

Relatório Aula Prática 1

Planeamento e Gestão de Redes MIEEC

João Nuno Barbosa Neves – up201405198@fe.up.pt
Francisco Fernandes Xavier de Barros – up201506338@fe.up.pt

1. Computadores ligados ao Switch

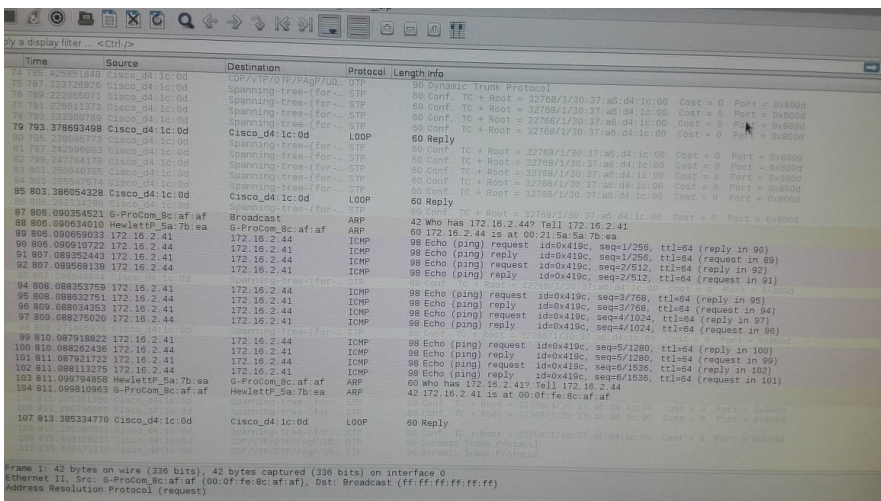
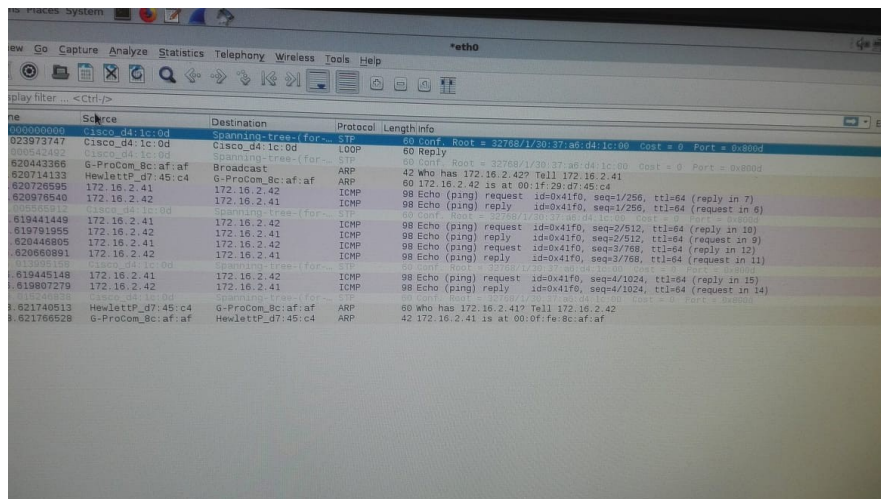
Após ligar o tux1, 2 e 4 ao switch e tendo efetuado a configuração default deste, configurámos os tux's como o guião pede:

tux1: ifconfig eth0 down
ifconfig eth0 172.16.2.41/24
ifconfig eth0 up

tux2: ifconfig eth0 down
ifconfig eth0 172.16.2.42/24
ifconfig eth0 up

tux4: ifconfig eth0 down
ifconfig eth0 172.16.2.44/24
ifconfig eth0 up

Verificou-se que a conexão está corretamente estabelecida com o ping do tux1 para 2 e 4 a funcionar:



2. Criação de vlan x0 no Switch

Nesta experiência era pedido que criássemos uma vlan para o tux 1 e 2 e que provássemos a conectividade entre estes e a não conectividade entre estes e tux 3 e 4 que continuariam na vlan 1, criada por defeito no switch.

Configuração do switch:

```
configure terminal
vlan 40
end
configure terminal
interface fastEthernet 0/1
switchport mode access
switchport access vlan 40
configure terminal
interface fastEthernet 0/2
switchport mode access
switchport access vlan 41
```

Configuração do tux1:

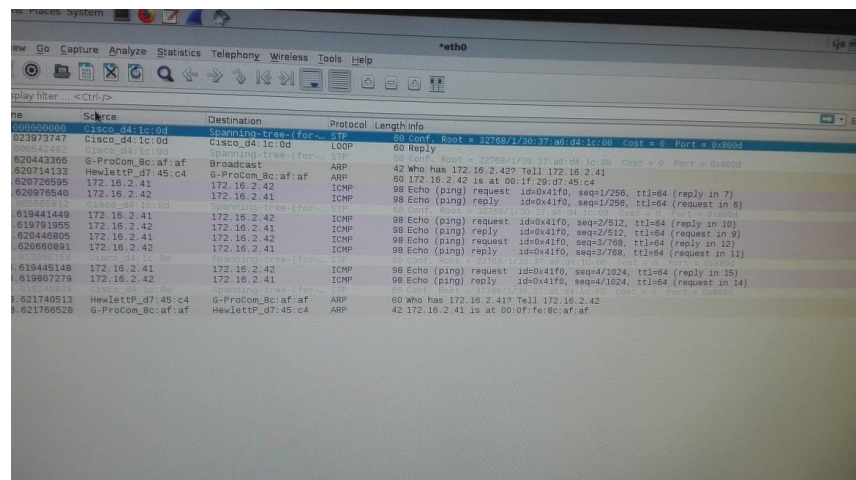
```
ifconfig eth0 down
ifconfig eth0 172.16.40.41/24
ifconfig eth0 up
```

Configuração do tux2:

```
ifconfig eth0 down
ifconfig eth0 172.16.40.42/24
ifconfig eth0 up
```

Seria obviamente necessário fazer as ligações físicas corretamente, neste caso ligar a porta eth0 do tux 1 e 2 às interfaces 1 e 2 do switch, respetivamente.

A conectividade entre estes prova-se na seguinte imagem:



A conectividade entre tux1ou2 e tux3ou4 não é possível pois no tux1ou2 não há rotas configuradas para redes fora de 172.16.40.0.

3. Criação de VLAN em 2 switches

Na experiência 3 era pedido que criássemos uma ligação entre tux's de diferentes bancadas através da mesma vlan configurada com trunking em 2 switches diferentes.

Começámos por configurar o switch da bancada 4 da seguinte forma:

```
configure terminal
vlan 46
end
configure terminal
interface fastEthernet 0/4
switchport mode access
switchport access vlan 46
end
configure terminal
interface gigabitEthernet 0/1
switchport trunk encapsulation dot1q (especificação do protocolo 801.q)
switchport mode trunk
switchport trunk allowed vlan all (permissão as vlan's para monitorizar o tráfego – SPAN)
end
```

O switch da bancada 6 terá a mesma configuração.

O tux4 da bancada 4:

```
ifconfig eth0 down
ifconfig eth0 172.16.46.44/24
ifconfig eth0 up
```

E o tux 4 da bancada 6:

```
ifconfig eth0 down
ifconfig eth0 172.16.46.64/24
ifconfig eth0 up
```

Após fazer as ligações físicas do tux4 para a interface 4 do switch e da interface gigabitEthernet 0/1 do switch para a porta 6.17/4.17 presente na bancada e fazer o paralelo na bancada 6, configuramos uma sessão SPAN no switch para monitorizar o tráfego entre as bancadas:

```
monitor session 1 interface gigabitEthernet 0/1 both
monitor session 1 interface fastEthernet 0/2 encapsulation dot1q
```

Em seguida abrimos o Wireshark no tux2 ligado à interface fastEthernet 0/2 e obtivemos os seguintes logs:

... <Ctrl-J>				
	Source	Destination	Protocol	Length Info
9	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=24/6144, ttl=64 (request in 22)
6	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/46/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
7	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=25/6400, ttl=64 (reply in 26)
13	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=25/6400, ttl=64 (request in 25)
17	Cisco_d4:1c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
22	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
27	Cisco_d4:1c:01	CDP/VTP/DTP/PagP/UD...	DTP	64 Dynamic Trunk Protocol
32	Cisco_d4:1c:01	PVST+	STP	68 Conf. TC + Root = 32768/40/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
37	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=26/6656, ttl=64 (reply in 32)
42	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=26/6656, ttl=64 (request in 31)
47	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/46/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
52	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=27/6912, ttl=64 (reply in 35)
57	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=27/6912, ttl=64 (request in 34)
62	Cisco_d4:1c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
67	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
72	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/40/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
77	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=28/7168, ttl=64 (reply in 40)
82	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=28/7168, ttl=64 (request in 39)
87	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/46/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
92	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=29/7424, ttl=64 (reply in 43)
97	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=29/7424, ttl=64 (request in 42)
102	Cisco_d4:1c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
107	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
112	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/40/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
117	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=30/7680, ttl=64 (reply in 48)
122	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=30/7680, ttl=64 (request in 47)
127	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/46/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
132	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=31/7936, ttl=64 (reply in 51)
137	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=31/7936, ttl=64 (request in 50)
142	Cisco_d4:1c:01	Spanning-tree-(for...	STP	60 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
147	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/1/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
152	Cisco_d4:1c:01	PVST+	STP	68 Conf. Root = 32768/40/30:37:a6:d4:1c:00 Cost = 0 Port = 0x8001
157	172.16.46.44	172.16.46.64	ICMP	102 Echo (ping) request id=0x44cc, seq=32/8192, ttl=64 (reply in 56)
162	172.16.46.64	172.16.46.44	ICMP	98 Echo (ping) reply id=0x44cc, seq=32/8192, ttl=64 (request in 55)
bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0				
... 00:21:5a:c5:61:bb (00:21:5a:c5:61:bb) Dst: HewlettP_c5:61:bb (00:21:5a:c5:61:bb)				

