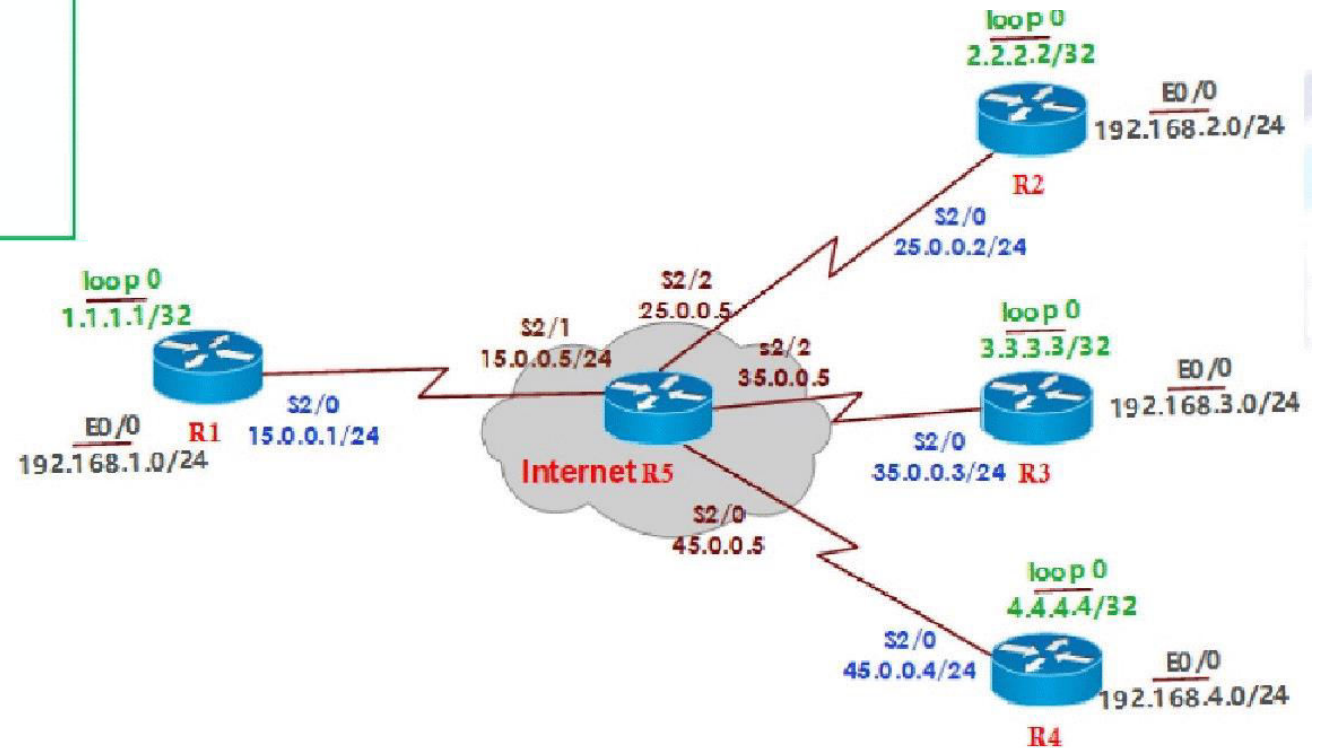
```
D 192.168.1.0/24 [90/26905600] via 10.0.0.1, 00:00:14, Tunnel0
D 192.168.3.0/24 [90/28185600] via 10.0.0.3, 00:00:12, Tunnel0
D 192.168.4.0/24 [90/28185600] via 10.0.0.4, 00:00:14, Tunnel0
```

```
R1(config)# interface tunnel 0
```

```
R1(config-if)# no ip next-hop-self eigrp 100
```

DMVPN Phase 2 - EIGRP ROUTING

```
R2#traceroute 192.168.3.3 source 192.168.2.2
Type escape sequence to abort.
Tracing the route to 192.168.3.3
VRF info: (vrf in name/id, vrf out name/id)
 1 10.0.0.3 20 msec 19 msec 20 msec
```



