

Engenharia de Tráfego com MPLS

Catarina Pereira, PG53733, Inês Neves, PG53864, e Leonardo Martins, PG53996

Resumo

Este trabalho refere a conectividade e o desempenho das redes de acesso IP, utilizando ferramentas como `ping`, `traceroute` e `iperf`. São realizados testes de conectividade entre o HOME-PC e o SERVER, analisando os tempos de transmissão e o RTT para diferentes tamanhos de pacotes. Modificações na conexão residencial são feitas para torná-la assimétrica, seguidas por novos testes de conectividade. A análise de desempenho é conduzida utilizando `ping` para medir a largura de banda máxima em transferências de dados em TCP (Transmission Control Protocol) e UDP (User Datagram Protocol). Os diferentes parâmetros de configuração na conexão residencial são explorados e os seus efeitos no desempenho são avaliados em ambos os protocolos.

Index Terms

Wireshark, TCP, UDP, Túneis MPLS, Plataforma de emulação EVE-NG, Configuração do router, Balanceamento de carga, Protocolo de roteamento OSPF, Mapas de rotas, Comandos Ping e Traceroute

I. INTRODUÇÃO

ESTE relatório apresenta o trabalho prático realizado na disciplina de Redes de Acesso e Núcleo, do 2º semestre do 1º ano do Mestrado em Engenharia de Telecomunicações e Informática. Desenvolvido em resposta a um desafio do docente, o trabalho concentra-se na configuração da plataforma de emulação de redes EVE-NG, utilizando as ferramentas e materiais disponíveis na plataforma BlackBoard [1, 2]. O EVE-NG é uma ferramenta versátil que suporta uma variedade de dispositivos de redes, como *routers*, *switches*, PCs Linux, firewalls, de várias marcas populares, incluindo Cisco, Juniper, Fortinet e MikroTik. A configuração envolve a ativação do mecanismo de virtualização Intel VT-x/EPT, adição de imagens de nós e instalação de imagens específicas para diferentes dispositivos, como *routers* Cisco, *switches* e *hosts* Linux. Também inclui a instalação do NETem para modelagem de parâmetros de enlace, como perda e atraso de pacotes. O relatório oferece um guia abrangente sobre como preparar e configurar o EVE-NG para emulação de redes.

II. BALANCEAMENTO DE CARGA DE FORMA DESIGUAL ENTRE DOIS TÚNEIS MPLS

COMO se pode observar na Figura 1, a topologia da rede apresentada sob a forma de (duplo) peixe, onde parte das ligações possui um capacidade *full-duplex* de 1000.0 Mbps e 2.0 Gbps. Os *routers* designados como LER são o R1 e o R5, enquanto os LSR, para o caminho superior, são os R2, R3 e R4 e os *routers* para o caminho inferior são o R6 e o R7.

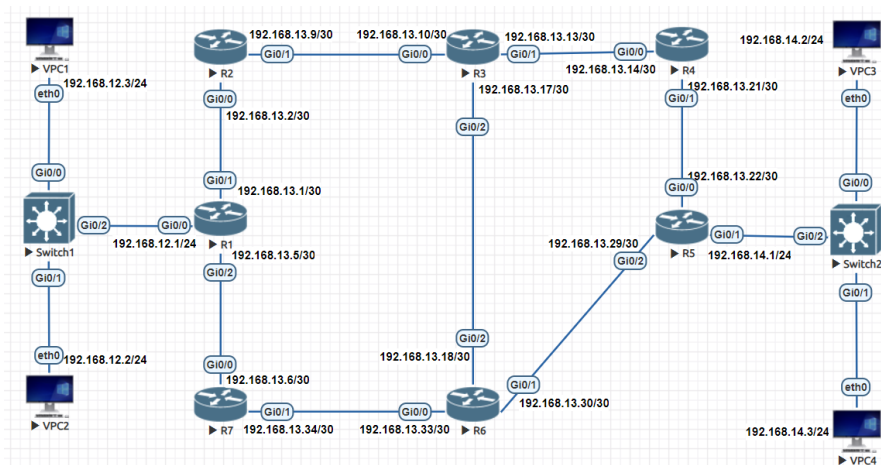


Figura 1: Topologia desenvolvida.

Na Tabela I pode ser observado os endereços IP das interfaces dos *routers*, e na Tabela II, é apresentado os endereços IP dos *hosts*.

Os estudantes são do 1º ano do Mestrado de Engenharia de Telecomunicações e Informática da Universidade do Minho.

Os endereços de email do Grupo 3 são os seguintes:

Catarina Pereira: pg537336@alunos.uminho.pt.

Inês Neves: pg53864@alunos.uminho.pt.

Leonardo Martins: pg53996@alunos.uminho.pt.

