



Universidade do Minho
Escola de Engenharia

Tecnologias e Protocolos Internet

Trabalho Prático 2
MiEI - 4º Ano - 1º Semestre

A85308	Filipe Miguel Teixeira Freitas Guimarães
A85242	Maria Miguel Albuquerque Regueiras
A86271	Renata Gomes Dias Ribeiro

Braga,
24 de janeiro de 2021

Conteúdo

1	Introdução	3
2	Resolução	4
2.1	Etapa 1	4
2.2	Etapa 2	6
2.3	Etapa 3	7
2.4	Etapa 4	9
2.5	Etapa 5	10
2.6	Etapa 6	12
3	Conclusão	15

Lista de Figuras

1	Topologia da rede.	4
2	Ativação do protocolo MPLS no router R1.	5
3	Configuração das interfaces do router R1.	5
4	Comando <i>show mpls ldp discovery</i>	6
5	Comando <i>show mpls interfaces detail</i>	6
6	Configuração do protocolo OSPF no router R1.	7
7	Tabela de encaminhamento do router R1.	7
8	Esquema representativo dos túneis.	8
9	Configuração dos túneis MPLS.	9
10	Comando <i>show ip route 10.0.13.1</i>	10
11	Comando <i>show mpls forwarding</i> no router R1.	10
12	Comando <i>show mpls forwarding</i> no router R5.	11
13	Captura do tráfego do Túnel 1.	11
14	Captura do tráfego do Túnel 2.	11
15	Captura do tráfego nas interfaces do R5 que compõe o túnel. . .	12
16	Políticas para filtragem de tráfego no router R1.	13
17	Definição dos route-map no router R1.	13
18	Captura do tráfego no túnel 1.	14
19	Captura do tráfego no túnel 2.	14

1 Introdução

Neste relatório apresenta-se todo o processo de implementação do protocolo **MPLS** numa topologia específica a fim de criar túneis particulares essenciais à engenharia de tráfego.

Nesta área prevalece um conceito muito importante para os profissionais e todos que nela estão integrados: **Engenharia de Tráfego**. Esta compreende um conjunto de técnicas utilizadas para gerir tráfego em redes e telecomunicações de forma a criar um ambiente eficiente, robusto e com uma boa utilização dos recursos da rede perante o tráfego que nela circula.

No nosso caso em concreto, será criada uma topologia onde é necessário balancear o tráfego entre dois caminhos (desde uma origem e um destino estipulados) usando um protocolo conhecido como **MPLS** (Multi Protocol Label Switching).

Assim, ao longo deste relatório estão documentadas as ações realizadas para cumprir cada etapa de enunciado bem como as justificações e provas de cada decisão.

2 Resolução

Ao longo deste capítulo serão explicadas as etapas e como estas foram resolvidas através da demonstração das configurações acompanhadas pelas respectivas descrições.

2.1 Etapa 1

Identifique na sua topologia, os routers LSR (internos ao domínio MPLS) e os routers LER (routers de fronteira com interfaces dentro e interfaces fora do domínio MPLS). Configure todas as interfaces internas ao domínio MPLS para que usem IP MPLS.

Na figura 1 encontra-se a topologia criada e os respectivos routers LSR (Label Switching Router) e LER (Label Edge Router). São LSR os routers R2, R3, R4 e LER os routers R1 e R5.

