



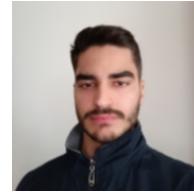
**Universidade do Minho**  
Escola de Engenharia

# Redes Fixas e Móveis

## 2023/2024

### Trabalho Prático 2

André Alves :: pg53651; Renato Gomes :: pg54174; Afonso Marques :: pg53601



#### Abstract

Nos anos 90, surgiu o MPLS (Multiprotocol Label Switching) com o objetivo inicial de melhorar a integração entre redes IP, baseadas em comutação de pacotes, e redes ATM, baseadas em comutação de circuitos. Visava criar um plano de controlo capaz de abranger tanto *routers* IP quanto *switches* ATM, enquanto dotava o IP de mecanismos de engenharia de tráfego, como restrições de largura de banda. Inicialmente, previa-se que o IP dominaria a periferia das redes e o ATM o núcleo, justificando a integração. Porém, o MPLS rapidamente assumiu o papel principal, com os ISPs oferecendo circuitos ATM e Frame Relay "sobre" MPLS. Hoje, o MPLS é dominante nas redes de acesso e essencial no núcleo da rede, oferecendo funcionalidades além da comutação rápida e engenharia de tráfego.

Uma propriedade fundamental do MPLS é a sua capacidade de estabelecer túneis, abstraindo caminhos entre extremidades da rede via comutação de etiquetas. Os túneis são sinalizados por protocolos como RSVP ou LDP e podem ser recursivos, com prioridades de estabelecimento e manutenção. Mecanismos de engenharia de tráfego permitem criar túneis com garantias de largura de banda ou restrições administrativas, calculadas por classe de serviço DiffServ.

#### Index Terms

MPLS, Cisco, EVE, Engenharia de Tráfego, Redes.

#### I. INTRODUÇÃO

O seguinte trabalho pretende explorar o ambiente EVE-NG para a criação de topologias e a familiarização com o MPLS IP e com os conceitos de encaminhamento por etiquetas. Pretende também estudar a conceção e teste de soluções de engenharia de tráfego IP MPLS simples (MPLS-TE) e de engenharia de tráfego MPLS DiffServ-TE, consolidando assim os conhecimentos do módulo teórico sobre MPLS.

A topologia utilizada para este trabalho é a seguinte:

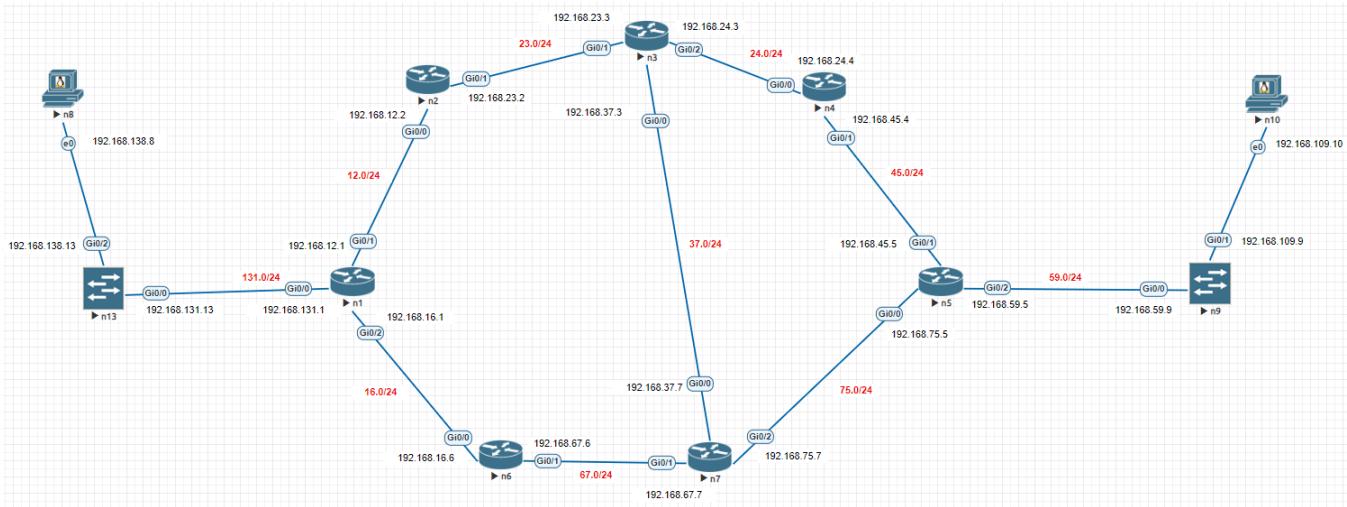


Fig. 1: Topologia utilizada

## II. DESENVOLVIMENTO

### A. Configurações Realizadas

#### 1) Descrição da topologia:

A topologia utilizada tem a forma de um duplo peixe, onde o domínio MPLS consiste no círculo de *routers* CISCO compreendido entre n1 e n7. Os *routers* LER, ou seja, os *routers* de fronteira com as interfaces dentro e interfaces fora do domínio MPLS, são o n1 e n5, enquanto que os LSR, *routers* internos ao domínio MPLS, são os restantes (n2, n3, n4, n6 e n7).

Foram determinados dois caminhos disjuntos LSP (Label Switched Paths), que são caminhos estabelecidos numa rede MPLS para encaminhar o tráfego de forma eficiente e determinística. Ambos começam em n1 e terminam em n5. O caminho LSP1 está identificado a azul e o caminho LSP2 a verde.

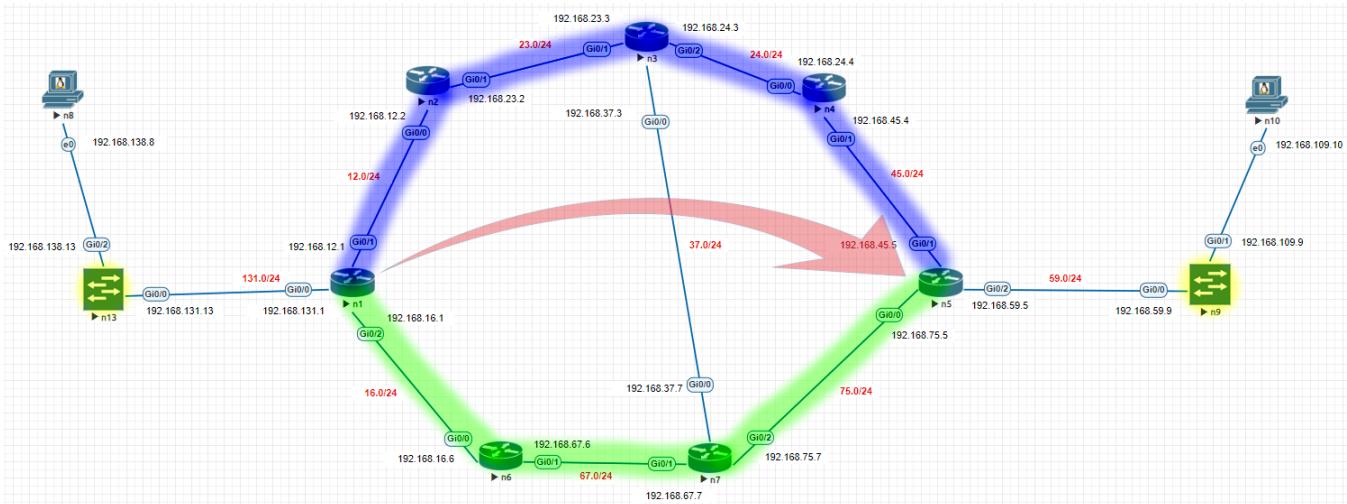


Fig. 2: Caminhos LSP

Identificado a amarelo, encontra-se o sistema final de origem que é o n13 e o sistema final de destino que é o n9, ambos fora do domínio MPLS. A seta cor-de-rosa indica o túnel MPLS "geral" estabelecido com início em n1 e fim em n5.

## 2) Configurações dos nodos:

Segue agora uma descrição mais detalhada das configurações feitas em cada nodo:

- Linux's:

- existem dois nodos Linux: n8, n10;

- foi utilizada a seguinte imagem Linux para cada um dos hosts, *linux-tinycore-6.4.tar.gz*, que permite interagir com a máquina através de uma GUI (Graphical User Interface) e personalizar mais rapidamente os endereços e gateways de cada um dos PC.