DMVPN Phase 1 & 2 -Limitations

Phase -1

- No Spoke to Spoke Tunnels built (all Traffic goes via HUB)
- Summarization/default routing at hub is allowed (Limit Routes on spokes)

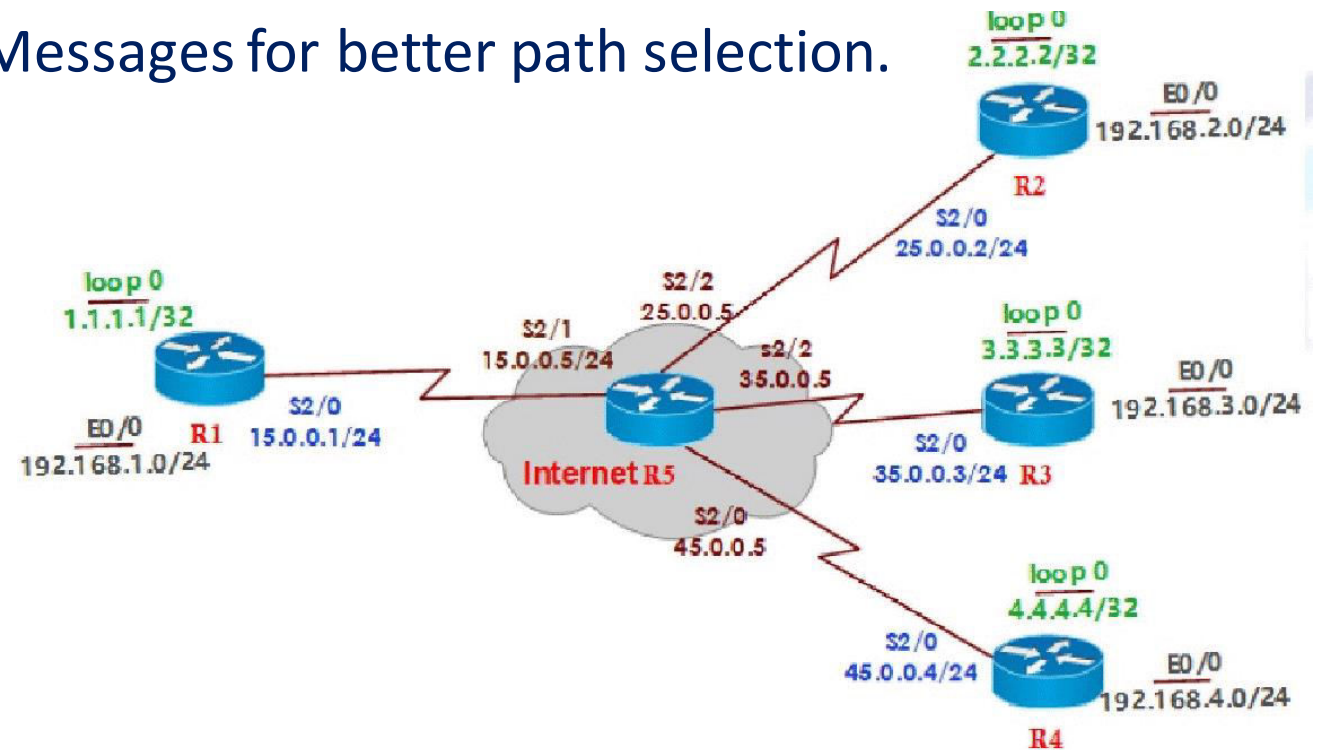
Phase -2

- Spoke to Spoke Tunnels built.
- All Traffic goes via HUB initially only (Re-routed spoke-to-spoke if spoke-to-spoke GRE is formed)
- On spokes routes have a next-hop of remote spokes for remote spokes prefixes.
- Summarization/default routing at hub is NOT allowed (Routing scalability issues)

DMVPN Phase 1 & 2 -Limitations

Phase -3

- Spoke to Spoke Tunnels built
- Spokes with next-hop as HUB in Routing table (allow Summarization/Default Routes)
- Uses NHRP redirect / Shortcut Messages for better path selection.