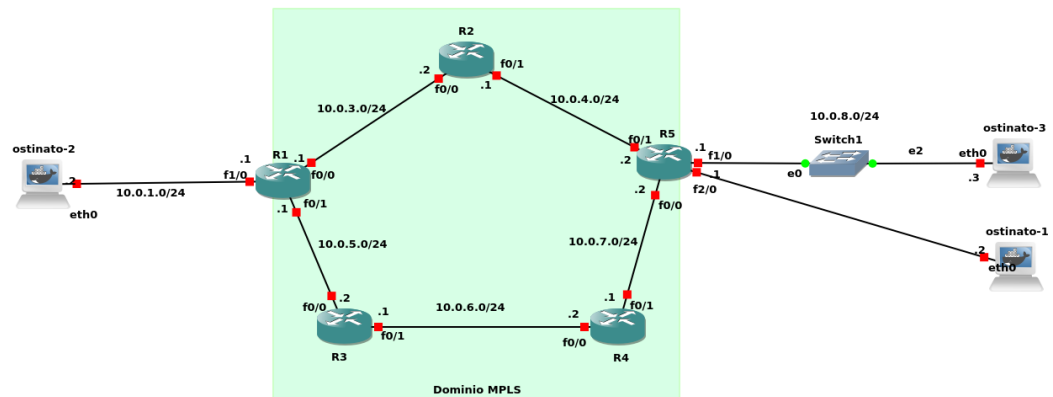


Figura 1: Topologia da rede.

Num primeiro passo foi necessário configurar os endereços IP de cada interface dos routers da topologia manualmente antes de ser possível configurar os protocolos de encaminhamento em si.

Para além disso, foi necessário configurar os routers desta para usarem o protocolo. Para tal realizaram-se os seguintes comandos (em modo config nos routers):

- ***ip cef*** para ativar o modo de envio rápido;
- ***mpls ip*** para ativar o protocolo MPLS assumindo que o tráfego que está a ser encaminhado é tráfego IP;
- ***mpls label protocol ldp*** para definir o protocolo de distribuição das etiquetas;

- ***mpls ldp router-id Loopback0*** para definir o router-id como sendo a interface de Loopback0 com uma máscara 255.255.255.255 (para garantir que nunca vai abaixo).
- ***mpls traffic-eng tunnels*** para ativar a engenharia de tráfego.

Adicionalmente, foi necessário configurar cada interface interna ao domínio MPLS correndo (em modo config-if) os comandos:

- ***mpls ip*** e ***mpls traffic-eng tunnels***;
- ***mpls mtu 1504*** para determinar a *Max Transfer Unit* (neste caso 1504).
- ***ip rsvp bandwidth 512 512*** apenas nas interfaces que iriam constituir os túneis para dizer quanta largura de banda está disponível para reserva.

A título de exemplo, encontram-se de seguida nas figuras 2 e 3 as configurações do router R1:

```
mpls ip
mpls label protocol ldp
mpls ldp router-id Loopback0
mpls traffic-eng tunnels
```

Figura 2: Ativação do protocolo MPLS no router R1.

```
interface Loopback0
ip address 10.0.9.1 255.255.255.255
!
interface FastEthernet0/0
bandwidth 512
ip address 10.0.3.1 255.255.255.0
mpls ip
mpls mtu 1504
mpls traffic-eng tunnels
ip rsvp bandwidth 512 512
!
interface FastEthernet0/1
bandwidth 512
ip address 10.0.5.1 255.255.255.0
mpls ip
mpls mtu 1504
mpls traffic-eng tunnels
ip rsvp bandwidth 512 512
!
interface FastEthernet1/0
ip address 10.0.1.1 255.255.255.0
!
interface FastEthernet2/0
ip address 10.0.2.1 255.255.255.0
```

Figura 3: Configuração das interfaces do router R1.

Podemos verificar que as interfaces estão bem configuradas assim como o protocolo LDP pelas figuras seguintes.
Na figura 4 observa-se que o protocolo LDP está ativado.

```

R1#show mpls ldp discovery
Local LDP Identifier:
10.0.9.1:0
Discovery Sources:
Interfaces:
  FastEthernet0/0 (ldp): xmit/recv
    LDP Id: 10.0.10.1:0; no route
  FastEthernet0/1 (ldp): xmit/recv
    LDP Id: 10.0.11.1:0; no route

```

Figura 4: Comando *show mpls ldp discovery*.