- atribuição de um IP e gateway específico de cada host, ficando n8 com IP 192.168.138.8 e gateway 192.168.138.13 e n10 com IP 192.168.109.10 e gateway 192.168.109.9, como é possível ver na descrição e imagens em baixo:

```
gn3@box:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:50:00:00:07:00
          inet addr:192.168.138.8  Bcast:192.168.138.255  Mask:255.255.255.0
          inet6 addr: fe80::205:fffe%eth0 brd fe00:0:0:0:0:0:0:1 Scope:Link
            UP BROADCAST RUNNING NOARP  MTU:1500 Metric:1
            RX packets:110 errors:0 dropped:0 overruns:0 frame:0
            TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 queueingdiscipline:1000
            RX bytes:10931 (10.6 KiB)  TX bytes:9996 (9.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536 Metric:1
            RX packets:288 errors:0 dropped:0 overruns:0 frame:0
            TX packets:288 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 queueingdiscipline:0
            RX bytes:22638 (22.1 KiB)  TX bytes:22636 (22.1 KiB)
```

Fig. 3: Configuração do n8

```
gn3@box:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:50:00:00:10:00
          inet addr:192.168.109.10  Bcast:192.168.109.255  Mask:255.255.255.0
          inet6 addr: fe80::205:fffe%eth0 brd fe00:0:0:0:0:0:0:1 Scope:Link
            UP BROADCAST MULTICAST  MTU:1500 Metric:1
            RX packets:184 errors:0 dropped:0 overruns:0 frame:0
            TX packets:184 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 queueingdiscipline:1000
            RX bytes:8157 (6.0 KiB)  TX bytes:6408 (6.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536 Metric:1
            RX packets:184 errors:0 dropped:0 overruns:0 frame:0
            TX packets:184 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 queueingdiscipline:0
            RX bytes:14432 (14.0 KiB)  TX bytes:14432 (14.0 KiB)
```

Fig. 4: Configuração do n10

- Routers do Domínio MPLS

- existem sete routers no domínio MPLS: n1, n2, n3, n4, n5, n6 e n7;

- foi utilizada a seguinte imagem para cada um dos routers, *vios-adventerprisek9-m.SPA.156-1.T*;

- para cada um destes routers, definiu-se que as suas redes seriam identificadas com endereços que utilizam o seguinte formato, XX.XX.XX.XX, onde X é o número do nodo do router; por exemplo o router n3 fica com rede 3.3.3.3.

- em cada um destes routers foi necessário estabelecer em quais das interfaces existem endereços IPv4 de MPLS juntamente com a largura de banda em cada uma. Usamos o valor de 1000000 kbps (1000 Mbps) em cada;

- para os nodos n1 e n5 foi necessário fazer o passo extra de configurar as interfaces dos dois túneis MPLS, indicando em ambos que o seu destino seria a rede 5.5.5.5 para o router n1 e 1.1.1.1 para o router n5 e as suas respectivas larguras de banda.

- realizou-se a criação dos caminhos LSP explícitos em cada nodo n1 e n5, ficando definidos da seguinte forma com o objetivo de respeitar o desenho da figura 2:

```

ip explicit-path name LSP1 enable
next-address 2.2.2.2
next-address 3.3.3.3
next-address 4.4.4.4
next-address 5.5.5.5
!
ip explicit-path name LSP2 enable
next-address 6.6.6.6
next-address 7.7.7.7
next-address 5.5.5.5
ip explicit-path name LSP1 enable
next-address 4.4.4.4
next-address 3.3.3.3
next-address 2.2.2.2
next-address 1.1.1.1
!
ip explicit-path name LSP2 enable
next-address 7.7.7.7
next-address 6.6.6.6
next-address 1.1.1.1

```

Fig. 5: LSP em n1 e n5 respectivamente

- criação de listas de acesso (ACL) que foram usadas na ultima etapa do trabalho práctico, onde o objetivo é implementar uma solução de engenharia de tráfego em que o tráfego HTTP na porta 80 ou 8080 vá por um percurso e o tráfego UDP, nas portas 16384 a 32767, vá por outro alternativo; as listas criadas (101 e 102) são definidas como:

```

n1#show access-list
Extended IP access list 101
 10 deny tcp any any eq www
 20 deny tcp any any eq 8080
 30 permit udp any any (26 matches)
 40 permit icmp any any (74 matches)
Extended IP access list 102
 10 deny udp any any range 16384 32767
 20 permit tcp any any
 30 permit icmp any any

```

Fig. 6: ACL's criados

- indicar à interface do túnel 0 que o caminho a usar é LSP1 e indicar que todo o tráfego UDP nas portas entre 16384 e 32767 passa por ali;

- indicar à interface do túnel 1 que o caminho que lhe pertence é o LSP2 e que todo o tráfego TCP nas portas 80 e 8080, deve passar por lá, o que poderá ser vantajoso visto que este caminho tem menos saltos para realizar entre n1 e n5, o que permite diminuir o número de transações TCP realizadas, que são mais custosas do que as UDP.