DMVPN Phase 3

- Spoke to Spoke direct communication allowed with better scalability using NHRP Redirects*
- Uses NHRP redirect / Shortcut Messages for better path selection.
- mGRE interface on the hub and spokes.(like Phase 2)
- Spoke-to-spoke GRE tunnels can be build (like Phase 2)
- Data-plane traffic between spokes routed via the hub
 - Initially Re-routed spoke-to-spoke if spoke-to-spoke GRE is formed.
- NHRP Role - Same like in Phase 2
- Overlay Routing (like in Phase 1)
 - On spokes all routes have a next-hop the hub
 - Summarization/default routing at hub is allowed

DMVPN Phase 3 - Tunnel Configuration

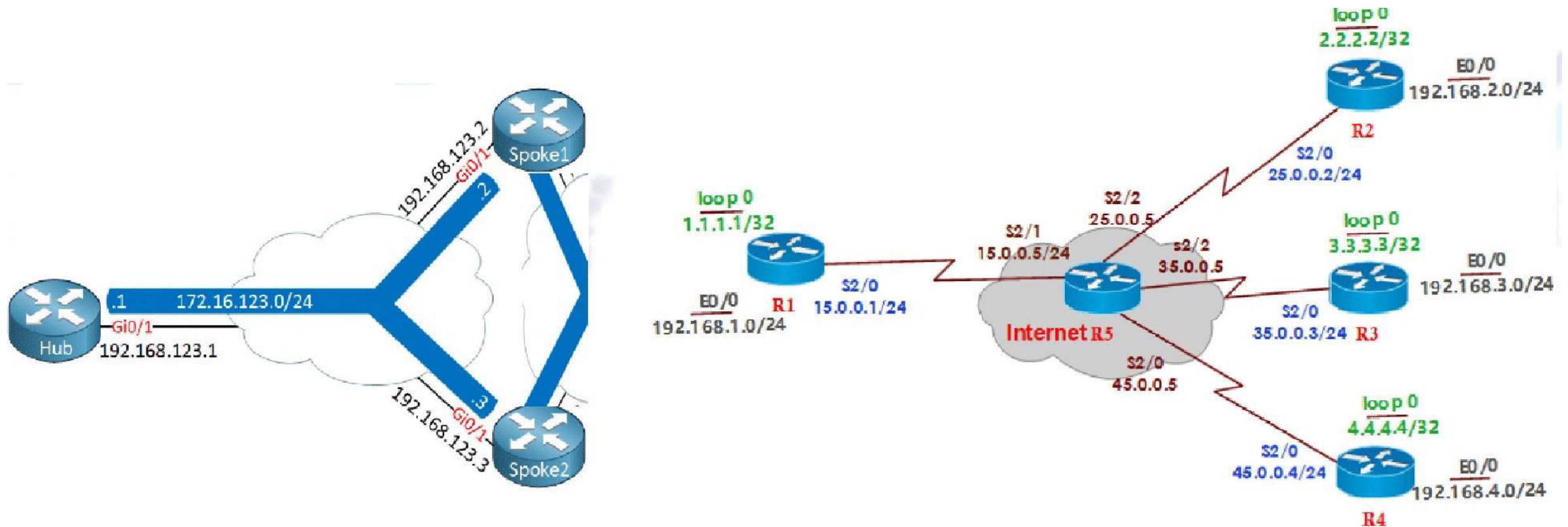
HUB ROUTER (R1)

```
R1(config)# int tu 0
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# tunnel source s2/0
R1(config-if)# tunnel mode gre multipoint
R1(config-if)# ip nhrp network-id 1
R1(config-if)# ip nhrp map multicast dynamic
```

On SPOKES (R2/R3/R4)

```
Rx(config)# interface tunnel 0
Rx(config-if)# ip address 10.0.0.X 255.255.255.0
Rx(config-if)# tunnel source s2/0
Rx(config-if)# tunnel mode gre multipoint
Rx(config-if)# ip nhrp network-id 2
Rx(config-if)# ip nhrp map 10.0.0.1 15.0.0.1
Rx(config-if)# ip nhrp nhs 10.0.0.1
Rx(config-if)# ip nhrp map multicast 15.0.0.1
```

