
Trabalho Laboratorial 4

Planeamento e Gestão de Redes

Routing usando os Protocolos OSPF e BGP4

Diogo Remião & Miguel Pinheiro

Maio 2021



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Faculdade de Engenharia da Universidade do Porto
TEC

Conteúdos

1	Introdução	2
2	Fundamentos Teóricos	3
2.1	OSPF	3
2.2	BGP	4
3	Routing Interior	5
3.1	Configuração da Bancada e Tipologia de rede	5
3.2	Configuração do Quagga	6
3.3	Configuração do <i>daemon</i> Zebra	6
3.4	Configuração da Switch e Router	9
3.5	Configuração do OSPF no Router	10
3.6	Teste de conectividade	11
3.7	Teste de Routing do OSPF	13
3.8	Teste com falha de ligação no OSPF	15
4	Routing Interior	17
4.1	Tipologia de rede e interfaces	17
4.2	Teste de Routing do BGP	19
4.3	Configuração para teste - <i>Highest AS</i>	21
5	Conclusão	24
	Bibliografia	25

1 Introdução

O objetivo deste trabalho consiste na compreensão de vários conceitos ligados ao **Routing**.

Primeiramente será feita uma avaliação teórica dos conceitos de **Routing Interior**, nomeadamente o protocolo **OSPF**. Deste modo será criada com recurso aos sistemas de cada bancada de trabalho um **Autonomous System**.

A segunda parte do trabalho prende-se com a implementação de um protocolo de **Routing Exterior**, o **BGP**. Este protocolo permitira a ligação aos diferentes AS das restantes bancadas, pelo que envolve a cooperação dos restantes colegas na sala.

2 Fundamentos Teóricos

2.1 OSPF

OSPF (*Open Shortest Path First*) é um protocolo de *routing* de internet TCP/IP [1]. Classifica-se como sendo um protocolo de *routing* interior, partilhando deste modo informação entre todos os routers que pertencem ao mesmo AS (*Autonomous System*).

Um **Autonomous System** é um conjunto de routers que comunicam entre e trocam informação através de um protocolo de *routing* comum. Deste modo, um AS está unificado na sua política de *routing* ao exterior.

Este protocolo baseia-se no tecnologia *link-state*. Esta abordagem implica que cada router tem guardada numa base de dados interna a tipologia de rede do AS. Se estiver a operar corretamente, todos os routers têm que ter a mesma informação na base de dados. A comunicação do estado das ligações aos restantes routers é feito através de *flooding*, pelo que rapidamente é possível atualizar todas as bases de dados.

Devido a esta abordagem de comunicação dentro do AS, o OSPF classifica-se também como um **Protocolo de Routing Dinâmico**. Quando a tipologia do AS é mudada, como por exemplo um router desligar-se, o sistema rapidamente deteta a falha. Após um curto tempo de estabilização, associado à frequência com que as mensagens de sincronização HELLO são enviadas, novas rotas são geradas e medidas de forma a apresentar a nova rota ótima, i.e., com o menor custo.