- Routers Destino (fora do domínio MPLS)

- existem dois routers de destino fora do domínio MPLS: n13 e n9;

- estes routers não têm qualquer tipo de configuração relacionada com o domínio MPLS; apenas se configurou os endereços de cada interface bem como a rede que existe em cada um (13.13.13.13 e 9.9.9.9 respectivamente);

- inicialmente, o plano era configurar estes dois nodos como switches (usando a imagem *viosl2-adventurek9-m-15.2.4055*), mas devido a complicações nas configurações da interface dos mesmos e por apenas serem usados dois hosts em lados opostos da topologia, optamos por alterar estes nodos para routers Cisco, mantendo o uso da imagem *vios-adventurek9-m.SPA.156-1.T*;

## B. Testes Realizados E Discussão

### 1) Teste Domínio MPLS - Ponto 7:

Tendo terminado as devidas configurações da topologia, segue-se agora o teste da conectividade da mesma, em particular do domínio estabelecido de MPLS. O objetivo é de garantir que o todo o processo foi bem executado e que é possível observar que as ligações estão ativas e que é possível estabelecer rotas entre diversos pontos da topologia.

```
nl>sh ip int br
Interface          IP-Address      OK? Method Status          Protocol
GigabitEthernet0/0 192.168.131.1   YES NVRAM  up           up
GigabitEthernet0/1 192.168.12.1    YES NVRAM  up           up
GigabitEthernet0/2 192.168.16.1    YES NVRAM  up           up
GigabitEthernet0/3 unassigned      YES unset  administratively down down
Loopback0          1.1.1.1        YES NVRAM  up           up
Tunnel0            1.1.1.1        YES TFTP   up           up
Tunnel1            1.1.1.1        YES TFTP   up           up
nl>
```

```
nl#show mpls interfaces
Interface          IP          Tunnel   BGP  Static Operational
GigabitEthernet0/1 Yes (ldp)    Yes     No   No    Yes
GigabitEthernet0/2 Yes (ldp)    Yes     No   No    Yes
Tunnel0            No          No      No   No    Yes
Tunnel1            No          No      No   No    Yes
```

Fig. 7: Interfaces do router n1

As imagens acima ilustram as interfaces do router n1, com destaque para as interfaces dos túneis, Tunnel0 e Tunnel1, indicando que estão ativas e operacionais. Em n5 o cenário é igual. A configuração para as interfaces dos túneis é a seguinte para n1 e n5 respetivamente, fazendo referência aos caminhos LSP mencionados no capítulo das configurações:

### Router n1

```
interface Tunnel0
no shutdown
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 5.5.5.5
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 1 explicit name LSP1
tunnel mpls traffic-eng load-share 1
no routing dynamic
```

```
interface Tunnel1
no shutdown
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 5.5.5.5
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 750
tunnel mpls traffic-eng path-option 1 explicit name LSP2
tunnel mpls traffic-eng load-share 1
no routing dynamic
```

### Router n5

```
interface Tunnel0
no shutdown
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 1.1.1.1
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 1 explicit name LSP1
tunnel mpls traffic-eng load-share 1
no routing dynamic
```

```

interface Tunnel1
no shutdown
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 1.1.1.1
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 1 explicit name LSP2
tunnel mpls traffic-eng load-share 1
no routing dynamic