DMVPN Phase 3 - Tunnel Configuration



Configuration on tunnel interface is same we did in the Phase-2

DMVPN Phase 3 - EIGRP ROUTING

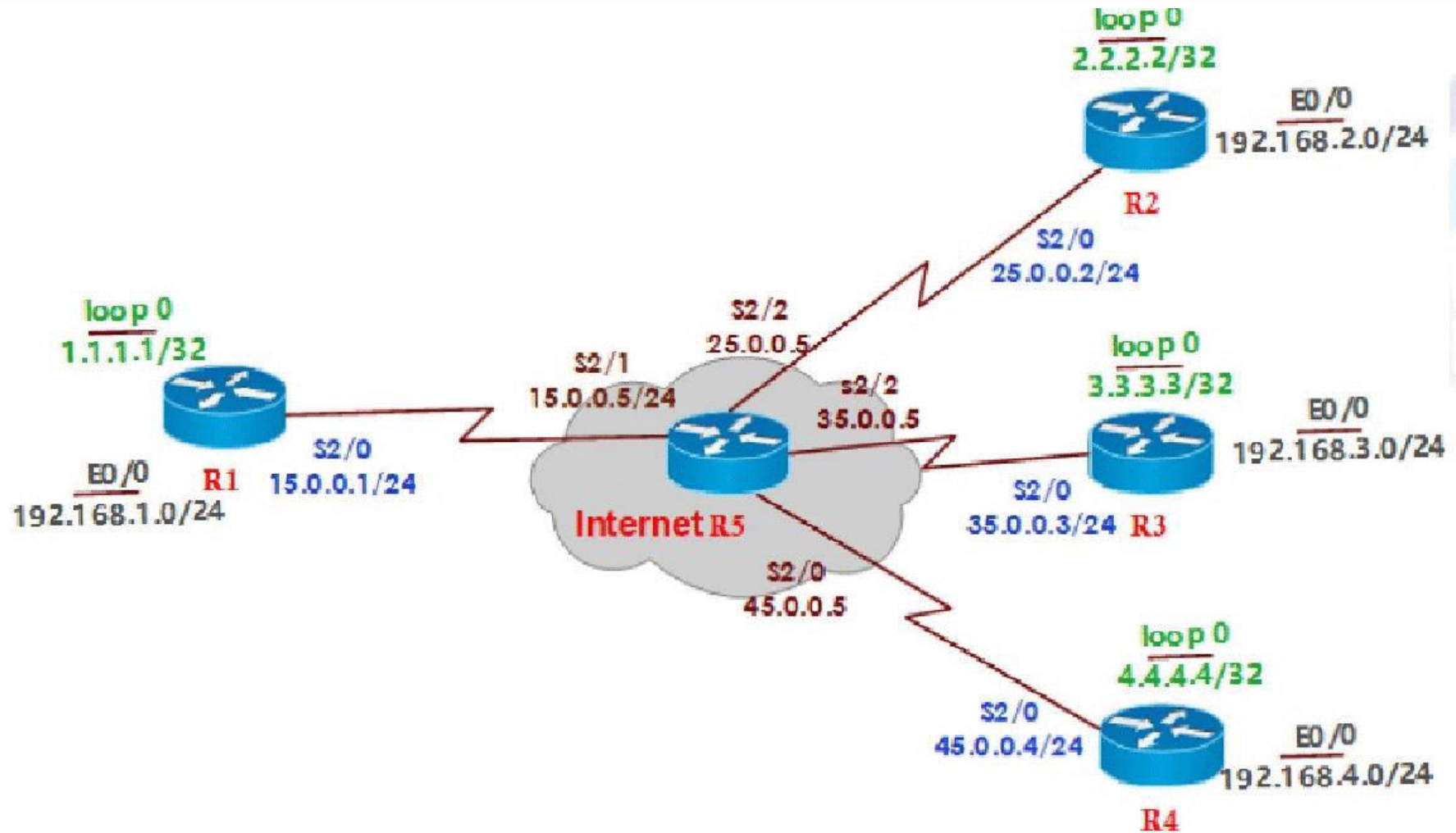
ON HUB – R1

```
R1(config)# int tunnel 0
R1(config-if)# no ip split-horizon eigrp 100
R1(config-if)# ip next-hop-self eigrp 100
R1(config-if)# ip nhrp redirect
R1(config-if)# exit
```

