

Overview

No OSPF e IS-IS, o roteamento é feito com base no *Shortest Path First (SPF)*. Com isso, fluxos de diferentes origens são encaminhados pelo mesmo caminho o que pode causar congestionamento no tráfego.

A Engenharia de Tráfego ou *Traffic Engineering (TE)* previne esta situação. Cada link possui atributos que são configurados pelo operador e anunciados através do OSPF-TE. O *Constrained Shortest Path First (CSPF)* calcula o melhor caminho baseado nas restrições e atributos dos links. Contudo, é necessário um protocolo para sinalizar o túnel TE ao longo do caminho.

O RSVP foi originalmente desenhado para sinalização no modelo IntServ (QoS). Para adaptá-lo ao MPLS, vários novos objetos relacionados Engenharia de Tráfego foram definidos para permitir o transporte de *labels* e outras especificações. O RSVP usa PATH e RESV *messages* para sinalização do túnel.

Cenário de referência

A Figura 1 ilustra os elementos e informações que serão utilizadas na configuração do túnel TE.

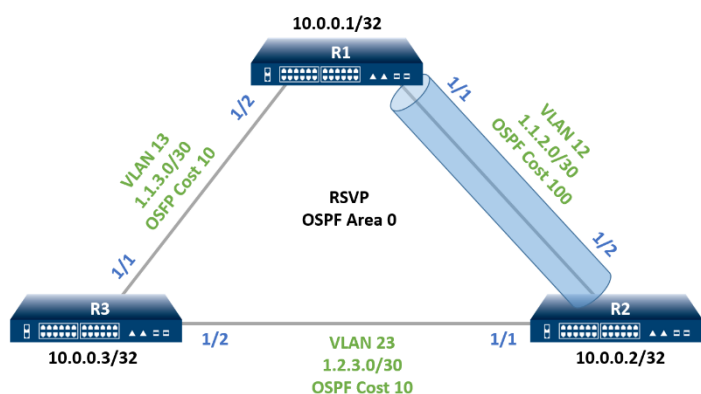


Figura 1 – Túnel TE

MODEL	FIRMWARE VERSION
DM4000, DM4100	14.4

Tabela 1 - Versão de firmware

Configurações

Inicialmente serão apresentadas as configurações mínimas de infraestrutura L3 e MPLS.

Infraestrutura IP/MPLS

O custo do OSPF no link entre R1 e R2 é maior para que o roteamento entre esses dois *routers* seja feito através de R3.

R1:

```
configure terminal
!
interface loopback 0
ip address 10.0.0.1/32
mpls enable
!
interface vlan 12
ip address 1.1.2.1/30
set-member tagged ethernet 1/1
ip ospf cost 100
!
interface vlan 13
ip address 1.1.3.2/30
set-member tagged ethernet 1/2
ip ospf cost 10
```

```
!
router ospf
router-id 10.0.0.1
network 10.0.0.1/32 area 0
network 1.1.2.0/30 area 0
network 1.1.3.0/30 area 0
```

R2:

```
configure terminal
!
interface loopback 0
ip address 10.0.0.2/32
mpls enable
!
interface vlan 12
ip address 1.1.2.2/30
set-member tagged ethernet 1/2
ip ospf cost 100
!
interface vlan 23
ip address 1.2.3.1/30
set-member tagged ethernet 1/1
ip ospf cost 10
!
router ospf
router-id 10.0.0.2
network 10.0.0.2/32 area 0
network 1.1.2.0/30 area 0
network 1.2.3.0/30 area 0
```

R3:

```
configure terminal
!
interface loopback 0
ip address 10.0.0.3/32
mpls enable
!
interface vlan 13
ip address 1.1.3.1/30
set-member tagged ethernet 1/1
ip ospf cost 10
!
interface vlan 23
ip address 1.2.3.2/30
set-member tagged ethernet 1/2
ip ospf cost 10
!
router ospf
router-id 10.0.0.3
network 10.0.0.3/32 area 0
network 1.1.3.0/30 area 0
network 1.2.3.0/30 area 0
```

RSVP

O RSVP é habilitado nas VLAN's de infraestrutura e na configuração do OSPF-TE para que seja possível distribuir as informações de atributos e restrições dos links.