Tabela I: Endereços IP das Interfaces dos Routers.

Router	Interface G0/0	Interface G0/1	Interface G0/2	Loopback 0
R1	192.168.12.1/24	192.168.13.1/30	192.168.13.5/30	1.1.1.1
R2	192.168.13.2/30	192.168.13.9/30	Não aplicável	2.2.2.2
R3	192.168.13.10/30	192.168.13.13/30	192.168.13.17/30	3.3.3.3
R4	192.168.13.14/30	192.168.13.21/30	Não aplicável	4.4.4.4
R5	192.168.13.22/30	192.168.14.1/24	192.168.13.29/30	5.5.5.5
R6	192.168.13.33/30	192.168.13.30/30	192.168.13.18/30	6.6.6.6
R7	192.168.13.6/30	192.168.13.34/30	Não aplicável	7.7.7.7

Tabela II: Endereços IP dos Hosts.

Hosts	eth0
VPC1	192.168.12.3/24
VPC2	192.168.12.2/24
VPC3	192.168.14.2/24
VPC4	192.168.14.3/24

III. SOLUÇÃO IMPLEMENTADA

No decorrer do enunciado, foi proposta a implementação de um sistema de MPLS, para isso, altera-se as configurações dos routers da topologia apresentada na Figura 1.

Para além dos comandos utilizados no materiais disponíveis na plataforma BlackBoard [1, 2]. Também utilizou-se os seguintes comandos:

- `no passive-interface G0/x`: O comando `no passive-interface` na configuração do protocolo de roteamento Cisco reverte o efeito do comando `passive-interface`. Permite a transmissão de atualizações de roteamento numa interface previamente configurada como passiva, permitindo que atualizações de roteamento sejam enviadas e recebidas nesta interface. Este comando é usado em configurações dos routers OSPF para controlar quais interfaces participam ativamente nas atualizações do protocolo de roteamento. No caso deste projeto serão todas as interfaces que participam ativamente.
- `copy running-config startup-config`: Este comando é usado para copiar a configuração atual em execução (`running-config`) de um dispositivo Cisco para a configuração de inicialização (`startup-config`) armazenada na RAM não volátil (NVRAM) [3]. Isto garante que o dispositivo carregará a configuração especificada durante a próxima reinicialização, preservando quaisquer alterações feitas na configuração do dispositivo. Este comando é comumente usado para salvar alterações feitas na configuração do dispositivo, garantindo que elas persistam durante as reinicializações.

Na Figura 2, é possível ver a configuração do router R1 (LER). Na Figura 2 pode-se observar a criação de dois caminhos para os dois túneis entre o router 1 e o router 5, um caminho para cada túnel. O “Caminho” representa o Túnel 1, que percorrer o seguinte percurso router 1, router 2, router 3, router 4 e, finalmente, router 5. O “Caminho2” representa o Túnel 2, que percorrer o seguinte percurso router 1, router 7, router 6 e, finalmente, router 5.

```

interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface Tunnel1
 ip unnumbered Loopback0
 ip load-sharing per-packet
 tunnel mode mpls traffic-eng
 tunnel destination 5.5.5.5
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng path-option 10 explicit name Caminho
 no routing dynamic
!
interface Tunnel2
 ip unnumbered Loopback0
 ip load-sharing per-packet
 tunnel mode mpls traffic-eng
 tunnel destination 5.5.5.5
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng path-option 20 explicit name Caminho2
 no routing dynamic
!
interface GigabitEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 ip policy route-map MAP
 duplex auto
 speed auto
 media-type rj45
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/1
 ip address 192.168.13.1 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/2
 ip address 192.168.13.5 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 network 1.1.1.1 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
 network 192.168.13.0 0.0.0.3 area 0
 network 192.168.13.4 0.0.0.3 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
ip explicit-path name Caminho enable
 next-address 192.168.13.2
 next-address 192.168.13.10
 next-address 192.168.13.14
 next-address 192.168.13.22
!
ip explicit-path name Caminho2 enable
 next-address 192.168.13.6
 next-address 192.168.13.33
 next-address 192.168.13.29

```

Figura 2: Configuração do *router* R1.

Na Figura 3, é possível ver a configuração do *router* R2 (LSR).

```

interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet0/0
 ip address 192.168.13.2 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/1
 ip address 192.168.13.9 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.13.0 0.0.0.3 area 0
 network 192.168.13.8 0.0.0.3 area 0

```

Figura 3: Configuração do *router* R2.