Na figura 5 observa-se o resultado do comando *show mpls interfaces detail*. Nesta podemos observar que as interfaces que são internas ao domínio (as que vão fazer parte dos túneis MPLS) estão bem configuradas e com o IP labeling ativado (o protocolo LDP) assim como o MTU a 1504.

```

R1#show mpls interfaces detail
Interface FastEthernet0/0:
  IP labeling enabled (ldp):
  Interface config
  LSP Tunnel labeling enabled
  BGP tagging not enabled
  Tagging operational
  Fast Switching Vectors:
    IP to MPLS Fast Switching Vector
    MPLS Turbo Vector
  MTU = 1504
Interface FastEthernet0/1:
  IP labeling enabled (ldp):
  Interface config
  LSP Tunnel labeling enabled
  BGP tagging not enabled
  Tagging operational
  Fast Switching Vectors:
    IP to MPLS Fast Switching Vector
    MPLS Turbo Vector
  MTU = 1504

```

Figura 5: Comando *show mpls interfaces detail*.

2.2 Etapa 2

Reconfigure o protocolo de estado da ligação OSPF, de modo a que passe a anunciar informação útil para a engenharia de tráfego MPLS.

Primeiramente, foi preciso ativar e configurar o protocolo de estado da ligação OSPF uma vez que o MPLS necessita de conhecer toda a topologia, utilizando para isso a ajuda deste. Escolheu-se o OSPF visto que foi abordado nas aulas e trabalhado a pormenor na prática, facilitando todo o processo.

Na figura 6 podemos observar a configuração do protocolo OSPF do router R1. Estas configurações foram replicadas para os router R1, R2, R3, R4 e R5 (adaptando-as de acordo com as networks adequadas). Para além da configuração normal deste protocolo, acrescentaram-se algumas características para que este fornecesse informações úteis ao MPLS.

Com a instrução *log-adjacency-changes* os vizinhos do router R1 serão avisados imediatamente aquando uma mudança no estado entre eles.

De seguida, ativou-se o modo de engenharia de tráfego através dos comandos *mpls traffic-eng area 0* e *mpls traffic-eng router-id Loopback0*, onde no primeiro identifica-se a área onde o router se encontra (área 0) e no segundo atribui-se o id como sendo o endereço da interface Loopback0.

É importante realçar também que a rede onde se encontra a interface Loopback0 é também anunciada.

```
router ospf 1
log-adjacency-changes
mpls traffic-eng area 0
mpls traffic-eng router-id Loopback0
network 10.0.1.0 0.0.0.255 area 0
network 10.0.2.0 0.0.0.255 area 0
network 10.0.3.0 0.0.0.255 area 0
network 10.0.5.0 0.0.0.255 area 0
network 10.0.9.0 0.0.0.255 area 0
```

Figura 6: Configuração do protocolo OSPF no router R1.

Na figura 7 podemos observar a tabela de encaminhamento do router R1.

```
10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
O    10.0.11.1/32 [110/196] via 10.0.5.2, 00:00:30, FastEthernet0/1
C    10.0.9.1/32 is directly connected, Loopback0
O    10.0.8.0/24 [110/391] via 10.0.13.1, 00:00:30, Tunnel1
      [110/391] via 10.0.13.1, 00:00:30, Tunnel2
O    10.0.13.1/32 [110/391] via 10.0.13.1, 00:00:30, Tunnel1
      [110/391] via 10.0.13.1, 00:00:30, Tunnel2
C    10.0.2.0/24 is directly connected, FastEthernet2/0
C    10.0.3.0/24 is directly connected, FastEthernet0/0
C    10.0.1.0/24 is directly connected, FastEthernet1/0
O    10.0.6.0/24 [110/390] via 10.0.5.2, 00:00:31, FastEthernet0/1
O    10.0.7.0/24 [110/585] via 10.0.13.1, 00:00:31, Tunnel1
      [110/585] via 10.0.13.1, 00:00:31, Tunnel2
      [110/585] via 10.0.5.2, 00:00:31, FastEthernet0/1
O    10.0.4.0/24 [110/390] via 10.0.3.2, 00:00:38, FastEthernet0/0
C    10.0.5.0/24 is directly connected, FastEthernet0/1
```

Figura 7: Tabela de encaminhamento do router R1.