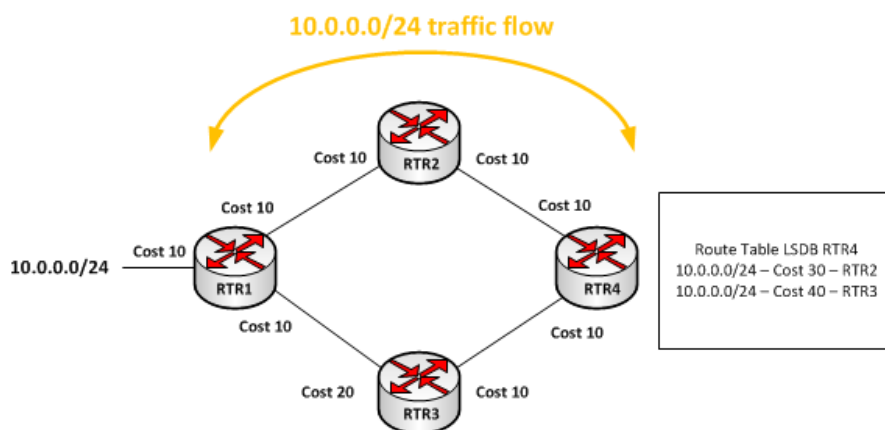


Figure 2.1: Exemplo de tipologia de rede de um AS com custos associados

O custo das rotas está associada à *bandwidth* de cada ligação. Quando maior a largura de banda, menor será o custo da rota. No caso de mais que uma rota ter o mesmo custo, no caso da CISCO, até um máximo de 6 opções são guardadas para o próximo *hop*.

No exemplo apresentado, as duas rotas possíveis de RTR1 para RTR4 são:

- Por RTR2, com custo de 20, rota ótima
- Por RTR3, com custo de 30, rota secundária

No caso do RTR2 deixar de funcionar, o sistema rapidamente iria atualizar e definir a rota pelo RTR3 com a melhor (e única) solução.

2.2 BGP

BGP (*Border Gateway Protocol*) é um protocolo de *routing* externo [2]. Se com o OSPF definimos uma AS e as comunicações entre as diferentes *networks* que dela fazem parte, com BGP permite-se a comunicação entre AS de forma a partilhar informações sobre as *networks* internas de cada AS.

Também é partilhada a informação sobre os AS que estão no caminho para comunicar com essas *networks*.

Deste modo, é possível mapear toda a rede AS, e definir quais os caminhos ótimos para atingir cada AS.

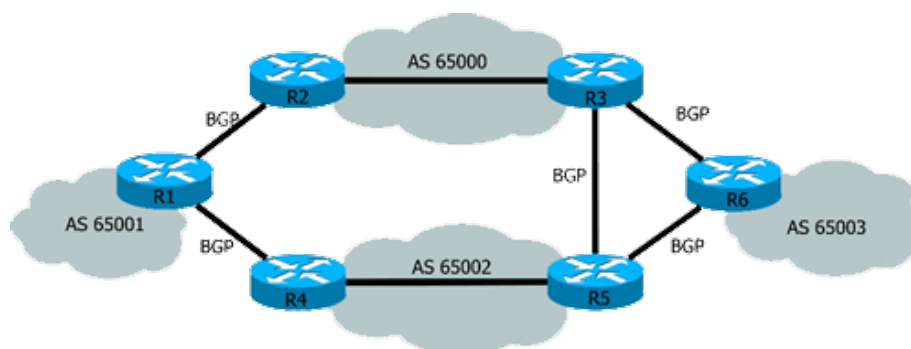


Figure 2.2: Exemplo de tipologia de rede com diferentes AS

Na tipologia exemplo acima apresentada, 4 AS estão interligados através de routers que suportam o protocolo BGP. É de notar que por exemplo, o R5 tem 3 caminhos diferentes possíveis para comunicar com o R3. De forma similar ao **OSPF**, no caso de falha de uma ligação, o sistema é capaz de se adaptar e criar novas rotas.

3 Routing Interior

3.1 Configuração da Bancada e Tipologia de rede

De forma a simular a tipologia de rede pretendida neste trabalho, as interfaces dos *tuxs* foram configuradas da seguinte forma:

- Tux 12 eth2 - 172.16.11.1/24
- Tux 13 eth1 - 172.16.11.2/24
- Tux 13 eth2 - 172.16.12.2/24
- Tux 14 eth1 - 172.16.12.3/24