Na Figura 4, é possível ver a configuração do router R3.

```
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface GigabitEthernet0/0
 ip address 192.168.13.10 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/1
 bandwidth 2000000
 ip address 192.168.13.13 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 2000000
!
interface GigabitEthernet0/2
 ip address 192.168.13.17 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000

interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 network 3.3.3.3 0.0.0.0 area 0
 network 192.168.12.8 0.0.0.3 area 0
 network 192.168.12.12 0.0.0.3 area 0
 network 192.168.12.16 0.0.0.3 area 0
 network 192.168.13.8 0.0.0.3 area 0
 network 192.168.13.12 0.0.0.3 area 0
 network 192.168.13.16 0.0.0.3 area 0
```

Figura 4: Configuração do *router* R3.

Na Figura 5, é possível ver a configuração do router R4.

```
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!
interface GigabitEthernet0/0
 bandwidth 2000000
 ip address 192.168.13.14 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 2000000
!
interface GigabitEthernet0/1
 bandwidth 2000000
 ip address 192.168.13.21 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 2000000
!

interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
 ip rsvp bandwidth 2000000
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 network 4.4.4.4 0.0.0.0 area 0
 network 192.168.13.12 0.0.0.3 area 0
 network 192.168.13.20 0.0.0.3 area 0
```

Figura 5: Configuração do *router* R4.

Na Figura 6, é possível ver a configuração do router R5.

```

interface Loopback0
 ip address 5.5.5.5 255.255.255.255
!
interface GigabitEthernet0/0
 bandwidth 2000000
 ip address 192.168.13.22 255.255.255.252
 ip load-sharing per-packet
 ip ospf cost 1
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 2000000
!
interface GigabitEthernet0/1
 ip address 192.168.14.1 255.255.255.0
 ip load-sharing per-packet
 ip policy route-map MAP
 duplex auto
 speed auto
 media-type rj45
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/2
 bandwidth 2000000
 ip address 192.168.13.29 255.255.255.252
 ip load-sharing per-packet
 ip ospf cost 2
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 2000000
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 network 5.5.5.5 0.0.0.0 area 0
 network 192.168.13.20 0.0.0.3 area 0
 network 192.168.13.28 0.0.0.3 area 0
 network 192.168.14.0 0.0.0.255 area 0

```

Figura 6: Configuração do *router* R5.

Na Figura 7, é possível ver a configuração do router R6.

```

interface Loopback0
 ip address 6.6.6.6 255.255.255.255
!
interface GigabitEthernet0/0
 ip address 192.168.13.33 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/1
 bandwidth 2000000
 ip address 192.168.13.30 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 2000000
!
interface GigabitEthernet0/2
 ip address 192.168.13.18 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 network 6.6.6.6 0.0.0.0 area 0
 network 19.168.13.12 0.0.0.3 area 0
 network 19.168.13.28 0.0.0.3 area 0
 network 19.168.13.32 0.0.0.3 area 0
 network 192.168.13.16 0.0.0.3 area 0
 network 192.168.13.28 0.0.0.3 area 0
 network 192.168.13.32 0.0.0.3 area 0

```

Figura 7: Configuração do *router* R6.

Na Figura 8, é possível ver a configuração do router R7.

```
interface Loopback0
ip address 7.7.7.7 255.255.255.255
!
interface GigabitEthernet0/0
ip address 192.168.13.6 255.255.255.252
duplex auto
speed auto
media-type rj45
mpls traffic-eng tunnels
mpls ip
ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/1
ip address 192.168.13.34 255.255.255.252
duplex auto
speed auto
media-type rj45
mpls traffic-eng tunnels
mpls ip
ip rsvp bandwidth 1000000
!
interface GigabitEthernet0/2
no ip address
shutdown
duplex auto
speed auto
media-type rj45

interface GigabitEthernet0/3
no ip address
shutdown
duplex auto
speed auto
media-type rj45
!
router ospf 1
mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
network 7.7.7.7 0.0.0.0 area 0
network 192.168.13.4 0.0.0.3 area 0
network 192.168.13.32 0.0.0.3 area 0
```

Figura 8: Configuração do *router* R7.

Na Figura 9 é usado o comando `traceroute` para confirmar que os túneis configurados em R1 e R5 que estão a funcionar devidamente. O endereço IP 192.168.13.22 utilizado é o endereço das interface G0/0 do router R5.

```
R1#traceroute 192.168.13.22
Type escape sequence to abort.
Tracing the route to 192.168.13.22
VRF info: (vrf in name/id, vrf out name/id)
 1 192.168.13.2 [MPLS: Label 30 Exp 0] 99 msec
   192.168.13.6 [MPLS: Label 19 Exp 0] 69 msec
   192.168.13.2 [MPLS: Label 28 Exp 0] 44 msec
 2 192.168.13.10 [MPLS: Label 29 Exp 0] 111 msec
   192.168.13.33 [MPLS: Label 29 Exp 0] 79 msec
   192.168.13.10 [MPLS: Label 28 Exp 0] 82 msec
 3 192.168.13.14 [MPLS: Label 30 Exp 0] 142 msec
   192.168.13.29 96 msec
   192.168.13.14 62 msec
```

Figura 9: Comando *traceroute*.