on SPOKES (R2/R3/R4)

```
R2(config)# int tunnel 0
24 R2(config-if)# ip nhrp shortcut
R2(config-if)# exit
```

DMVPN Phase 3 - EIGRP ROUTING



DMVPN Phase 3 - EIGRP ROUTING

- ▶ NHRP Redirect is a special NHRP message sent by the Hub to the spoke to tell the spoke that there is a better path to the remote spoke than through the Hub.
- ▶ All it does is enforces the spoke to trigger an NHRP resolution request to IP destination.
- ▶ The "ip nhrp redirect" command should be configured on the Hub only!
- ▶ Note that we do not need "no ip next-hop-self eigrp" command in the DMVPN Phase 3.

- ▶ The only difference on the spoke is that the spoke has NHRP Shortcut configured.
- ▶ This will work together with NHRP Redirect on the Hub to send a new Resolution Request NHRP message and overwrite CEF entry to use direct spoke to spoke tunnel instead of the Hub.
- ▶ This command should be configured on spokes only

DMVPN Phase 3 - EIGRP ROUTING

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

Type escape sequence to abort.

Tracing the route to 192.168.2.2

VRF info: (vrf in name/id, vrf out name/id)

```
1 10.0.0.1 19 msec 20 msec 18 msec
```

```
2 10.0.0.2 39 msec 36 msec 34 msec
```

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

Type escape sequence to abort.

Tracing the route to 192.168.2.2

VRF info: (vrf in name/id, vrf out name/id)

```
1 10.0.0.2 19 msec 23 msec 21 msec
```

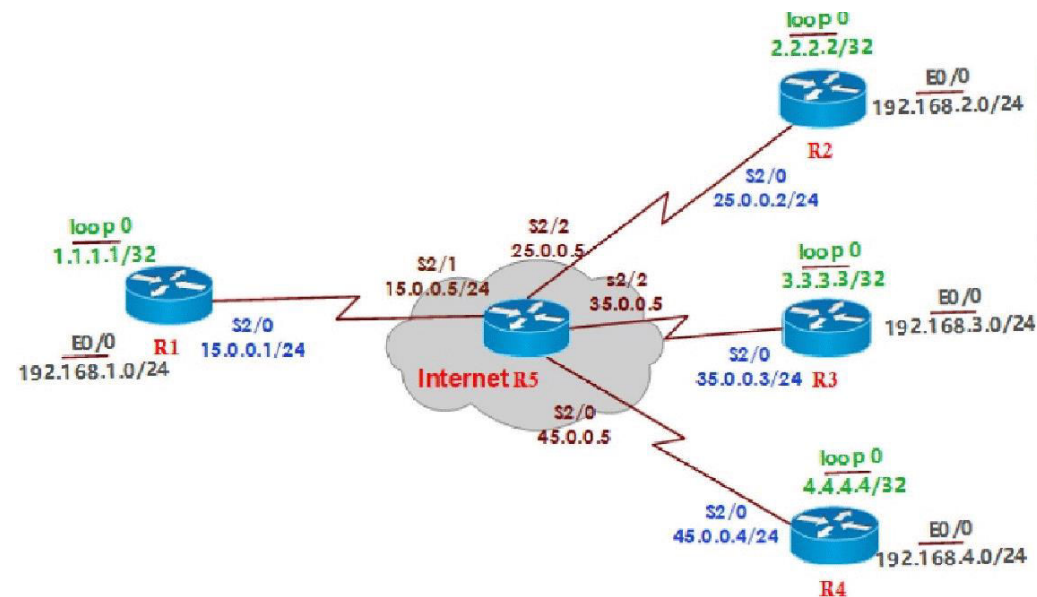
```
R4#show ip route eigrp
```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

```
D 192.168.1.0/24 [90/26905600] via 10.0.0.1, 00:00:18, Tunnel0
```

```
D 192.168.2.0/24 [90/28185600] via 10.0.0.1, 00:00:18, Tunnel0
```

```
D 192.168.3.0/24 [90/28185600] via 10.0.0.1, 00:00:18, Tunnel0
```