2.3 Etapa 3

Defina um sistema final de origem e um sistema final de destino, ambos fora do domínio MPLS e estabeleça dois caminhos explícitos LSP, disjuntos, entre o LER de entrada ligado à origem e o LER de saída ligado ao destino.

Nesta etapa definimos como sistema final de origem o *host ostinato-2* e como sistema final de destino o *ehlers-ostinato-1*, ambos fora do domínio MPLS. Na figura 8 podemos observar os dois túneis definidos nesta etapa (Caminho1 e Caminho2), entre o LER de entrada ligado à origem (router R1) e o LER de saída ligado ao destino (router R5).

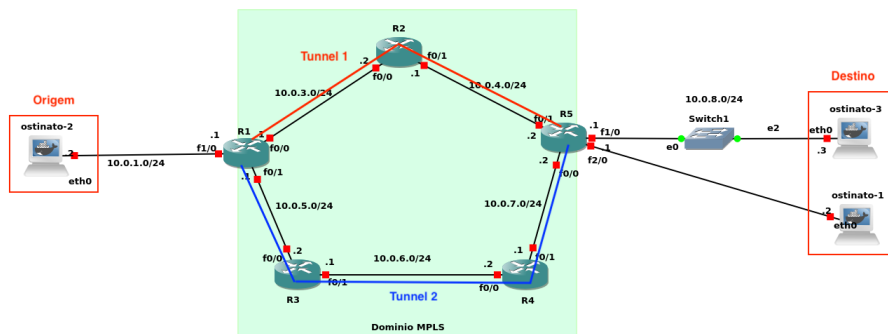


Figura 8: Esquema representativo dos túneis.

Na figura 9 estão apresentadas as configurações de ambos túneis e os respectivos caminhos. Definimos o destino ou cauda, *tunnel destination*, como o endereço de *loopback* do router R5 (10.0.13.1). Optamos por utilizar o comando ***tunnel mpls traffic-eng autoroute announce***, que indica ao router "à cabeça" (R1) para tratar o túnel como um *link* diretamente ligado à cauda (R5) dispensando assim o uso de rotas estáticas da origem para o cauda do túnel.

Para definir o túnel foi necessário primeiro construir explicitamente o caminho que este vai tomar, ou seja, indicar os endereços dos LSP que vão compor o túnel utilizando o comando ***tunnel mpls traffic-eng path-option*** e a cláusula ***ip explicit-path name***.

```

interface Tunnel1
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 10.0.13.1
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 10 explicit name Caminho1
tunnel mpls traffic-eng load-share 1
no routing dynamic
!
interface Tunnel2
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 10.0.13.1
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 20 explicit name Caminho2
tunnel mpls traffic-eng load-share 1
no routing dynamic
!
ip explicit-path name Caminho1 enable
next-address 10.0.3.2
next-address 10.0.4.2
!
ip explicit-path name Caminho2 enable
next-address 10.0.5.2
next-address 10.0.6.2
next-address 10.0.7.2

```

Figura 9: Configuração dos túneis MPLS.

2.4 Etapa 4

Force o balanceamento de tráfego MPLS entre os dois percursos LSP definidos na alínea anterior. A proporção de tráfego a enviar por cada caminho alternativo deve ser definida em percentagem a 50%.

O balanceamento de carga pode ser efetuado por pacote, no entanto, esta é uma técnica muito ineficiente e exige a reordenação dos pacotes no destino. Outro método é o balanceamento por fluxo/destino, ou seja, todos os pacotes que pertencem à mesma conexão seguem pelo mesmo túnel, evitando assim o problema anterior. Isto pode ser feito através da identificação do hash do id da conexão em causa entre os pacotes. Se todos forem iguais, então trata-se da mesma conexão.