```

De forma a comprovar que a conectividade na rede estava estabelecida, realizamos alguns testes recorrendo às ferramentas *ping* e Wireshark para confirmar que de facto a topologia está toda conectada, em particular testar se o host n8 consegue chegar ao host n10 e se o router n1 conseguia aceder à rede do router n9, fora do domínio MPLS. Usando o Wireshark na interface 1 do router n3, conseguimos verificar que os pacotes que por lá passam vem com um cabeçalho adicional relativo ao MPLS (Multi Protocol Label Switching). As imagens em baixo ilustram os resultados obtidos:

```

n1#traceroute 9.9.9.9 source 1.1.1.1
Type escape sequence to abort.
Tracing the route to 9.9.9.9
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.12.2 [MPLS: Label 25 Exp 0] 8 msec
    192.168.16.6 [MPLS: Label 26 Exp 0] 6 msec
    192.168.12.2 [MPLS: Label 25 Exp 0] 7 msec
  2 192.168.67.7 [MPLS: Label 27 Exp 0] 3 msec
    192.168.23.3 [MPLS: Label 25 Exp 0] 8 msec
    192.168.67.7 [MPLS: Label 27 Exp 0] 6 msec
  3 192.168.24.4 [MPLS: Label 24 Exp 0] 8 msec
    192.168.75.5 8 msec
    192.168.24.4 [MPLS: Label 24 Exp 0] 10 msec
  4 192.168.59.9 10 msec
    192.168.45.5 8 msec
    192.168.59.9 10 msec

```

Fig. 8: Traceroute para a rede 9.9.9.9 a partir de n1

<pre> gns3@box:~\$ ping -c5 192.168.109.10 PING 192.168.109.10 (192.168.109.10): 56 data bytes 64 bytes from 192.168.109.10: seq=0 ttl=57 time=15.416 ms 64 bytes from 192.168.109.10: seq=1 ttl=57 time=12.484 ms 64 bytes from 192.168.109.10: seq=2 ttl=57 time=12.491 ms 64 bytes from 192.168.109.10: seq=3 ttl=57 time=8.314 ms 64 bytes from 192.168.109.10: seq=4 ttl=57 time=8.993 ms  --- 192.168.109.10 ping statistics --- 5 packets transmitted, 5 packets received, 0% packet loss round-trip min/avg/max = 8.314/11.539/15.416 ms </pre>	<pre> gns3@box:~\$ ping -c5 192.168.138.8 PING 192.168.138.8 (192.168.138.8): 56 data bytes 64 bytes from 192.168.138.8: seq=0 ttl=57 time=15.651 ms 64 bytes from 192.168.138.8: seq=1 ttl=57 time=8.324 ms 64 bytes from 192.168.138.8: seq=2 ttl=57 time=19.440 ms 64 bytes from 192.168.138.8: seq=3 ttl=57 time=9.670 ms 64 bytes from 192.168.138.8: seq=4 ttl=57 time=13.262 ms  --- 192.168.138.8 ping statistics --- 5 packets transmitted, 5 packets received, 0% packet loss round-trip min/avg/max = 8.324/13.269/19.440 ms </pre>
---	--

Fig. 9: Ping de n8 (192.168.138.8) para n10 (192.168.109.10) e vice-versa

No.	Time	Source	Destination	Protocol	Length	Info
14	3.649728	192.168.138.8	192.168.109.10	ICMP	102	Echo (ping) request id=0x1f0b, seq=30/7680, ttl=62 (reply in 15)
15	3.655826	192.168.109.10	192.168.138.8	ICMP	102	Echo (ping) reply id=0x1f0b, seq=30/7680, ttl=62 (request in 14)
16	3.997598	192.168.23.3	224.0.0.2	LDP	76	Hello Message
17	4.650238	192.168.138.8	192.168.109.10	ICMP	102	Echo (ping) request id=0x1f0b, seq=31/7936, ttl=62 (reply in 18)
18	4.659324	192.168.109.10	192.168.138.8	ICMP	102	Echo (ping) reply id=0x1f0b, seq=31/7936, ttl=62 (request in 17)
19	4.997729	192.168.23.2	224.0.0.2	LDP	76	Hello Message
20	5.191107	192.168.23.2	224.0.0.5	OSPF	94	Hello Packet
21	5.651317	192.168.138.8	192.168.109.10	ICMP	102	Echo (ping) request id=0x1f0b, seq=32/8192, ttl=62 (reply in 22)
22	5.656552	192.168.109.10	192.168.138.8	ICMP	102	Echo (ping) reply id=0x1f0b, seq=32/8192, ttl=62 (request in 21)

```

> Frame 18: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0
> Ethernet II, Src: 50:00:00:0e:00:01 (50:00:00:0e:00:01), Dst: 50:00:00:0c:00:01 (50:00:00:0c:00:01)
< MultiProtocol Label Switching Header, Label: 24, Exp: 0, S: 1, TTL: 60
  0000 0000 0000 0001 1000 .... .... .... = MPLS Label: 24
  .... .... .... 000. .... .... = MPLS Experimental Bits: 0
  .... .... .... ..1 .... = MPLS Bottom Of Label Stack: 1
  .... .... .... .... 0011 1100 = MPLS TTL: 60
> Internet Protocol Version 4, Src: 192.168.109.10, Dst: 192.168.138.8
> Internet Control Message Protocol

0000 50 00 00 0c 00 01 50 00 00 0e 00 01 88 47 00 01 P....P....G...
0010 81 3c 45 00 00 54 90 e8 00 00 3e 01 73 5d c0 a8 <E-T->s...
0020 6d 0a c0 a8 8a 08 00 00 a4 5b 1f 0b 00 1f 78 3b m.....[...x;
0030 c4 3e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >.....
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Fig. 10: Pacotes ICMP capturados em n3

Observando as figuras 8, 9 e 10 em cima, que ilustram os pacotes capturados na interface 1 do router n3, conseguimos ver que da panoplia de pacotes enviados durante as operações de ping e traceroute, os pacotes contêm um cabeçalho MPLS. Relativamente aos pacotes RSVP, é possível observar qual o túnel que está a ser usado, como se pode observar nas figuras seguintes (fig.11) captadas em n3 e n7, respectivamente. O resultado é o esperado, com n3 fazendo parte do LSP1 que está associado ao túnel 0 e n7 fazendo parte do LSP2 que está associado ao túnel 1.