Na figura 10 é usado o endereço IP do *Loopback0* do *router* R1 e ao usar este endereço não se está a escolher em específico o caminho de cima ou de baixo.

Através do `traceroute` é possível verificar que está a usar os dois túneis até encontrar o destino.

```

R1#traceroute 5.5.5.5
Type escape sequence to abort.
Tracing the route to 5.5.5.5
VRF info: (vrf in name/id, vrf out name/id)
 1 192.168.13.2 [MPLS: Label 30 Exp 0] 43 msec
   192.168.13.6 [MPLS: Label 19 Exp 0] 149 msec
   192.168.13.2 [MPLS: Label 30 Exp 0] 42 msec
 2 192.168.13.33 [MPLS: Label 29 Exp 0] 46 msec
   192.168.13.10 [MPLS: Label 29 Exp 0] 99 msec
   192.168.13.33 [MPLS: Label 29 Exp 0] 52 msec
 3 192.168.13.14 [MPLS: Label 30 Exp 0] 67 msec
   192.168.13.29 54 msec
   192.168.13.14 [MPLS: Label 30 Exp 0] 85 msec

```

Figura 10: Comando *traceroute* como o IP do *Loopback0*.

IV. BALANCEAMENTO 50/50

De modo a testar a conexão, para um balanceamento de 50% para cada túnel, realizam-se testes utilizando o comando *ping* e para a visualização do *wireshark* pressiona-se com o botão direito o router 1 e abre-se o *capture* e escolhe-se as interfaces dos túneis que estão a ser percorridas para o *tunnel1* e *tunnel2*, ou seja, G0/1 e G0/2, respetivamente.

Na Figura 11 apresenta o envio de dez pacotes do VPC1 (192.168.12.3) para VPC3 (192.168.14.2).

```

VPCS> ping 192.168.14.2 -c 10

84 bytes from 192.168.14.2 icmp_seq=1 ttl=60 time=164.599 ms
84 bytes from 192.168.14.2 icmp_seq=2 ttl=59 time=118.124 ms
84 bytes from 192.168.14.2 icmp_seq=3 ttl=60 time=91.153 ms
84 bytes from 192.168.14.2 icmp_seq=4 ttl=59 time=91.092 ms
84 bytes from 192.168.14.2 icmp_seq=5 ttl=60 time=148.219 ms
84 bytes from 192.168.14.2 icmp_seq=6 ttl=59 time=140.292 ms
84 bytes from 192.168.14.2 icmp_seq=7 ttl=60 time=75.805 ms
84 bytes from 192.168.14.2 icmp_seq=8 ttl=59 time=109.184 ms
84 bytes from 192.168.14.2 icmp_seq=9 ttl=60 time=94.102 ms
84 bytes from 192.168.14.2 icmp_seq=10 ttl=59 time=127.074 ms

```

Figura 11: Ping do VPC1 para o VPC3.

Na Figura 12 e na Figura 13 observa-se que para o caminho percurso router 1, router 2, router 3, router 4 e, finalmente, router 5 (*tunnel1*) foram cinco pacotes e para o caminho de percurso router 1, router 7, router 6 e, finalmente, router 5 (*tunnel2*) os outros cinco pacotes.

No.	Time	Source	Destination	Protocol	Length	Info
127	142.552352	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xe58a, seq=2/512, ttl=63 (reply in 128)
128	142.642424	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xe58a, seq=2/512, ttl=60 (request in 127)
129	143.493036	50:00:00:00:00:01	50:00:00:00:00:01	LOOP	60	Reply
130	144.006668	192.168.13.1	224.0.0.5	OSPF	94	Hello Packet
131	144.756708	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xe88a, seq=4/1024, ttl=63 (reply in 133)
132	144.794703	192.168.13.2	224.0.0.2	LDP	76	Hello Message
133	144.808899	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xe88a, seq=4/1024, ttl=60 (request in 131)
134	144.910096	2.2.2.2	1.1.1.1	LDP	72	Keep Alive Message
135	145.025860	50:00:00:00:00:00	50:00:00:00:00:00	LOOP	60	Reply
136	145.183892	1.1.1.1	2.2.2.2	TCP	60	646 → 45542 [ACK] Seq=55 Ack=55 Win=3912 Len=0
137	146.999168	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xea8a, seq=6/1536, ttl=63 (reply in 138)
138	147.106117	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xea8a, seq=6/1536, ttl=60 (request in 137)
139	147.224285	192.168.13.1	224.0.0.2	LDP	76	Hello Message
140	149.222125	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xec8a, seq=8/2048, ttl=63 (reply in 141)
141	149.300851	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xec8a, seq=8/2048, ttl=60 (request in 140)
142	150.756566	192.168.13.2	224.0.0.2	LDP	76	Hello Message
143	151.414154	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xee8a, seq=10/2560, ttl=63 (reply in 144)
144	151.523173	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xee8a, seq=10/2560, ttl=60 (request in 143)
145	152.039885	1.1.1.1	5.5.5.5	RSVP	270	PATH Message. SESSION: IPv4-LSP, Destination 5.5.5.5, Short Call ID 0, Tunnel ID 1, Ext ID 1010101. SENDER TEMPLATE: IPv4-LSP, Tu...

> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 > Ethernet II, Src: 50:00:00:00:00:00 (50:00:00:00:00:00), Dst: 50:00:00:00:00:00 (50:00:00:00:00:00)
 > Configuration Test Protocol (loopback)
 > Data (40 bytes)

```

0000  50 00 00 0c 00 00 50 00  00 0c 00 00 90 00 00 00  P....P: .....
0010  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....

```

Figura 12: *Wireshark* pelo caminho de *tunnel1*.

No.	Time	Source	Destination	Protocol	Length	Info
99	113.900557	50:00:00:08:00:00	50:00:00:08:00:00	LOOP	60	Reply
100	114.411037	192.168.13.6	224.0.0.2	LDP	76	Hello Message
101	114.495330	192.168.13.5	224.0.0.2	LDP	76	Hello Message
102	114.795065	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xe48a, seq=1/256, ttl=63 (reply in 103)
103	114.914774	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xe48a, seq=1/256, ttl=61 (request in 102)
104	116.909208	50:00:00:0d:00:02	50:00:00:0d:00:02	LOOP	60	Reply
105	117.067375	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xe68a, seq=3/768, ttl=63 (reply in 106)
106	117.134409	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xe68a, seq=3/768, ttl=61 (request in 105)
107	119.274575	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xe98a, seq=5/1280, ttl=63 (reply in 108)
108	119.368918	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xe98a, seq=5/1280, ttl=61 (request in 107)
109	119.838508	192.168.13.6	224.0.0.2	LDP	76	Hello Message
110	119.913382	192.168.13.5	224.0.0.2	LDP	76	Hello Message
111	119.953001	192.168.13.5	224.0.0.5	OSPF	94	Hello Packet
112	121.543752	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xeb8a, seq=7/1792, ttl=63 (reply in 113)
113	121.590833	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xeb8a, seq=7/1792, ttl=61 (request in 112)
114	123.734007	192.168.12.3	192.168.14.2	ICMP	102	Echo (ping) request id=0xed8a, seq=9/2304, ttl=63 (reply in 115)
115	123.805180	192.168.14.2	192.168.12.3	ICMP	98	Echo (ping) reply id=0xed8a, seq=9/2304, ttl=61 (request in 114)
116	124.405446	192.168.13.6	224.0.0.5	OSPF	94	Hello Packet
117	125.148368	192.168.13.6	224.0.0.2	LDP	76	Hello Message
118	125.585217	192.168.13.5	224.0.0.2	LDP	76	Hello Message

> Frame 40: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
 > Ethernet II, Src: 50:00:00:08:00:00 (50:00:00:08:00:00), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
 > Internet Protocol Version 4, Src: 192.168.13.6, Dst: 224.0.0.5
 > Open Shortest Path First

```

0000  01 00 5e 00 00 05 50 00 08 00 00 08 00 45 c0  ..^...P...E-
0010  00 50 10 e4 00 00 01 59 f9 fd c0 a8 0d 06 e0 00  -P....Y.....
0020  00 05 02 01 00 30 07 07 07 07 00 00 00 40 32  ....0...@2
0030  00 00 00 00 00 00 00 00 00 00 ff ff fc 00 0a  ....
0040  12 01 00 00 00 28 c0 a8 0d 05 c0 a8 0d 06 01 01  ....(-.....
0050  01 01 ff f6 00 03 00 01 00 04 00 00 00 01  ....

```

Figura 13: Wireshark pelo caminho de tunnel2.

V. SEPARAÇÃO DE HTTP DE UDP

COM o intuito de segregar o tráfego HTTP do UDP estabelece-se umas listas de acesso, que são depois utilizadas pelos *route-maps*, para impor estas novas regras.

No router R1 foram configurados os *route-maps* e como se observa na Figura 14 a configuração do *route-map* para HTTP foi feita para o caminho de cima (tunnel1) e para UDP no caminho de baixo (tunnel2).

```

ip access-list extended HTTP
 permit tcp any any eq www
 permit tcp any any eq 8080
ip access-list extended UDP
 permit udp any any range 16384 32767
 deny    tcp any any eq www
 deny    tcp any any eq 8080
!
!
route-map MAP permit 10
 match ip address HTTP
 set interface Tunnel1
!
route-map MAP permit 20
 match ip address UDP
 set interface Tunnel2

```

Figura 14: Configuração do *route-map* para HTTP e UDP.