Nesta etapa foi proposta a realização de um balanceamento de tráfego entre os dois percursos construídos na alínea anterior. A proporção de tráfego a enviar por cada caminho é de 50%, ou seja, o tráfego deve ser dividido igualmente entre os dois túneis. Para tal, na configuração dos túneis da figura 9, foi adicionado o comando **tunnel mpls traffic-eng load-share X**, em que X tomou o valor 1. Somando os $X(i)$ de todos os túneis i , temos o X_{Total} e a fracção de cada túnel é

$$\frac{X(i)}{X_{Total}}$$

Assim, tendo cada túnel o valor de 1 e um total igual a 2, cada túnel tem capacidade 1/2 da capacidade (50 %).

2.5 Etapa 5

Teste a solução de forma adequada e mostre que funciona como pretendido.

Nesta etapa serão apresentados os testes efetuados à solução apresentada e os respetivos resultados. Primeiramente, começamos por verificar que os túneis estavam de facto estabelecidos através do comando **show ip route 10.0.13.1** no router R1 sendo que este endereço refere-se ao endereço de Loopback0 do router de destino R5. Pela figura 10 podemos observar que existem duas alternativas de R1 para R5 sendo estas os túneis criados na etapa anterior.

```
R1#show ip route 10.0.13.1
Routing entry for 10.0.13.1/32
  Known via "ospf 1", distance 110, metric 391, type intra area
  Last update from 10.0.13.1 on Tunnel2, 00:00:52 ago
  Routing Descriptor Blocks:
  * 10.0.13.1, from 10.0.13.1, 00:00:52 ago, via Tunnel1
    Route metric is 391, traffic share count is 1
    10.0.13.1, from 10.0.13.1, 00:00:52 ago, via Tunnel2
    Route metric is 391, traffic share count is 1
```

Figura 10: Comando *show ip route 10.0.13.1*

Através do comando **show mpls forwarding** é possível exibir o conteúdo da MPLS Label Forwarding Information Base (LFIB). Aplicando este comando em ambos os routers de origem e destino podemos verificar ,através das figuras 11 e 12, a correta distribuição das etiquetas.

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	10.0.4.0/24	0	Fa0/0	10.0.3.2
17	Pop tag	10.0.6.0/24	0	Fa0/1	10.0.5.2
18	16	10.0.7.0/24	0	Fa0/1	10.0.5.2
19	Untagged[T]	10.0.8.0/24	0	Tu1	point2point
	Untagged[T]	10.0.8.0/24	0	Tu2	point2point
20	Pop tag	10.0.11.1/32	0	Fa0/1	10.0.5.2
21	Pop tag [T]	10.0.13.1/32	0	Tu1	point2point
	Pop tag [T]	10.0.13.1/32	0	Tu2	point2point

Figura 11: Comando *show mpls forwarding* no router R1.

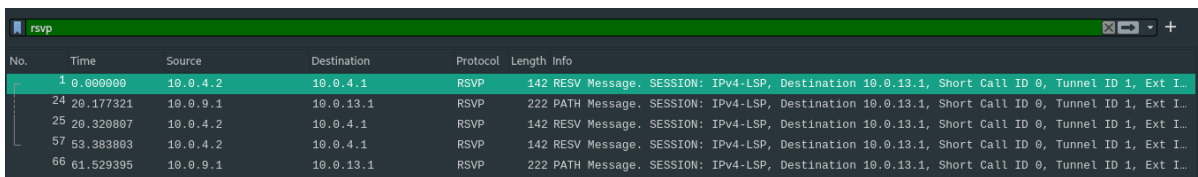
```

R5#show mpls forwarding-table
Local   Outgoing   Prefix      Bytes tag  Outgoing     Next Hop
tag     tag or VC   or Tunnel Id switched   interface
16      Pop tag     10.0.6.0/24 0          Fa0/0        10.0.7.1
17      Pop tag     10.0.12.1/32 0          Fa0/0        10.0.7.1
18      Pop tag     10.0.3.0/24 0          Fa0/1        10.0.4.1
19      20          10.0.5.0/24 0          Fa0/0        10.0.7.1
        17          10.0.5.0/24 0          Fa0/1        10.0.4.1
20      20          10.0.1.0/24 0          Fa0/1        10.0.4.1
21      22          10.0.9.1/32 0          Fa0/1        10.0.4.1
22      Pop tag     10.0.10.1/32 0          Fa0/1        10.0.4.1
23      25          10.0.11.1/32 0          Fa0/0        10.0.7.1