```

> Frame 33: 254 bytes on wire (2032 bits), 254 bytes captured (2032 bits) on interface 0
> Ethernet II, Src: 50:00:00:0c:00:01 (50:00:00:0c:00:01), Dst: 50:00:00:0e:00:01 (50:00:00:0e:00:01)
> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 5.5.5.5
< Resource Reservation Protocol (RSVP): PATH Message. SESSION: IPv4-LSP, Destination 5.5.5.5, Short Call ID 0, T
  > RSVP Header. PATH Message.
  > SESSION: IPv4-LSP, Destination 5.5.5.5, Short Call ID 0, Tunnel ID 0 Ext ID 1010101.
    HOP: IPv4, 192.168.23.2
  > TIME VALUES: 30000 ms
  > EXPLICIT ROUTE: IPv4 192.168.23.3, IPv4 192.168.24.3, IPv4 192.168.24.4, ...
  > LABEL REQUEST: Basic: L3PID: IPv4 (0x0800)
  > SESSION ATTRIBUTE: SetupPrio 7, HoldPrio 7, SE Style, [n1_t0]
  > SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 1.1.1.1, Short Call ID: 0, LSP ID: 7.
  > SENDER TSPEC: IntServ, Token Bucket, 0 bytes/sec.
  > ADSPEC
Captado em n3 na interface 1

> Frame 12: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits) on interface 0
> Ethernet II, Src: 50:00:00:0b:00:01 (50:00:00:0b:00:01), Dst: 50:00:00:0d:00:01 (50:00:00:0d:00:01)
> Internet Protocol Version 4, Src: 5.5.5.5, Dst: 1.1.1.1
< Resource Reservation Protocol (RSVP): PATH Message. SESSION: IPv4-LSP, Destination 1.1.1.1, Short Call ID 0, T
  > RSVP Header. PATH Message.
  > SESSION: IPv4-LSP, Destination 1.1.1.1, Short Call ID 0, Tunnel ID 1 Ext ID 5050505.
    HOP: IPv4, 192.168.67.7
  > TIME VALUES: 30000 ms
  > EXPLICIT ROUTE: IPv4 192.168.67.6, IPv4 192.168.16.6, IPv4 192.168.16.1,
  > LABEL REQUEST: Basic: L3PID: IPv4 (0x0800)
  > SESSION ATTRIBUTE: SetupPrio 7, HoldPrio 7, SE Style, [n5_t1]
  > SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 5.5.5.5, Short Call ID: 0, LSP ID: 15.
  > SENDER TSPEC: IntServ, Token Bucket, 0 bytes/sec.
  > ADSPEC
Captado em n7 na interface 1

```

Fig. 11: Pacotes RSVP capturados em n3 e n7

Outro comprovativo de que o MPLS está a funcionar corretamente é a tabela de encaminhamento que é gerada nos routers n1 e n5, que permitem perceber os pontos exatos onde o uso dos túneis MPLS e seus respetivos caminhos LSP se justificam.