O comando `ping 192.168.14.2 -3 -p 8080` é usado para enviar pacotes através da porta 8080 e para o endereço IP de destino 192.168.14.2. A opção “-3” especifica que o ping deve usar TCP em vez de ICMP.

A saída mostra o *status* de cada pacote enviado, incluindo o tempo necessário para estabelecer a conexão, enviar os dados e encerrar a conexão. A saída também inclui o número de sequência, Time To Live (TTL) e o tempo gasto para cada pacote.


```
VPCS> ping 192.168.14.2 -3 -p 8080

Connect 8080@192.168.14.2 seq=1 ttl=59 time=37.487 ms
SendData 8080@192.168.14.2 seq=1 ttl=60 time=29.866 ms
Close 8080@192.168.14.2 seq=1 ttl=60 time=61.303 ms
Connect 8080@192.168.14.2 seq=2 ttl=59 time=48.314 ms
SendData 8080@192.168.14.2 seq=2 ttl=60 time=42.590 ms
Close 8080@192.168.14.2 seq=2 ttl=60 time=54.242 ms
Connect 8080@192.168.14.2 seq=3 ttl=59 time=32.008 ms
SendData 8080@192.168.14.2 seq=3 ttl=60 time=28.285 ms
Close 8080@192.168.14.2 seq=3 ttl=60 time=53.603 ms
Connect 8080@192.168.14.2 seq=4 ttl=59 time=31.290 ms
SendData 8080@192.168.14.2 seq=4 ttl=60 time=17.201 ms
Close 8080@192.168.14.2 seq=4 ttl=60 time=43.721 ms
Connect 8080@192.168.14.2 seq=5 ttl=59 time=24.621 ms
SendData 8080@192.168.14.2 seq=5 ttl=60 time=22.241 ms
Close 8080@192.168.14.2 seq=5 ttl=60 time=15.613 ms
```

Figura 15: Ping da porta TCP.

Na Figura 16 representa o wireshark os vários pacotes de TCP entre o router R1 e R5 pelo caminho superior da topologia.

1838	2121.386698	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSval=1713292720 TSecr=0
1839	2121.411031	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [ACK] Seq=1 Ack=58 Win=2920 Len=0
1840	2121.445290	192.168.12.3	192.168.14.2	TCP	70 [TCP ACKed unseen segment] 42160 → 8080 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSval=1713292720 TSecr=0
1841	2121.963983	192.168.13.1	224.0.0.2	LDP	76 Hello Message
1842	2122.264296	50:00:00:0c:00:00	50:00:00:0c:00:00	LOOP	60 Reply
1843	2122.451948	192.168.12.3	192.168.14.2	TCP	78 [TCP Port numbers reused] 42160 → 8080 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSval=1713292721 TSecr=0 WS=2
1844	2122.483823	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
1845	2122.500448	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=1713292721 TSecr=0
1846	2122.538769	192.168.12.3	192.168.14.2	TCP	126 42160 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=56 TSval=1713292721 TSecr=0
1847	2122.614489	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSval=1713292721 TSecr=0
1848	2122.637921	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [ACK] Seq=1 Ack=58 Win=2920 Len=0
1849	2122.665756	192.168.12.3	192.168.14.2	TCP	70 [TCP ACKed unseen segment] 42160 → 8080 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSval=1713292721 TSecr=0
1850	2123.667817	192.168.12.3	192.168.14.2	TCP	78 [TCP Port numbers reused] 42160 → 8080 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSval=1713292722 TSecr=0 WS=2
1851	2123.687274	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
1852	2123.692969	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=1713292722 TSecr=0
1853	2123.724724	192.168.12.3	192.168.14.2	TCP	126 42160 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=56 TSval=1713292722 TSecr=0
1854	2123.776671	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSval=1713292722 TSecr=0
1855	2123.802855	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [ACK] Seq=1 Ack=58 Win=2920 Len=0
1856	2123.828766	192.168.12.3	192.168.14.2	TCP	70 [TCP ACKed unseen segment] 42160 → 8080 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSval=1713292722 TSecr=0
1857	2124.419878	192.168.13.1	224.0.0.5	OSPF	94 Hello Packet
1858	2124.829672	192.168.12.3	192.168.14.2	TCP	78 [TCP Port numbers reused] 42160 → 8080 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSval=1713292723 TSecr=0 WS=2
1859	2124.853007	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
1860	2124.860037	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=1713292723 TSecr=0
1861	2124.880497	192.168.12.3	192.168.14.2	TCP	126 42160 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=56 TSval=1713292724 TSecr=0
1862	2124.914415	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSval=1713292724 TSecr=0
1863	2124.937370	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [ACK] Seq=1 Ack=58 Win=2920 Len=0
1864	2124.960927	192.168.12.3	192.168.14.2	TCP	70 [TCP ACKed unseen segment] 42160 → 8080 [ACK] Seq=58 Ack=2 Win=2920 Len=0 TSval=1713292724 TSecr=0
1865	2125.957565	192.168.12.3	192.168.14.2	TCP	78 [TCP Port numbers reused] 42160 → 8080 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 TSval=1713292725 TSecr=0 WS=2
1866	2125.975470	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0
1867	2125.984097	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=1713292725 TSecr=0
1868	2125.999816	192.168.12.3	192.168.14.2	TCP	126 42160 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=56 TSval=1713292725 TSecr=0
1869	2126.032795	192.168.12.3	192.168.14.2	TCP	70 42160 → 8080 [FIN, PSH, ACK] Seq=57 Ack=1 Win=2920 Len=0 TSval=1713292725 TSecr=0
1870	2126.043589	192.168.14.2	192.168.12.3	TCP	54 8080 → 42160 [ACK] Seq=1 Ack=58 Win=2920 Len=0
1871	2126.048834	192.168.12.3	192.168.14.2	TCP	70 [TCP Keep-Alive] [TCP ACKed unseen segment] 42160 → 8080 [ACK] Seq=57 Ack=2 Win=2920 Len=0 TSval=1713292725 TSecr=0

