Configuración de VPNs en capa 3.

Pablo Collado Soto

Ingeniería de Tráfico

1. Introducción

En esta práctica configuramos 2 VPNs de capa 3 (L3VPN) con varias sedes. Para ello empleamos la tecnología MPLS/LDP para soportar los túneles así como OSPF para encaminar la red del proveedor y BGP para comunicar los PEs a través de los que podemos acceder a las distintas sedes. En definitiva, todo se reduce a configurar los diferentes encaminadores de manera correcta. Se pueden consultar las configuraciones de los encaminadores aquí.

2. Paso previo: conectividad *Full-Mesh*

Antes de poner a funcionar la topología que se requiere empezamos por añadir una tercera sede a ambas VPNs. En este escenario intermedio cada VPN alcanza a las otras con un solo "salto" en el sentido de que se busca directamente el PE asociado a la VPN destino. Esto es, existe conectividad *punto a punto* entre las VPNs. La forma de conseguir esto es forzar a cada PE a importar las rutas de todos los demás en base a los RTs (*Routing Targets*) configurados. Todo se traduce a configurar el mismo RT para importar y exportar las rutas en cada PE. Eso sí, estos siguen siendo distintos para cada VPN. Las configuraciones se pueden consultar aquí.

3. Topología final: *Hub & Spoke* y *Star*

En la topología que se nos requería debemos configurar cada VPN de manera distinta. Es por eso que dedicamos una sección a cada una de ellas.

3.1. VPN1: *Hub & Spoke*

La topología *Hub & Spoke* se caracteriza por que todo el tráfico entre las sedes secundarias (PE2 y PE4) pasa por la sede central (PE1). Para lograrlo veremos cómo se establecen en el fondo 2 túneles distintos. Uno de ellos se encarga de comunicar a las sedes secundarias con la central mientras que el otro comunica ambas sedes secundarias **a través** de la central.

La clave para lograr que esta estructura lógica funcione es forzar a las sedes secundarias (*spokes*) a **solo** importar las rutas anunciadas por la sede central (*hub*) mientras que la sede central importará las rutas anunciadas por ambas sedes secundarias. Podemos lograrlo tener este esquema a través de la configuración de los RTs de manera que la sede central exporta el `RT 101:222` e importa el `RT 103:222` mientras que cada *spoke* importa el `RT 101:222` e importa el `RT 103:222`.

El último detalle del que debemos encargarnos es de que el *hub* exporte una ruta por defecto de manera que los *spokes* encaminen tráfico a cualquier otra VPN a través de él. Así logramos que todo el tráfico de la VPN1 tenga como primer destino la sede central si se origina en una sede secundaria y va destinado a la otra.

Nos gustaría mencionar a modo de curiosidad que si no incluimos la linea 114 de la configuración de PE1 solo exportaremos la ruta por defecto a cada uno de los *spokes* y no la ruta que nos lleva específicamente a la sede central desde cada sede secundaria. Esto se traduce en que en las tablas de encaminamiento de cada *spoke* solo aparece la de la sede que tienen conectada y la ruta por defecto (0.0.0.0). Así, todo paquete que que sea emitido por una sede secundaria, ya vaya a la otra sede

secundaria o a la central, llevará la misma etiqueta a nivel de `VPN`. Ya que el ejemplo que se ofrece en el guión de la práctica contiene la línea en cuestión veremos que en el fondo, y tal y como adelantábamos al inicio de la sección, en el fondo tenemos "2" túneles desde cada *spoke*, cada uno con su correspondiente etiqueta. Uno de ellos irá a la sede central y otro a la otra sede secundaria **a través** de la sede central.

3.2. `VPN2`: *Star*

En este caso la estructura lógica es idéntica a la anterior salvo por un pequeño detalle: debemos eliminar la conectividad entre las sedes secundarias. Dado que para esta `VPN` la sede central es `PE2` en vez de `PE1` solo debemos evitar que `PE2` publique una ruta por defecto como lo estaba haciendo `PE1` en la topología anterior. Esto implica que la configuración de la topología *star* es "más sencilla" en cuanto a que requiere un paso de configuración menos. Al igual que antes, ahora `PE2` debe importar los `RT`s de ambas sedes secundarias (`RT 104:222`) (`PE1` y `PE4`) y exportar el suyo propio (`RT 102:222`). Tanto `PE1` como `PE4` importaran **solo** el `RT` de `PE2` mientras que exportarán el mismo.

Con lo anterior se elimina la conectividad entre sedes secundarias tal y como se requiere.