R1:

```
interface vlan 12
  rsvp enable
!
interface vlan 13
  rsvp enable
!
router ospf
  mpls traffic-eng
```

R2:

```
interface vlan 12
  rsvp enable
!
interface vlan 23
  rsvp enable
!
router ospf
  mpls traffic-eng
```

R3:

```
interface vlan 13
  rsvp enable
!
interface vlan 23
  rsvp enable
!
router ospf
  mpls traffic-eng
```

Túnel TE

A criação do Túnel TE permitirá que o tráfego MPLS entre R1 e R2 seja encaminhado através do caminho de maior custo. Inicialmente são criados caminhos específicos onde são listados explicitamente todos os saltos por onde os túneis serão formados. O túnel TE é unidirecional e é criado a partir do *head end* com destino ao *tail end*.

R1:

```
mpls expl-path
explicit-path identifier 1
  tsp-hop 1 path-option 1 next-address ipv4 1.1.2.2 strict
!
mpls te
interface te-tunnel 12
  tunnel mpls destination 10.0.0.2
  tunnel mpls traffic-eng path-option 1 explicit-path identifier 1
no shutdown
```

R2:

```
mpls expl-path
explicit-path identifier 1
  tsp-hop 1 path-option 1 next-address ipv4 1.1.2.1 strict
!
mpls te
interface te-tunnel 21
  tunnel mpls destination 10.0.0.1
  tunnel mpls traffic-eng path-option 1 explicit-path identifier 1
no shutdown
```

FRR

O *Fast ReRoute (FRR)* quando habilitado, permite que um túnel *Detour* seja criado automaticamente por um caminho diferente daquele utilizado pelo túnel principal. O método *Point of Local Repair (PLR)* permite que o desvio do tráfego seja feito o mais próximo do ponto de falha. Com isso, a convergência ocorre em sub 50ms.

R1:

```
mpls te
interface te-tunnel 12
  tunnel mpls traffic-eng fast-reroute one-to-one
```

R2:

```
mpls te
interface te-tunnel 21
  tunnel mpls traffic-eng fast-reroute one-to-one
```

L2VPNorRSVP

É possível a criação de até dois túneis por caminhos diferentes entre os mesmos *head end* e *tail end*. Isso permite selecionar o caminho pelo qual uma L2VPN será estabelecida. A seleção do túnel é feita na configuração da L2VPN. Além disso, é necessário que haja uma *target session* LDP para sinalização da L2VPN.

R1:

```
interface vlan 100
  set-member tagged ethernet 1/10
!
mpls ldp neighbor 10.0.0.2
!
mpls vpws
vpn 100
  xconnect vlan 100 vc-type vlan
  neighbor 10.0.0.2 pwid 100 mplstype te-tunnel 12
no shutdown
```

R2:

```
interface vlan 100
  set-member tagged ethernet 1/10
!
mpls ldp neighbor 10.0.0.1
!
mpls vpws
vpn 100
  xconnect vlan 100 vc-type vlan
  neighbor 10.0.0.1 pwid 100 mplstype te-tunnel 21
no shutdown
```

L2VPNorLDPorRSVP

O encapsulamento de LSP's através de túneis TE (LDPorRSVP) permite o estabelecimento de L2VPN's em redes que não operam com RSVP em todos os segmentos. O parâmetro *autoroute announce* permite o anúncio dos túneis como *next hop* na tabela de roteamento do IGP.