Figura 16: Wireshark do tunnel1.

O comando `ping 192.168.14.2 -2 -p 17000` é usado para enviar pacotes UDP para o endereço IP de destino 192.168.14.2, Figura 17. Este comando é usado para testar a conectividade e o desempenho da rede e pode fornecer informações sobre a capacidade da rede de lidar com pacotes grandes e o atraso, instabilidade e perda de pacotes do caminho.

Na saída fornecida, o comando `ping` foi bem-sucedido e os tempos de resposta de cada pacote são exibidos. Isto indica que a rede entre a origem e o destino está a funcionar corretamente e que o *host* de destino está acessível e a funcionar.

```
VPCS> ping 192.168.14.2 -2 -p 17000

84 bytes from 192.168.14.2 udp_seq=1 ttl=59 time=74.375 ms
84 bytes from 192.168.14.2 udp_seq=2 ttl=60 time=14.839 ms
84 bytes from 192.168.14.2 udp_seq=3 ttl=59 time=26.585 ms
84 bytes from 192.168.14.2 udp_seq=4 ttl=60 time=41.938 ms
84 bytes from 192.168.14.2 udp_seq=5 ttl=59 time=26.419 ms
```

Figura 17: Ping da porta UDP.

Na Figura 18 representa o wireshark os vários pacotes de UDP entre o router R1 e R5 pelo caminho inferior da topologia.

É possível observar que os cinco pacotes na captura do Wireshark demonstra a comunicação UDP bem-sucedida entre os endereços IP de origem e destino especificados, validando a configuração e a funcionalidade da configuração da rede. Este fornece evidências do funcionamento adequado dos componentes de rede envolvidos na transmissão do tráfego UDP.

16	16.617094	192.168.12.3	192.168.14.2	UDP	102 53212 → 17000 Len=56
17	17.009428	50:00:00:0d:00:02	CDP/VTP/DTP/PagP/UD...	CDP	355 Device ID: R1 Port ID: GigabitEthernet0/2
18	17.392790	1.1.1.1	7.7.7.7	LDP	72 Keep Alive Message
19	17.664381	7.7.7.7	1.1.1.1	TCP	60 34158 → 646 [ACK] Seq=19 Ack=19 Win=4074 Len=0
20	17.683909	192.168.12.3	192.168.14.2	UDP	102 53212 → 17000 Len=56
21	17.691010	192.168.14.2	192.168.12.3	UDP	98 17000 → 53212 Len=56
22	17.814065	192.168.13.5	224.0.0.2	LDP	76 Hello Message
23	18.705659	192.168.12.3	192.168.14.2	UDP	102 53212 → 17000 Len=56
24	19.245018	192.168.13.6	224.0.0.2	LDP	76 Hello Message
25	19.727033	192.168.12.3	192.168.14.2	UDP	102 53212 → 17000 Len=56
26	19.748596	192.168.14.2	192.168.12.3	UDP	98 17000 → 53212 Len=56
27	20.777825	192.168.12.3	192.168.14.2	UDP	102 53212 → 17000 Len=56
28	22.246079	192.168.13.6	224.0.0.5	OSPF	94 Hello Packet
29	22.807657	192.168.13.5	224.0.0.2	LDP	76 Hello Message
30	23.793404	50:00:00:08:00:00	50:00:00:08:00:00	LOOP	60 Reply
31	25.245601	192.168.13.6	224.0.0.2	LDP	76 Hello Message
32	25.772488	192.168.13.5	224.0.0.5	OSPF	94 Hello Packet

Figura 18: Wireshark do tunnel2.