DMVPN Phase 3 - EIGRP ROUTING

- ▶ Next-hop to reach all other spokes routes Next hop is HUB ROUTER.
- ▶ This is because of “ip nhrp redirect” Command on Hub & “IP Nhrp shortcut” on spokes.
- ▶ They override the entries in the routing table

DMVPN Phase 3 - OSPF Routing

ON Hub & Spokes)

```
Rx(config)# int tunnel 0
```

```
Rx(config-if)# ip ospf network point-to-multipoint
```

ON HUB – R1

```
R1(config)# int tunnel 0
```

```
R1(config-if)# ip nhrp redirect
```

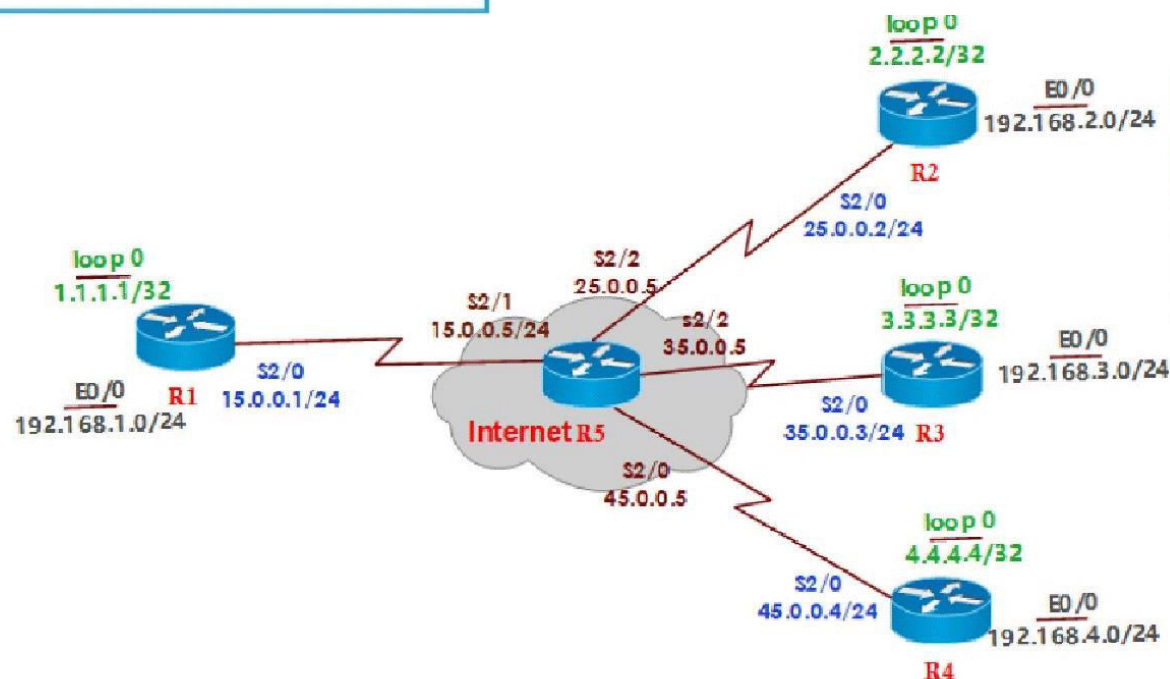
```
R1(config-if)# exit
```

on SPOKES (R2/R3/R4)

```
R2(config)# int tunnel 0
```

```
R2(config-if)# ip nhrp shortcut
```

```
R2(config-if)# exit
```