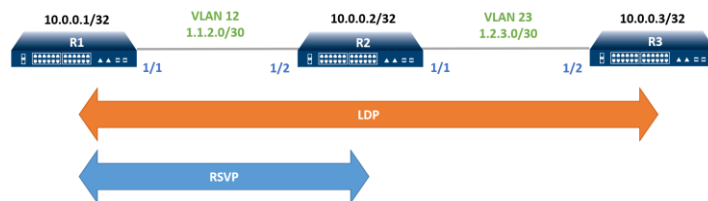


Figura 2 - Topologia LDPorRSVP

R1:

```
interface vlan 12
ip address 1.1.2.1/30
set-member tagged ethernet 1/1
rsvp enable
ldp enable
!
router ospf
router-id 10.0.0.1
network 10.0.0.1/32 area 0
network 1.1.2.0/30 area 0
mpls traffic-eng
!
mpls expl-path
explicit-path identifier 1
  tsp-hop 1 path-option 1 next-address ipv4 1.1.2.2 strict
!
mpls te
interface te-tunnel 12
  tunnel mpls destination 10.0.0.2
  tunnel mpls traffic-eng autoroute announce
  tunnel mpls traffic-eng path-option 1 explicit-path identifier 1
no shutdown
!
mpls ldp neighbor 10.0.0.3
!
interface vlan 100
  set-member tagged ethernet 1/10
!
mpls vpws
vpn 100
  xconnect vlan 100 vc-type vlan
  neighbor 10.0.0.3 pwid 100 mplstype non-te
no shutdown
```

R2:

```
interface vlan 12
ip address 1.1.2.2/30
set-member tagged ethernet 1/2
rsvp enable
ldp enable
!
interface vlan 23
ip address 1.2.3.1/30
set-member tagged ethernet 1/1
rsvp enable
ldp enable
!
router ospf
router-id 10.0.0.2
network 10.0.0.2/32 area 0
network 1.1.2.0/30 area 0
network 1.2.3.0/30 area 0
mpls traffic-eng
!
mpls expl-path
explicit-path identifier 1
  tsp-hop 1 path-option 1 next-address ipv4 1.1.2.1 strict
!
mpls te
interface te-tunnel 21
  tunnel mpls destination 10.0.0.1
  tunnel mpls traffic-eng autoroute announce
  tunnel mpls traffic-eng path-option 1 explicit-path identifier 1
no shutdown
```

R3:

```
interface vlan 23
```

```
ip address 1.2.3.2/30
set-member tagged ethernet 1/2
rsvp enable
ldp enable
!
router ospf
router-id 10.0.0.3
network 10.0.0.3/32 area 0
network 1.2.3.0/30 area 0
mpls traffic-eng
!
mpls ldp neighbor 10.0.0.1
!
interface vlan 100
set-member tagged ethernet 1/10
!
mpls vpls
vpn 100
xconnect vlan 100 vc-type vlan
neighbor 10.0.0.1 pwid 100 mplstype non-te
no shutdown
```

Verificações

A seguir são apresentados comandos para verificação do *status* dos túneis TE

- Mostra um *breaf* dos túneis que são encaminhados pelo switch:

```
show mpls te traffic-eng tunnels
```

- Mostrar os detalhes de cada túnel encaminhado através do switch:

```
show mpls te traffic-eng tunnels detail
```

- Mostrar uma tabela com o *status* e *next* hop do túnel principal e *detur* onde o switch é o *head end*:

```
show mpls rsvp-table
```

- Mostrar uma tabela consolidada com a informações sobre o status da proteção e as interfaces de uplink do túnel:

```
sho mpls frr-table
```

- Mostrar a tabela que apresenta as informações de action e labels sinalizados pelos protocolos LDP, BGP e RSVP:

```
show mpls forwarding-table
```

- Mostrar um brief do status das L2VPN

```
show mpls l2vpn
```

- Mostrar os detalhes de cada L2VPN

```
show mpls l2vpn detail
```

- Mostrar os contadores de pacotes por L2VPN

```
show mpls l2vpn counters
```