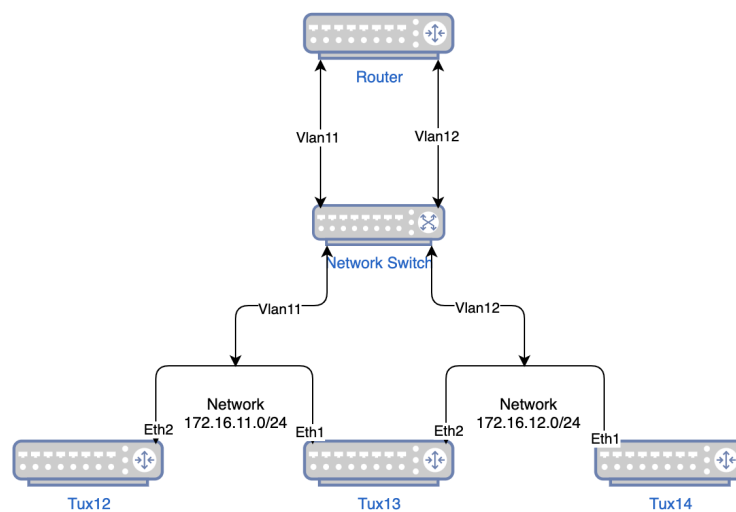


Figure 3.1: Tipologia da rede

Como observado na figura acima, é necessário criar duas Vlans separadas, de forma a simular duas *networks* distintas. Estas *networks* estão ligadas entre si através de dois routers, o router de bancada, e o router no *Tux13*.

3.2 Configuração do Quagga

O **Quagga** [3] é um *software* de routing que permite a implementação dos protocolos de routing **OSPF**, **BGP** e **RIP**. Funciona sobre um *daemon*, **zebra**, que cria uma *layer* de abstração ao *kernel* do Linux, permitindo que diferentes serviços *routing* operem entre os clientes do *Quagga*. Deste modo, este foi instalado em todos os *tuxs*.

O primeiro passo é especificar ao *Quagga* quais os serviços ou *daemons* que vão ser utilizados e que precisam de ser ativados. Neste caso, o ficheiro `/etc/quagga/daemons` foi configurado da seguinte forma:

- `zebra = yes`
- `bgpd = no`
- `ospfd = yes`
- `ospf6d = no`
- `ripd = yes`
- `ripngd = no`

É também necessário ativar *Packet Forwarding* IPv4. Esta configuração está pro predefinição desativada dado que não se espera que um computador se comporte como um *Router* em situações normais. Isto é precisamente o que queremos que aconteça, sendo que esta funcionalidade foi ativada com o comando:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

3.3 Configuração do *daemon* Zebra

O Zebra, tal como indicado anteriormente, é o *daemon* que permite correr os protocolos de routing sobre o *kernel* do Linux. Para configurar o Zebra, é preciso modificar o seu ficheiro `.conf`. Dado que por predefinição nenhum existe, copiamos o ficheiro *default* do diretório `/usr/share/doc/quagga-core/examples/zebra.conf.sample` para o diretório do quagga `/etc/quagga/zebra.conf`.

Neste ficheiro, especificamos vários componentes [4]:

- **Host-name:** O nome do Router que vai operar no computador
- **Password:** Password para acesso root ao terminal do Router
- **Log file:** Ficheiro onde se vai registar as operações do serviço
- **Interfaces** (e respetivos IPs) que vão ser usadas no routing

```
! *- zebra *-  
! zebra sample configuration file  
! $Id: zebra.conf.sample,v 1.1 2002/12/13 20:15:30 paul Exp $  
!  
hostname Router-tux12  
password zebra  
enable password zebra  
log file /var/log/zebra.log  
!  
! Interface's description.  
!  
interface eth2  
    ip address 172.16.11.1/24
```

(a) Ficheiro zebra.conf no Tux12

```
! *- zebra *-  
! zebra sample configuration file  
! $Id: zebra.conf.sample,v 1.1 2002/12/13 20:15:30 paul Exp $  
!  
hostname Router-tux13  
password zebra  
enable password zebra  
log file /var/log/zebra.log  
!  
! Interface's description.  
!  
interface eth1  
    ip address 172.16.12.3/24
```