```

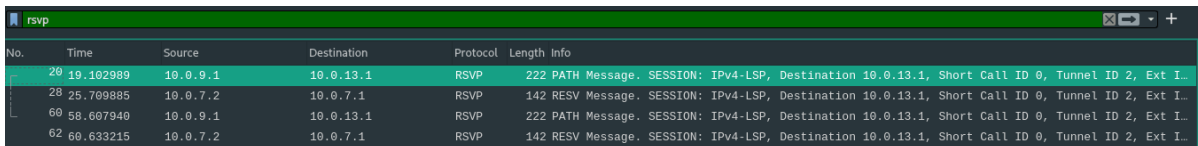
Figura 12: Comando *show mpls forwarding* no router R5.

Ainda para confirmar o correto funcionamento dos túneis implementados, utilizamos a ferramenta *Wireshark* para fazer uma captura do tráfego dos túneis em questão. Através da filtragem por protocolos da captura feita, procurando em específico o protocolo *Resource reSerVation Protocol* (RSVP), é possível observar a troca de mensagens periódica para "refrescar" o estado da ligação. As capturas estão apresentadas nas figuras 13 e 14.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.4.2	10.0.4.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 1, Ext I...
24	20.177321	10.0.9.1	10.0.13.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 1, Ext I...
25	20.320807	10.0.4.2	10.0.4.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 1, Ext I...
57	53.383803	10.0.4.2	10.0.4.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 1, Ext I...
66	61.529395	10.0.9.1	10.0.13.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 1, Ext I...

Figura 13: Captura do tráfego do Túnel 1.



No.	Time	Source	Destination	Protocol	Length	Info
20	19.102989	10.0.9.1	10.0.13.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 2, Ext I...
28	25.709885	10.0.7.2	10.0.7.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 2, Ext I...
60	58.607940	10.0.9.1	10.0.13.1	RSVP	222	PATH Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 2, Ext I...
62	60.633215	10.0.7.2	10.0.7.1	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 10.0.13.1, Short Call ID 0, Tunnel ID 2, Ext I...

Figura 14: Captura do tráfego do Túnel 2.

Para demonstrar o balanceamento de tráfego entre os dois percursos LSP definidos na alínea anterior utilizamos de novo a ferramenta *Wireshark* para capturar o tráfego dos túneis quando é feito o comando *ping* entre os *ostinato* inicial (2) e os dois *ostinatos* finais (1 e 3).

Como podemos observar na figura 15, ao efetuar o *ping*, é possível verificar que quando é enviado tráfego do *ostinato* 2 para o *ostinato* 1, este utiliza o Tunnel 1 e quando é enviado tráfego para o *ostinato* 3 utiliza o Tunnel 2. Quer isto dizer que o balanceamento está a ser definido pelo endereço de destino.