3.3. Índice de figuras con las comprobaciones

Ya que vamos a incluir 17 figuras hemos recopilado en una pequeña lista las que se refieren a cada nodo de la red:

- `PE1`:
  - `Genéricas`: 1, 4, 5.
  - `VPN1`: 2, 6, 7.
  - `VPN2`: 3.
- `PE2`:
  - `Genéricas`: 14.
  - `VPN1`: 8, 10, 11.
  - `VPN2`: 9, 12, 13.
- `PE4`:
  - `VPN1`: 15.
  - `VPN2`: 16.
- `P3`:
  - `Genéricas`: 17.

4. Conclusiones

Tras realizar la práctica vemos cómo la configuración de `VPN`s a nivel de red es tremendamente versátil y que cambios rápidos y escuetos nos permiten alterar tremendamente la topología.

```
PE_1#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

     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
C       10.1.13.0/24 is directly connected, FastEthernet0/1
O       10.1.1.2/32 [110/3] via 10.1.13.2, 00:05:26, FastEthernet0/1
O       10.1.1.3/32 [110/2] via 10.1.13.2, 00:05:26, FastEthernet0/1
C       10.1.1.1/32 is directly connected, Loopback0
O       10.1.1.4/32 [110/3] via 10.1.13.2, 00:05:26, FastEthernet0/1
O       10.1.23.0/24 [110/2] via 10.1.13.2, 00:05:26, FastEthernet0/1
O       10.1.33.0/24 [110/2] via 10.1.13.2, 00:05:26, FastEthernet0/1
```

Figura 1: Tabla de encaminamiento genérica de PE1

```
PE_1#sh ip bgp vpnv4 vrf VPN1-1 labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65001:30 (VPN1-1)
   0.0.0.0          0.0.0.0         105/aggregate(VPN1-1)
   10.10.1.0/24     0.0.0.0         106/aggregate(VPN1-1)
   10.10.2.0/24     10.1.1.2        nolabel/205
   10.10.3.0/24     10.1.1.4        nolabel/105
```

Figura 2: Tabla de encaminamiento BGP de la VPN1 de PE1

```
PE_1#sh ip bgp vpnv4 vrf VPN2-1 labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65001:40 (VPN2-1)
   10.20.1.0/24     0.0.0.0         107/aggregate(VPN2-1)
   10.20.2.0/24     10.1.1.2        nolabel/206
```

Figura 3: Tabla de encaminamiento BGP de la VPN2 de PE1

```
PE_1#sh mpls forwarding-table
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
100    0           10.1.23.0/24      0          Fa0/1      10.1.13.2
101    0           10.1.1.3/32       0          Fa0/1      10.1.13.2
102    0           10.1.33.0/24      0          Fa0/1      10.1.13.2
103    301         10.1.1.2/32       0          Fa0/1      10.1.13.2
104    302         10.1.1.4/32       0          Fa0/1      10.1.13.2
105    Aggregate   0.0.0.0/0[V]      1652
106    Aggregate   10.10.1.0/24[V]   2280
107    Aggregate   10.20.1.0/24[V]   1056
```

Figura 4: LFIB de PE1

```
PE_1#traceroute 10.1.1.2

Type escape sequence to abort.
Tracing the route to 10.1.1.2

  1 10.1.13.2 [MPLS: Label 301 Exp 0] 8 msec 20 msec 24 msec
  2 10.1.23.1 [MPLS: Label 0 Exp 0] 36 msec *  48 msec
```

Figura 5: Traceroute a 10.1.1.2 desde PE1

```
PE_1#traceroute vrf VPN1-1 10.10.2.1

Type escape sequence to abort.
Tracing the route to 10.10.2.1

  1 10.1.13.2 [MPLS: Labels 301/205 Exp 0] 28 msec 16 msec 44 msec
  2 10.10.2.1 [MPLS: Labels 0/205 Exp 0] 40 msec *  40 msec
```

Figura 6: Traceroute a 10.10.2.1 (VPN1) desde PE1

```
PE_1#traceroute vrf VPN1-1 10.10.3.1

Type escape sequence to abort.
Tracing the route to 10.10.3.1

  1 10.1.13.2 [MPLS: Labels 302/105 Exp 0] 32 msec 20 msec 24 msec
  2 10.10.3.1 [MPLS: Label 105 Exp 0] 36 msec *  12 msec
```

Figura 7: `Traceroute` a `10.10.3.1` (`VPN1`) desde `PE1`

```
PC_2#sh ip bgp vpnv4 vrf VPN1-2 labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65001:30 (VPN1-2)
   0.0.0.0          10.1.1.1        nolabel/105
   10.10.1.0/24     10.1.1.1        nolabel/106
   10.10.2.0/24     0.0.0.0         205/aggregate(VPN1-2)
```