```

n1#show mpls forwarding-table
Local      Outgoing   Prefix          Bytes Label    Outgoing   Next Hop
Label      Label      or Tunnel Id   Switched
16         No Label   13.13.13.13/32  0           Gi0/0     192.168.131.13
17         No Label   192.168.138.0/24 2842        Gi0/0     192.168.131.13
18         [T]        No Label   9.9.9.9/32   0           Tu0       point2point
19         [T]        No Label   9.9.9.9/32   0           Tu1       point2point
20         16         7.7.7.7/32   0           Gi0/2     192.168.16.6
21         [T]        Pop Label  6.6.6.6/32   0           Gi0/2     192.168.16.6
22         [T]        Pop Label  5.5.5.5/32   0           Tu0       point2point
23         20         4.4.4.4/32   0           Tu1       point2point
24         16         3.3.3.3/32   0           Gi0/1     192.168.12.2
25         [T]        Pop Label  2.2.2.2/32   0           Gi0/1     192.168.12.2
26         [T]        No Label   192.168.109.0/24 0           Tu0       point2point
27         [T]        No Label   192.168.109.0/24 0           Tu1       point2point
28         [T]        No Label   192.168.59.0/24 0           Tu0       point2point
29         [T]        No Label   192.168.59.0/24 0           Tu1       point2point
30         23         192.168.45.0/24 0           Gi0/1     192.168.12.2
31         [T]        No Label   192.168.45.0/24 0           Tu0       point2point
32         [T]        No Label   192.168.45.0/24 0           Tu1       point2point
33         18         192.168.24.0/24 0           Gi0/1     192.168.12.2
34         17         192.168.37.0/24 0           Gi0/1     192.168.12.2
35         25         192.168.37.0/24 0           Gi0/2     192.168.16.6
36         20         192.168.75.0/24 0           Gi0/2     192.168.16.6
37         Pop Label  192.168.67.0/24 0           Gi0/2     192.168.16.6
38         Pop Label  192.168.23.0/24 0           Gi0/1     192.168.12.2

[T]      Forwarding through a LSP tunnel.
View additional labelling info with the 'detail' option

```

Fig. 12: Encaminhamento em n1

```

n1#show mpls traffic-eng tunnels summary
Signalling Summary:
  LSP Tunnels Process:          running
  Passive LSP Listener:        running
  RSVP Process:                running
  Forwarding:                  enabled
  Head: 2 interfaces, 2 active signalling attempts, 2 established
        2 activations, 0 deactivations
        0 SSO recovery attempts, 0 SSO recovered
  Midpoints: 0, Tails: 2
  Periodic reoptimization:     every 3600 seconds, next in 2347 seconds
  Periodic FRR Promotion:      Not Running
  Periodic auto-bw collection: every 300 seconds, next in 247 seconds

```

Fig. 13: Informação sumariada sobre os túneis MPLS

## 2) Teste Da Nova Solução de Tráfego - Ponto 8:

O objetivo agora é de propor 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, nas portas 16384 a 3276, vá por outro alternativo.

Optamos por encaminhar o tráfego UDP pelo caminho LSP1 e o tráfego HTTP pelo caminho LSP2. Usamos o seguinte *route map*, denominado CUSTOMROUTE, para colocar essas restrições nas interfaces exteriores ao domínio MPLS, sendo elas GigabitEthernet0/0 do router n1 e GigabitEthernet0/2 do router n5.