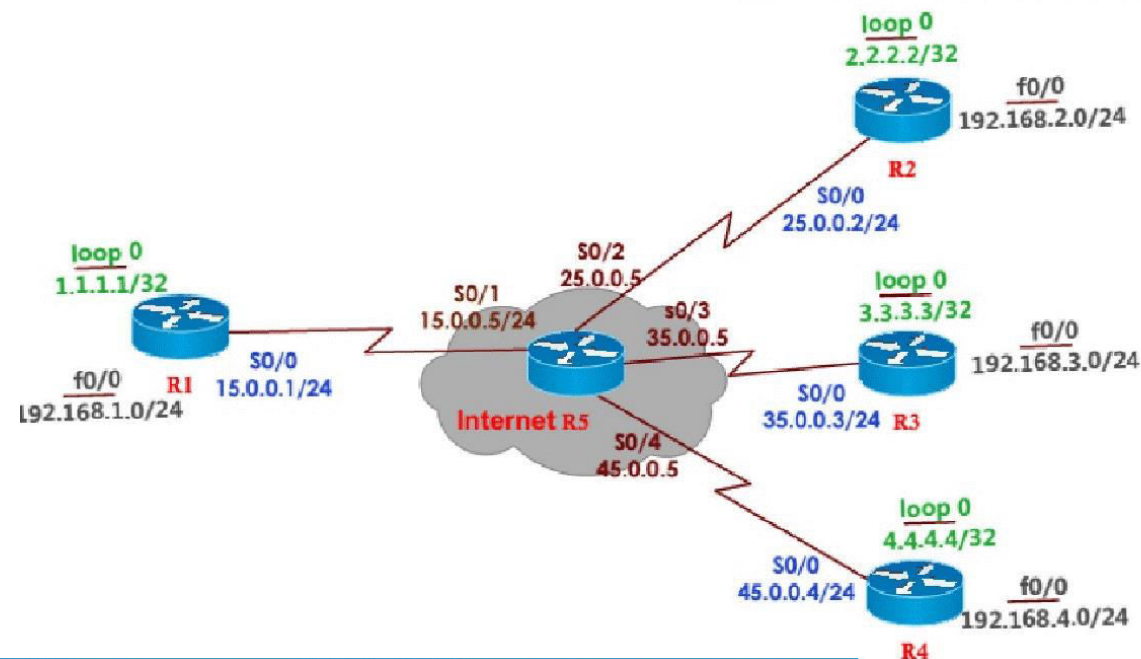
DMVPN Phase 3 - OSPF Routing

```
R2#sh ip route ospf
```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

- 10.0.0.1/32 [110/1000] via 10.0.0.1, 00:01:15, Tunnel0
- 10.0.0.3/32 [110/2000] via 10.0.0.1, 00:00:53, Tunnel0
- 10.0.0.4/32 [110/2000] via 10.0.0.1, 00:00:43, Tunnel0
- 192.168.1.0/24 [110/1010] via 10.0.0.1, 00:01:15, Tunnel0
- 192.168.3.0/24 [110/2010] via 10.0.0.1, 00:00:53, Tunnel0
- 192.168.4.0/24 [110/2010] via 10.0.0.1, 00:00:43, Tunnel0



- ▶ Next-hop to reach all other spokes routes Next hop is HUB ROUTER.
- ▶ This is because of “ip nhrp redirect” Command on Hub & “IP Nhrp shortcut” on spokes.
- ▶ They override the entries in the routing table

DMVPN Phase 3 - OSPF Routing

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

```
Type escape sequence to abort.
```

```
Tracing the route to 192.168.2.2
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
 1 10.0.0.1 19 msec 20 msec 18 msec
```

```
 2 10.0.0.2 39 msec 36 msec 34 msec
```

```
R4#traceroute 192.168.2.2 source 192.168.4.4
```

```
Type escape sequence to abort.
```

```
Tracing the route to 192.168.2.2
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
 1 10.0.0.2 19 msec 23 msec 21 msec
```



Thank You