VI. CONCLUSÃO

A solução implementada procurou estabelecer um sistema de MPLS na topologia de rede, configurando os routers de acordo com os requisitos do projeto. Foram criados túneis entre os routers R1 e R5, com dois caminhos distintos para cada túnel, permitindo balanceamento de carga entre eles.

Além disso, foi implementada a segregação de tráfego TCP e UDP usando listas de acesso e route-maps nos routers. Isto permitiu que o tráfego fosse encaminhado de acordo com as regras especificadas, proporcionando maior controle e segurança na rede.

No geral, o relatório fornece uma visão abrangente do processo de configuração da plataforma EVE-NG, da implementação de um sistema MPLS e do teste e análise do tráfego de rede, demonstrando um bom entendimento dos conceitos e técnicas abordados na disciplina.

RECONHECIMENTO

Os autores desejam expressar a sua sincera gratidão ao Professor Doutor Flávio Oliveira Silva pela orientação e ajuda. O professor não só respondeu a todas as perguntas e esclareceu todas as dúvidas relacionadas ao trabalho prático, mas também forneceu orientação sobre o enunciado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Flavio Silva. *TP2-SETUP Using EVE-NG Community Edition • Using EVE-NG*. Redes de Acesso e Núcleo. Acedido no dia 19 de março de 2024. URL: <https://www.eve-ng.net/index.php/download/#DL-WIN>.
- [2] Flavio Silva. *TP2-SETUP*. Redes de Acesso e Núcleo. Acedido no dia 19 de março de 2024.
- [3] Acedido no dia 16 de abril de 2024. URL: https://www.cisco.com/E-Learning/bulk/public/tac/cim/cib/using_cisco_ios_software/cmdrefs/show_startup-config.htm.



Catarina Pereira atualmente é aluna do primeiro ano do curso de Mestrado de Engenharia de Telecomunicações e Informática na Universidade do Minho, iniciando a sua formação em licenciatura no mesmo curso no ano letivo de 2020/2021. Além dos estudos, foi diretora do Departamento de Saídas Profissionais do NETIUM (Núcleo de Engenharia de Telecomunicações e Informática na Universidade do Minho) para o ano de 2023, tendo atuado como colaboradora do Departamento Pedagógico no ano anterior. Em 2022, ela participou de um projeto de handebol em parceria com o ABC, exercendo o papel de analista do projeto. Catarina também tem experiência em outras áreas, incluindo participação num festival internacional de dança, conclusão de um curso de Língua Gestual Portuguesa, uma residência artística, conclusão do ensino secundário com ênfase em Ciências e Tecnologias, curso de inglês e aulas de música e ballet.



Inês Neves atualmente é aluna do primeiro ano do curso de Mestrado Engenharia de Telecomunicações e Informática na Universidade do Minho, iniciando a sua formação em licenciatura no mesmo curso no ano letivo de 2020/2021. Além dos estudos, ela atuou como vice-diretora da Assembleia Geral do NETIUM (Núcleo de Engenharia de Telecomunicações e Informática na Universidade do Minho) para o ano de 2023, tendo atuado anteriormente como colaboradora do Departamento Pedagógico e do Departamento de Comunicação e Imagem em anos anteriores, em 2021 e em 2022, respetivamente. Em 2022, ela participou num projeto de handebol em parceria com o ABC, exercendo o papel de analista do projeto.

Inês também tem experiência em outras áreas, incluindo a conclusão de um curso de inglês na Royal School of Languages, a prática de natação por muitos anos, bem como ter obtido certificado em ballet. Ela também concluiu o ensino secundário com ênfase em Ciências e Tecnologias.



Leonardo Martins é um estudante do primeiro ano do curso de Mestrado de Engenharia de Telecomunicações e Informática na Universidade do Minho, tendo iniciado a sua formação em licenciatura no mesmo curso no ano letivo de 2020/2021.

Além dos estudos, ele realizou atividades no vice-diretor do Conselho Fiscal do NETIUM (Núcleo de Engenharia de Telecomunicações e Informática na Universidade do Minho) para o ano de 2023, tendo atuado como colaborador do Departamento de Comunicação e Imagem no ano anterior. Em 2022, ele participou de um projeto de handebol em parceria com o ABC, exercendo o papel de *developer* do projeto. Leonardo também tem experiência em outras áreas e conclusão do ensino secundário com ênfase em Ciências e Tecnologias.