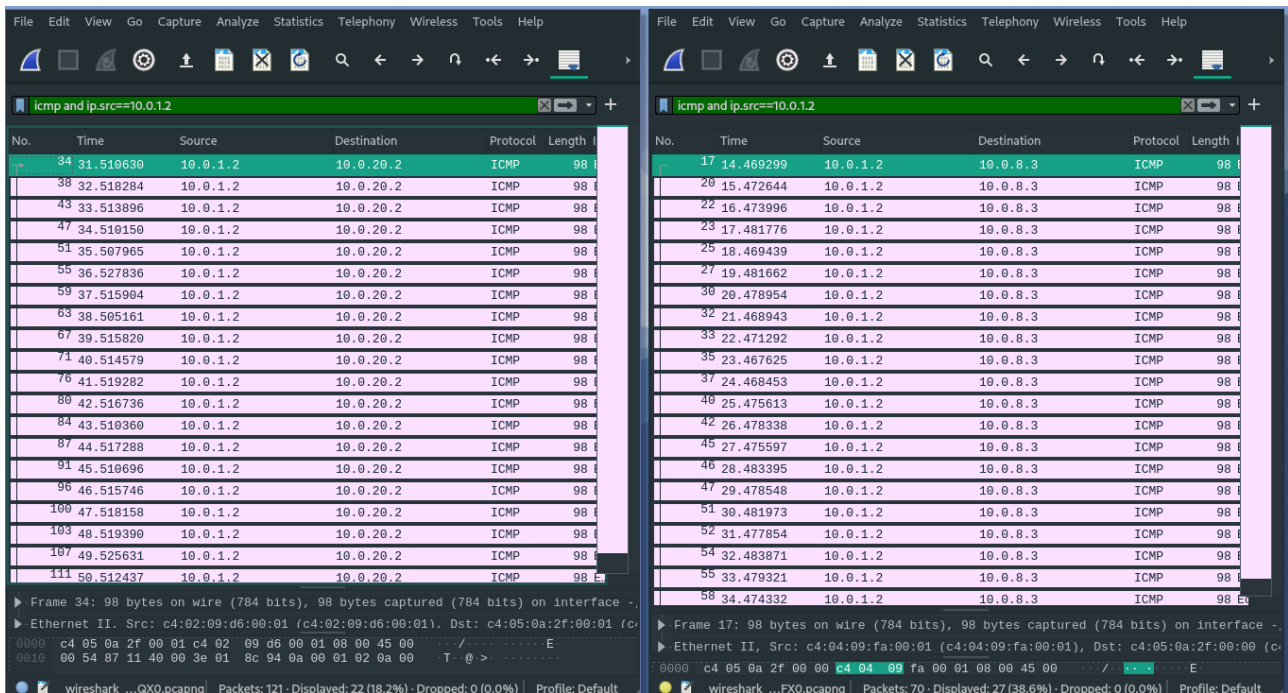


Figura 15: Captura do tráfego nas interfaces do R5 que compõe o túnel.

2.6 Etapa 6

Proponha uma nova solução de engenharia de tráfego em que o tráfego HTTP na porta 80 ou 8080 vá por um percurso e o tráfego UDP, portas 16384-3276, vá por outro alternativo; teste com auxílio de pequenos utilitários de geração de tráfego como por exemplo o iperf.

Para a realização desta etapa optou-se por utilizar hosts **ostinato** para gerar o tráfego em vez de utilitários iperf por razões de compatibilidade com o ambiente em que se desenvolveu a topologia (máquinas virtuais). Este cumpre os requisitos para a realização do que foi necessário para cumprir esta etapa final.

Primeiramente, foi necessário configurar, mais uma vez, o router R1 para indicar que o tráfego que passar por ele será filtrado e encaminhado segundo certos critérios. Dado o seu tipo, este será encaminhado ou pelo túnel 1 ou pelo túnel 2. Para isso criaram-se as políticas que se encontram na figura 16. Criaram-se duas políticas, um *route-map* para o tráfego do tipo HTTP e um para o tipo UDP, ambas na interface FastEthernet1/0, interface de entrada onde está diretamente conectado o utilitário ostinato de origem.

```

!
interface FastEthernet1/0
ip address 10.0.1.1 255.255.255.0
ip policy route-map HTTP
ip policy route-map UDP
!

```

Figura 16: Políticas para filtragem de tráfego no router R1.

Mais detalhadamente, cada uma destas políticas encontram-se definidas na figura 17.