(b) Ficheiro zebra.conf no Tux14

```
! *- zebra *-  
!  
! zebra sample configuration file  
!  
! $Id: zebra.conf.sample,v 1.1 2002/12/13 20:15:30 paul Exp $  
!  
hostname Router-tux13  
password zebra  
enable password zebra  
!  
! Interface's description.  
!  
interface eth1  
    ip address 172.16.11.2/24  
interface eth2  
    ip address 172.16.12.2/24  
log file /var/log/zebra.log
```

Figure 3.3: Ficheiro zebra.conf no Tux13

Neste ficheiros de configuração, especificamos as interfaces associadas ao serviço *Zebra*, que neste caso apenas vai ter o sub-serviço *OSPF*. Atribuímos também os IPs associados a cada interface, eliminando a necessidade de os configurar manualmente.

O passo seguinte consiste na configuração do *OSPF*. O serviço é criado com o comando `router ospf`, especificando-se a *network* e *area* onde opera. Consequentemente, especificamos as interfaces usadas no serviço *OSPF*.


```
! -*- ospf -*-
!
! OSPFd sample configuration file
!
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
  network 172.16.11.0/24 area 0

interface eth2
log file /var/log/ospf.log
```

(a) Ficheiro ospf.conf no Tux12

```
! -*- ospf -*-
!
! OSPFd sample configuration file
!
!
hostname ospfd
password zebra
enable password zebra
!
router ospf
  network 172.16.12.0/24 area 0

interface eth1
log file /var/log/ospf.log
```

(b) Ficheiro ospf.conf no Tux14

```
! -*- ospf -*-
!
! OSPFd sample configuration file
!
!
hostname ospfd-tux13
password zebra
enable password zebra

router ospf
  network 172.16.11.0/24 area 0
  network 172.16.12.0/24 area 0

interface eth1
interface eth2

log file /var/log/ospfd.log
```

Figure 3.5: Ficheiro ospf.conf no Tux13

3.4 Configuração da Switch e Router

O primeiro passo consiste em separar as duas *networks* em *Vlans* diferentes.

A **Vlan11** corresponde à rede 172.16.11.0/24. Paralelamente, a **Vlan12** corresponde à rede 172.16.12.0/24. Estas duas redes não estão conectadas entre si, pelo que a comunicação só pode ser feita numa layer superior, i.e., nos routers.

À rede 172.16.11.0/24 pertencem as interfaces **Tux12-eth2** e **Tux13-eth1**. À rede 172.16.12.0/24 pertencem as interfaces **Tux13-eth2** e **Tux14-eth1**.

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5 Fa0/6, Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/16, Fa0/17, Fa0/20, Fa0/21 Fa0/22, Fa0/24, Gi0/1
11	VLAN0011	active	Fa0/15, Fa0/19
12	VLAN0012	active	Fa0/18, Fa0/23
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

Figure 3.6: Vlans no Switch

```
interface GigabitEthernet0/2
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 1,11,12
  switchport mode trunk
```

Figure 3.7: Trunk para o Router

No router de bancada, é feita a discriminação das duas Vlans em sub-interfaces diferentes de forma a separar o tráfego que chega. Isto é feito através do encapsulamento configurado no Switch.

```
interface GigabitEthernet0/0.11
  encapsulation dot1Q 11
  ip address 172.16.11.4 255.255.255.0
!
interface GigabitEthernet0/0.12
  encapsulation dot1Q 12
  ip address 172.16.12.4 255.255.255.0
```

Figure 3.8: Interfaces no Router de Bancada

3.5 Configuração do OSPF no Router

Com o serviço OSPF configurado no 3 *Tuxs*, também é preciso configurar o OSPF no próprio router [5].

```
router ospf 1
log-adjacency-changes
auto-cost reference-bandwidth 10
network 172.16.11.0 0.0.0.255 area 0.0.0.0
network 172.16.12.0 0.0.0.255 area 0.0.0.0
default-information originate always
```

Figure 3.9: COnfiguração do OSPF no Router de Bancada

Nesta configuração, especificamos as duas *networks*, correspondentes às diferentes *Vlans*. O comando `default-information originate` especifica aquele router como a rota *default* para comunicações.