4. Configuração do Spanning Tree Protocol

Nesta experiência era pedido para primeiro voltar à configuração default e em seguida interligar os switches de 3 bancadas diferentes (4, 6 e 2 no nosso caso).

Assim o fizemos e pudemos observar com o comando `'sh spanning-tree active'` que a bancada 2 foi automaticamente estabelecida como root.

Após alguma pesquisa percebemos que isto aconteceu porque a priority da bancada 2 é menor que a das bancadas 4 e 6 e que num caso de empate seria comparado o endereço MAC dos switches das mesmas.

Em seguida vimos que a ligação entre a bancada 4 e 2 tanto de um lado como do outro estava automaticamente desligada, isto foi feito pelo protocolo Spanning-Tree para evitar loops na rede de switches. Quando o desligamos e tentamos pingar qualquer tux de qualquer bancada com qualquer outro verificámos que continuava a funcionar.

Voltamos a ligar o cabo cuja porta estaria bloqueada e desta vez desligamos o cabo que ligava a bancada 2 à 6. O protocolo percebeu esta desconexão e automaticamente atribuiu à ligação de 2 para 4 o que estava antes definido da 2 para a 6.

Em seguida configurámos o switch da bancada 4 para ser o root e vimos que desta vez a ligação entre as bancadas 2 e 6 foi desligada.

Ficou-nos a dúvida de porquê no primeiro caso a ligação entre 4 e 2(root) ter sido a escolhida para ser bloqueada mas no segundo caso foi a ligação entre 6 e 2 (ambos não root) a ser desligada.

5. Configuração de redes IP numa bancada

Nesta experiência começámos por voltar às configurações defeito do switch e em seguida criámos as vlans 40 e 41 e as interfaces a ligar:

Switch:

```
configure terminal
vlan 40
vlan 41
end
configure terminal
interface fastEthernet 0/1
switchport mode access
switchport access vlan 40
end
configure terminal
interface fastEthernet 0/2
switchport mode access
switchport access vlan 40
end
configure terminal
interface fastEthernet 0/4
switchport mode access
switchport access vlan 41
end
```

Ligamos os tux1 e 2 à vlan 40 e o tux 4 à vlan 41 e definimos a rota defeito a usar mais a frente para a ligação dos tux's ao router.

tux1:

```
ifconfig eth0 down
ifconfig eth0 172.16.40.41/24
ifconfig eth0 up
route add default gw 172.16.40.254
```

tux2:

```
ifconfig eth0 down
ifconfig eth0 172.16.40.42/24
ifconfig eth0 up
route add default gw 172.16.40.254
```

tux3:

```
ifconfig eth0 down
ifconfig eth0 172.16.42.43/24
ifconfig eth0 up
```



```
route add default gw 172.16.40.254
```

tux4:

```
ifconfig eth0 down  
ifconfig eth0 172.16.41.44/24  
ifconfig eth0 up  
route add default gw 172.16.41.254
```

O tux 3 estaria automaticamente ligado à vlan 1.

Efetuamos as ligações físicas entre tux's e switch, e a ligação entre a porta gigabitEthernet 0/1 do switch e a gigabitEthernet 0/0 do router.

Depois configuramos o router da seguinte forma:

```
configure terminal  
interface gigabitEthernet 0/0  
no shutdown  
interface gigabitEthernet 0/0.1  
encapsulation dot1Q 40  
ip address 172.16.40.254 255.255.255.0  
no shutdown  
interface gigabitEthernet 0/0.2  
encapsulation dot1Q 41  
ip address 172.16.41.254 255.255.255.0  
no shutdown  
interface gigabitEthernet 0/0.3  
encapsulation dot1Q 1 native  
ip address 172.16.42.254 255.255.255.0  
no shutdown  
end
```

E tentámos pingar o tux 4 e 3 a partir do 1 e 2 e tivemos um resultado positivo.

Conclusões

Este trabalho foi ótimo para rever conceitos, reativar as noções das aulas práticas aprendidas na cadeira de Redes de Computadores e aprender novas formas de interligar redes.

Só precisámos de ver melhor a parte do funcionamento do protocolo Spanning-Tree e tentar finalizar a experiência 6 que por falta de organização de tempo não foi possível concluir.