```

n1#show route-map
route-map CUSTOMROUTE, permit, sequence 15
  Match clauses:
    ip address (access-lists): 101
  Set clauses:
    interface Tunnel0
    Policy routing matches: 100 packets, 9167 bytes
route-map CUSTOMROUTE, permit, sequence 16
  Match clauses:
    ip address (access-lists): 102
  Set clauses:
    interface Tunnel1
    Policy routing matches: 0 packets, 0 bytes

```

Fig. 14: Informação sobre o route-map criado

<pre> interface GigabitEthernet0/0 ip address 192.168.131.1 255.255.255.0 ip policy route-map CUSTOMROUTE shutdown duplex auto speed auto media-type rj45 no cdp enable </pre>	<pre> interface GigabitEthernet0/2 ip address 192.168.59.5 255.255.255.0 ip policy route-map CUSTOMROUTE shutdown duplex auto speed auto media-type rj45 no cdp enable </pre>
--	---

Fig. 15: Interfaces exteriores configuradas

Começamos por instalar o iperf3 nos dois hosts. De seguida testamos um cenário onde n8 atua como um cliente e n10 atua como o servidor. Assim, tentamos chegar a n10 com um pacote TCP na porta 8080. O objetivo é comprovar que este pacote vai pelo caminho LSP2, tal como foi definido no route-map.

<pre> gns3@box:~\$ iperf3 -c 192.168.109.10 -p 8080 [ ID] Interval      Transfer     Bandwidth      Retr </pre>	<pre> gns3@box:~\$ iperf3 -s -p 8080 [ ID] Interval      Transfer     Bandwidth      Retr </pre>
---	--

Fig. 16: Comando iperf nos hosts

Tendo duas instâncias de Wireshark a executar, uma na interface 1 do router n3 e outra na interface 1 do router n7, a sequência de pacotes TCP enviadas foi apenas capturada no router n7, como é possível verificar nas imagens em baixo:

No.	Time	Source	Destination	Protocol	Length	Info
62	56.257605	192.168.67.6	224.0.0.2	LDP	76	Hello Message
63	57.543158	192.168.67.6	224.0.0.5	OSPF	94	Hello Packet
64	58.458155	50:00:00:0b:00:01	50:00:00:0b:00:01	LOOP	60	Reply
65	58.772827	192.168.138.8	192.168.109.10	TCP	78	38745 → 8080 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TStamp=267222 TSectr=0 WS=128
66	58.780560	192.168.109.10	192.168.138.8	TCP	78	8080 → 38745 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TStamp=46419 TSectr=267222 WS=128
67	58.786845	192.168.138.8	192.168.109.10	TCP	70	38745 → 8080 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TStamp=267227 TSectr=46419
68	58.786876	192.168.138.8	192.168.109.10	TCP	107	38745 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=37 TStamp=267227 TSectr=46419 [TCP segment of a reassembled PDU]
69	59.265956	192.168.67.7	224.0.0.2	LDP	76	Hello Message
70	61.008882	192.168.67.7	224.0.0.2	LDP	76	Hello Message
71	61.267919	5.5.5.5	1.1.1.1	RSVP	238	PATH Message. SESSION: IPv4-LSP, Destination 1.1.1.1, Short Call ID 0, Tunnel ID 1, Ext ID 5050505. SENDER TEMPLATE:
72	61.599075	192.168.67.7	192.168.67.6	RSVP	142	RESV Message. SESSION: IPv4-LSP, Destination 5.5.5.5, Short Call ID 0, Tunnel ID 1, Ext ID 1010101. FILTERSPEC: IPv4
73	61.787971	50:00:00:0b:00:01	CDP/FTP/DTP/PAgP/UD...	CDP	355	Device ID: n7 Port ID: GigabitEthernet0/1
74	62.011954	192.168.67.7	224.0.0.5	OSPF	94	Hello Packet

```

> Frame 65: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
> Ethernet II, Src: 50:00:00:0d:00:01 (50:00:00:0d:00:01), Dst: 50:00:00:0b:00:01 (50:00:00:0b:00:01)
< MultiProtocol Label Switching Header, Label: 18, Exp: 0, S: 1, TTL: 61
  0000 0000 0001 0010 .... .... .... = MPLS Label: 18
  .... .... .... 000. .... .... = MPLS Experimental Bits: 0
  .... .... .... ....1 .... .... = MPLS Bottom Of Label Stack: 1
  .... .... .... .... 0011 1101 = MPLS TTL: 61
> Internet Protocol Version 4, Src: 192.168.138.8, Dst: 192.168.109.10
> Transmission Control Protocol, Src Port: 38745, Dst Port: 8080, Seq: 0, Len: 0

```

Fig. 17: Pacotes capturados durante teste de iperf

Assim conseguimos concluir que a solução de engenharia de tráfego que propomos funciona dentro do esperado, respeitando as regras definidas no enunciado.

### III. CONCLUSÃO

Por fim, consideramos que atingimos os objetivos propostos, e aprendemos outros conceitos, como configuração de routers com a sintaxe da CISCO. Infelizmente, despendemos demasiado tempo a pesquisar como implementar os conceitos aprendidos nas aulas teóricas, sendo que inicialmente foi difícil saber que comandos aplicar, e onde. Ainda assim, ficamos satisfeitos com o trabalho final.

### REFERENCES

- [1] <http://tinycorelinux.net/>
- [2] <https://www.eve-ng.net/index.php/documentation/howtos/howto-create-own-linux-host-image/>