O comando `auto-cost reference-bandwidth ref-bw` aplica um custo similar para as interfaces do router. Isto deve-se ao facto das interfaces que ligam ao Router serem *GigaBit*, enquanto que as que entre os *Tuxs* (no Switch) são *Fast Ethernet*. Isto resulta em custos diferentes para as rotas. De forma a uniformizar todos os custos, este comando foi executado quer no Router, quer nos serviços OSPF dos *Tuxs* para garantir custo uniforme de 1.

3.6 Teste de conectividade

Para garantir a conectividade entre todos os sistemas, efetuamos os normais testes *ping* para todas as interfaces.

```
[root@tux13:~# ping 172.16.11.1
PING 172.16.11.1 (172.16.11.1) 56(84) bytes of data.
64 bytes from 172.16.11.1: icmp_seq=1 ttl=64 time=0.122 ms
^C
--- 172.16.11.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.122/0.122/0.122/0.000 ms
[root@tux13:~# ping 172.16.11.2
PING 172.16.11.2 (172.16.11.2) 56(84) bytes of data.
64 bytes from 172.16.11.2: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 172.16.11.2: icmp_seq=2 ttl=64 time=0.017 ms
^C
--- 172.16.11.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 15ms
rtt min/avg/max/mdev = 0.017/0.021/0.025/0.004 ms
[root@tux13:~# ping 172.16.12.2
PING 172.16.12.2 (172.16.12.2) 56(84) bytes of data.
64 bytes from 172.16.12.2: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 172.16.12.2: icmp_seq=2 ttl=64 time=0.016 ms
^C
--- 172.16.12.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 10ms
rtt min/avg/max/mdev = 0.016/0.020/0.025/0.006 ms
[root@tux13:~# ping 172.16.12.3
PING 172.16.12.3 (172.16.12.3) 56(84) bytes of data.
64 bytes from 172.16.12.3: icmp_seq=1 ttl=64 time=0.121 ms
64 bytes from 172.16.12.3: icmp_seq=2 ttl=64 time=0.098 ms
^C
--- 172.16.12.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 32ms
rtt min/avg/max/mdev = 0.098/0.109/0.121/0.015 ms
[root@tux13:~# ping 172.16.12.4
PING 172.16.12.4 (172.16.12.4) 56(84) bytes of data.
64 bytes from 172.16.12.4: icmp_seq=1 ttl=255 time=0.239 ms
64 bytes from 172.16.12.4: icmp_seq=2 ttl=255 time=0.309 ms
^C
--- 172.16.12.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 0.239/0.274/0.309/0.035 ms
root@tux13:~# █
```

Figure 3.10: Pings no Tux13

```
tux-rtr1#ping 172.16.11.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
tux-rtr1#ping 172.16.11.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
tux-rtr1#ping 172.16.12.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
tux-rtr1#ping 172.16.12.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Figure 3.11: Pings no Router de Bancada

Através deste teste de conectividade provamos que todas as interfaces estão corretamente configuradas, assim como as respectivas *Vlans*.

3.7 Teste de Routing do OSPF

O acesso ao terminal do Router virtual criado nos *Tuxs* é feito através do comando `telnet localhost 2604`. As *Routing Tables* apresentadas pelos diferentes sistemas OSPF são as seguintes:

```
[ospfd> show ip ospf route
===== OSPF network routing table =====
N    172.16.11.0/24      [1] area: 0.0.0.0
                        directly attached to eth2
N    172.16.12.0/24      [2] area: 0.0.0.0
                        via 172.16.11.2, eth2
                        via 172.16.11.4, eth2

===== OSPF router routing table =====
R    172.16.12.4         [1] area: 0.0.0.0, ASBR
                        via 172.16.11.4, eth2