O *route-map* com o nome HTTP refere-se à política que está responsável por gerir o que passa pela interface Tunnel1. Nesta é negado todo o tráfego do tipo UDP e permitido o do tipo TCP da porta 80 ou 8080. Por outro lado, o *route-map* com o nome UDP refere-se à política que gere o tráfego que passa pela interface Tunnel2. Aqui é negado todo o tráfego que seja do tipo TCP e permitido o do tipo UDP que seja entre as portas 16383 e 32768.

```

!
route-map HTTP
match ip address 101
set interface Tunnel1
!
access-list 101 deny udp any any
access-list 101 permit tcp any any eq 80
access-list 101 permit tcp any any eq 8080
!
route-map UDP
match ip address 102
set interface Tunnel2
!
access-list 102 deny tcp any any
access-list 102 permit udp any any gt 16383
access-list 102 permit udp any any lt 32768
!

```

Figura 17: Definição dos route-map no router R1.

Como forma de teste e para verificar se tal estava a ser cumprido segundo as políticas estabelecidas, fizeram-se capturas do tráfego através do **Wireshark** obtendo-se os resultados das figuras 18 e 19.

No.	Time	Source	Destination	Protocol	Length	Info
479	410.462442	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
480	411.471087	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
481	412.474262	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
484	413.465155	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
486	414.474389	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
487	415.458662	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
488	416.471401	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
490	417.477930	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
492	418.466929	10.0.8.3	10.0.1.2	TCP	58	80 → 65535 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Figura 18: Captura do tráfego no túnel 1.

No.	Time	Source	Destination	Protocol	Length	Info
193	162.217908	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
194	163.195013	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
197	164.210947	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
198	165.213239	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
199	166.194387	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
201	167.198506	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
204	168.206883	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
246	207.995434	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
251	208.999302	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
252	210.002913	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
253	211.000506	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
254	211.998726	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
256	213.002972	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
259	213.999588	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
260	215.000037	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18
261	215.990168	10.0.1.2	10.0.8.3	UDP	60	0 → 30000 Len=18

▶ Frame 13: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
 ▶ Ethernet II, Src: c4:04:09:fa:00:01 (c4:04:09:fa:00:01), Dst: c4:05:0a:2f:00:00 (c4:05:0a:2f:00:00)
 ▶ Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.8.3
 ▶ User Datagram Protocol, Src Port: 0, Dst Port: 30000
 ▶ Data (18 bytes)

0000 c4 05 0a 2f 00 00 c4 04 09 fa 00 01 00 00 45 00 ...
 wireshark-B6OLX0.pcapng

Packets: 261 · Displayed: 39 (14.9%) · Dropped: 0 (0.0%) · Profile: Default

Figura 19: Captura do tráfego no túnel 2.

Nestes testes realizaram-se duas capturas, uma na ligação entre o router R2 e R5 e na ligação entre o router R4 e R5. Na primeira apenas se deve verificar a passagem de tráfego do tipo HTTP (TCP) enquanto que na segunda só deve existir tráfego do tipo UDP e não de ambos em nenhum dos casos. Na primeira figura encontra-se a captura da ligação entre R2 e R5 e na segunda da ligação entre R4 e R5 e podemos observar que tal se verifica.

Este tráfego foi gerado através de duas *streams* criadas no *ostinato*. Uma das *streams* esta configurada para gerar tráfego TCP na porta 80 para simular pacotes HTTP enquanto a outra gera tráfego UDP na porta 30000.

3 Conclusão

Neste relatório abordou-se todas as etapas necessárias para criar túneis MPLS de forma a balancear tráfego entre eles, criando uma topologia equilibrada e robusta.

Com este trabalho final aprendeu-se não só como o protocolo MPLS em si funciona, mas também foram aprofundados os conhecimentos já existentes de outros aspetos (tal como configurações em si dos routers, entre outros). Desta vez utilizou-se um novo software para a simulação da rede, o GNS3, e portanto foi necessário também aprender como a ferramenta trabalhava.

Por fim, o grupo considerou este enunciado um bom exemplo de como a engenharia de tráfego se aplica em casos concretos, quais as suas vantagens e aplicações. Futuramente, poderá ser interessante aprofundar ainda mais estes conhecimentos combinando diferentes e novas técnicas para tal.