Figura 8: Tabla de encaminamiento `BGP` de la `VPN1` de `PE2`

```
PC_2#sh ip bgp vpnv4 vrf VPN2-2 labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65001:40 (VPN2-2)
   10.20.1.0/24     10.1.1.1        nolabel/107
   10.20.2.0/24     0.0.0.0         206/aggregate(VPN2-2)
   10.20.3.0/24     10.1.1.4        nolabel/106
```

Figura 9: Tabla de encaminamiento `BGP` de la `VPN2` de `PE2`

```
PC_2#traceroute vrf VPN1-2 10.10.1.1

Type escape sequence to abort.
Tracing the route to 10.10.1.1

  1 10.1.23.2 [MPLS: Labels 300/106 Exp 0] 36 msec 32 msec 36 msec
  2 10.10.1.1 [MPLS: Label 106 Exp 0] 24 msec *  44 msec
```

Figura 10: `Traceroute` a `10.10.1.1` (`VPN1`) desde `PE2`

```
PC_2#traceroute vrf VPN1-2 10.10.3.1

Type escape sequence to abort.
Tracing the route to 10.10.3.1

  1 10.1.23.2 [MPLS: Labels 300/105 Exp 0] 40 msec 16 msec 32 msec
  2 10.10.1.1 [MPLS: Label 105 Exp 0] 32 msec 40 msec 48 msec
  3 10.1.13.2 [MPLS: Labels 302/105 Exp 0] 52 msec 80 msec 44 msec
  4 10.10.3.1 [MPLS: Label 105 Exp 0] 68 msec *  36 msec
```

Figura 11: `Traceroute` a `10.10.3.1` (`VPN1`) desde `PE2`

```
PC_2#traceroute vrf VPN2-2 10.20.1.1

Type escape sequence to abort.
Tracing the route to 10.20.1.1

  1 10.1.23.2 [MPLS: Labels 300/107 Exp 0] 28 msec 20 msec 36 msec
  2 10.20.1.1 [MPLS: Label 107 Exp 0] 16 msec *  32 msec
```

Figura 12: `Traceroute` a `10.20.1.1` (`VPN2`) desde `PE2`

```
PC_2#traceroute vrf VPN2-2 10.20.3.1

Type escape sequence to abort.
Tracing the route to 10.20.3.1

  1 10.1.23.2 [MPLS: Labels 302/106 Exp 0] 24 msec 40 msec 32 msec
  2 10.20.3.1 [MPLS: Label 106 Exp 0] 48 msec *  44 msec
```

Figura 13: `Traceroute` a `10.20.3.1` (`VPN2`) desde `PE2`

```
PC_2#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
200    0           10.1.13.0/24     0          Fa0/1      10.1.23.2
201    300         10.1.1.1/32      0          Fa0/1      10.1.23.2
202    0           10.1.1.3/32      0          Fa0/1      10.1.23.2
203    0           10.1.33.0/24     0          Fa0/1      10.1.23.2
204    302         10.1.1.4/32      0          Fa0/1      10.1.23.2
205    Aggregate   10.10.2.0/24[V]  3316
206  _ Aggregate   10.20.2.0/24[V]  3488
```

Figura 14: LFIB de PE2

```
PE_4#sh ip bgp vpnv4 vrf VPN1-4 labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65001:30 (VPN1-4)
   0.0.0.0          10.1.1.1        nolabel/105
   10.10.1.0/24     10.1.1.1        nolabel/106
   10.10.3.0/24     0.0.0.0         105/aggregate(VPN1-4)
```

Figura 15: Tabla de encaminamiento BGP de la VPN1 de PE4

```
PE_4#sh ip bgp vpnv4 vrf VPN2-4 labels
   Network          Next Hop      In label/Out label
Route Distinguisher: 65001:40 (VPN2-4)
   10.20.2.0/24     10.1.1.2        nolabel/206
   10.20.3.0/24     0.0.0.0         106/aggregate(VPN2-4)
```

Figura 16: Tabla de encaminamiento BGP de la VPN2 de PE4

```
P_3#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
300    Pop tag     10.1.1.1/32      10350      Fa0/0      10.1.13.1
301    0           10.1.1.2/32      14731      Fa0/1      10.1.23.1
302 _  Pop tag     10.1.1.4/32      6337       Fa1/0      10.1.33.1
```

Figura 17: LFIB de P3