===== OSPF external routing table =====
N E2 0.0.0.0/0          [1/1] tag: 1
                        via 172.16.11.4, eth2
```

Figure 3.12: Routing table no Tux12

```
[ospfd-tux13> show ip ospf route
===== OSPF network routing table =====
N    172.16.11.0/24      [1] area: 0.0.0.0
                        directly attached to eth1
N    172.16.12.0/24      [1] area: 0.0.0.0
                        directly attached to eth2

===== OSPF router routing table =====
R    172.16.12.4         [1] area: 0.0.0.0, ASBR
                        via 172.16.11.4, eth1
                        via 172.16.12.4, eth2

===== OSPF external routing table =====
N E2 0.0.0.0/0          [1/1] tag: 1
                        via 172.16.11.4, eth1
                        via 172.16.12.4, eth2
```

Figure 3.13: Routing table no Tux13


```
[ospfd> show ip ospf route
===== OSPF network routing table =====
N    172.16.11.0/24      [2] area: 0.0.0.0
                                via 172.16.12.2, eth1
                                via 172.16.12.4, eth1
N    172.16.12.0/24      [1] area: 0.0.0.0
                                directly attached to eth1

===== OSPF router routing table =====
R    172.16.12.4         [1] area: 0.0.0.0, ASBR
                                via 172.16.12.4, eth1

===== OSPF external routing table =====
N E2 0.0.0.0/0          [1/1] tag: 1
                                via 172.16.12.4, eth1
```

Figure 3.14: Routing table no Tux14

```
[tux-rtr1#show ip ospf route

          OSPF Router with ID (172.16.12.4) (Process ID 1)

          Base Topology (MTID 0)

          Area BACKBONE(0.0.0.0)

          Intra-area Route List
* 172.16.12.0/24, Intra, cost 1, area 0.0.0.0, Connected
   via 172.16.12.4, GigabitEthernet0/0.12
* 172.16.11.0/24, Intra, cost 1, area 0.0.0.0, Connected
   via 172.16.11.4, GigabitEthernet0/0.11
```

Figure 3.15: Routing table no Router de Bancada

Como se pode observar, todos os sistemas têm rotas para as duas *networks*.

Se uma determinada interface fizer parte dessa *Network* (mesma *Vlan*), o custo associado a essa rota é de 1. Caso contrário, se tiver que passar por um outro Router para chegar à *network*, o custo associado é de 2.

O facto de estarem apresentadas como rotas possíveis para aceder a outra *network*, quer o *Tux13*, quer o Router de bancada, comprava que os custos associados a cada rota são iguais. Isto cria uma malha de Router todos equilibrados e com a mesma *bandwidth* aparente nas suas ligações.

Podemos também observar que o Router de Bancada (172.16.12.4), é definido como o caminho *default* em todos os *Tuxs*.

3.8 Teste com falha de ligação no OSPF

Analisemos o cenário representado na figura 3.12. A interface *eth2* pertence à *network* 172.16.11.0/24. Deste modo, se este *Tux* quiser comunicar com o IP 172.16.12.2 que pertence a outra *network*, precisa de passar por um router intermédio.

A tipologia de rede leva a uma preferência, como indicado na figura, de utilizar o *Tux13*, pois corresponde a um menor numero de *Hops* (1), do que passar pelo Router de bancada (2). Um *traceroute* comprova a teoria:

```
root@tux12:~# traceroute 172.16.12.2
traceroute to 172.16.12.2 (172.16.12.2), 30 hops max, 60 byte packets
 1 172.16.12.2 (172.16.12.2)  0.138 ms  0.117 ms  0.101 ms
root@tux12:~#
```

Figure 3.16: Traceroute inicial

No entanto, para testar a capacidade de multipath do protocolo OSPF, vamos simular um cenário de falha de uma ligação. Neste caso, vamos desligar a interface *eth1*, que faz a ponte entre o *Tux12* e *Tux13* na *network* 172.16.11.0/24. Este era o caminho com menor custo para o cenário considerado.

Cortando esta ligação com o comando no *Tux13* `ifconfig eth1 down`, o único caminho possível é através do Router de bancada 172.16.12.4. A tabela de *routing* resultante no *Tux12* para comunicação com a outra *network* já não apresenta o router no *Tux13* como um caminho possível.

```
[ospfd> show ip ospf route
===== OSPF network routing table =====
N    172.16.11.0/24    [1] area: 0.0.0.0
                        directly attached to eth2
N    172.16.12.0/24    [2] area: 0.0.0.0
                        via 172.16.11.4, eth2

===== OSPF router routing table =====
R    172.16.12.4      [1] area: 0.0.0.0, ASBR
                        via 172.16.11.4, eth2

===== OSPF external routing table =====
N E2 0.0.0.0/0        [1/1] tag: 1
                        via 172.16.11.4, eth2
```

Figure 3.17: Routing table no Tux12 depois da falha

Um *traceroute* confirma a rota apresentada pela tabela:


```
root@tux12:~# traceroute 172.16.12.2
traceroute to 172.16.12.2 (172.16.12.2), 30 hops max, 60 byte packets
 1  172.16.11.4 (172.16.11.4)  0.374 ms  0.454 ms  0.487 ms
 2  172.16.12.2 (172.16.12.2)  0.358 ms  0.344 ms  0.327 ms
```

Figure 3.18: Traceroute após modificação da rede

É de notar que demora algum tempo até o sistema atualizar a informação sobre a modificação de uma ligação, neste caso que foi cortada. No entanto, este tempo é curto, dado que por predefinição, o intervalo entre pacotes HELLO é de apenas 10 segundos

4 Routing Interior

4.1 Tipologia de rede e interfaces

Nesta secção será abordada a implementação do protocolo de *routing* exterior BGP.

O primeiro passo é definir qual é a tipologia de rede, i.e., como vão serão definidas as ligações entre os diferentes AS. Tendo cada router de bancada um número limite de interfaces, e estando já as ligações com fios pre-configuradas, a tipologia de rede é definida da seguinte forma:

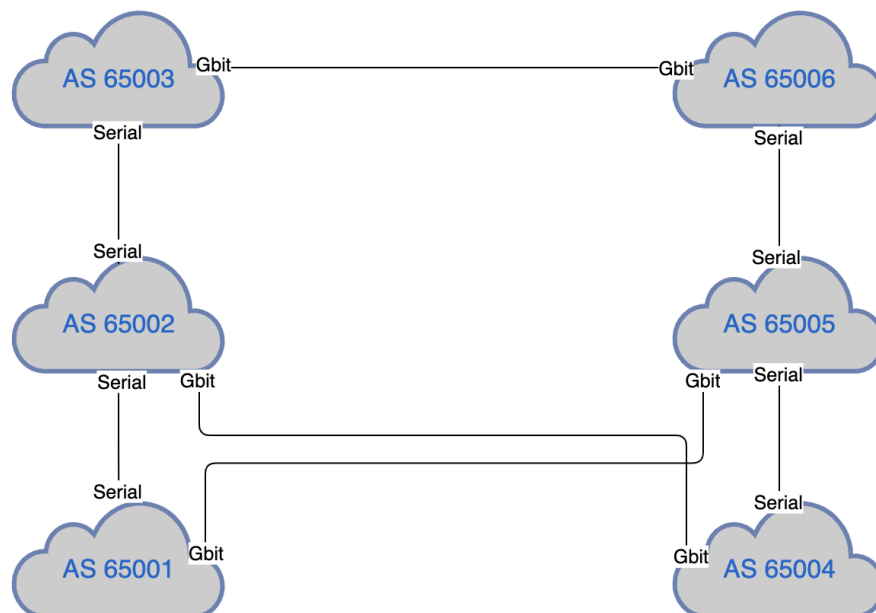


Figure 4.1: Tipologia de rede com os diferentes AS e as respetivas ligações

As ligações *Serial* são feitas sempre com as bancadas adjacentes. Isto deve-se ao facto dos cabos para esta interface serem mais curtos e desse modo não conseguem se ligar aos restantes routers.

As ligações *GigabitEthernet* tem como objetivo a ligação entre as duas filas de bancadas. Já existem disponíveis ligações físicas entre todas as bancadas, onde podemos estabelecer uma ligação a todas as bancadas. No entanto, cada router apenas tem disponível uma interface GigabitEthernet, logo apenas pode fazer uma ligação.

A AS que foi configurada foi a AS-65001. Esta interface é composta por 4 routers, sendo que router de bancada é o **ABR** (*Area Border Router*).

Como indicado no esquemático, a AS-65001 está diretamente ligada à:

- AS-65002 : Rede 192.168.1.48/30 por Serial
- AS-65005 : Rede 192.168.1.60/30 Por GigabitEthernet

Estas duas AS são então as AS vizinhas, informação importante na configuração do processo BGP.

A configuração das duas interfaces foi feita do seguinte modo:

```
interface Serial0/0/0
ip address 192.168.1.49 255.255.255.252
clock rate 2000000
```

Figure 4.2: Atribuição do IP correspondente à interface Serial0/0/1

```
interface GigabitEthernet0/1
ip address 192.168.1.61 255.255.255.252
duplex full
speed auto
```

Figure 4.3: Atribuição do IP correspondente à interface GigabitEthernet0/1

As interfaces vizinhas estão configuradas do seguinte modo:

- AS-65002: Serial - Ip:192.168.1.50
- AS-65005: GigabitEthernet - Ip:192.168.1.62

A configuração do processo BGP no router foi a seguinte [6]:

```
router bgp 65001
no synchronization
bgp log-neighbor-changes
network 172.16.11.0 mask 255.255.255.0
network 172.16.12.0 mask 255.255.255.0
neighbor 192.168.1.50 remote-as 65002
neighbor 192.168.1.62 remote-as 65005
no auto-summary
```

Figure 4.4: Configuração do processo BGP 65001 no router de bancada

Nesta configuração são especificados:

- Neighbors, neste caso o AS-65002 e AS-65005 com os respectivos IPs.
- Networks, que foram definidas no routing interno.
- no synchronization, de forma a impedir a utilização do IBGP no routing interno
- bgp log-neighbor-changes, para atualizar o status das AS vizinhas.
- no auto-summary é aplicado por predefinição e não tem relevância neste contexto

4.2 Teste de Routing do BGP

```
tux-rtr1#sh ip bgp
BGP table version is 35, local router ID is 192.168.1.61
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 172.16.11.0/24  0.0.0.0          0         32768 i
*> 172.16.12.0/24  0.0.0.0          0         32768 i
* 172.16.21.0/24  192.168.1.62     0             0 65005 65004 65002 i
*>                192.168.1.50     0             0 65002 i
* 172.16.22.0/24  192.168.1.62     0             0 65005 65004 65002 i
*>                192.168.1.50     0             0 65002 i
*> 172.16.31.0/24  192.168.1.50     0             0 65002 65003 i
*                  192.168.1.62     0             0 65005 65006 65003 i
*> 172.16.32.0/24  192.168.1.50     0             0 65002 65003 i
*                  192.168.1.62     0             0 65005 65006 65003 i
*> 172.16.41.0/24  192.168.1.62     0             0 65005 65004 i
*> 172.16.42.0/24  192.168.1.62     0             0 65005 65004 i
*> 172.16.51.0/24  192.168.1.62     0             0 65005 i
*> 172.16.52.0/24  192.168.1.62     0             0 65005 i
* 172.16.61.0/24  192.168.1.50     0             0 65002 65003 65006 i
*>                192.168.1.62     0             0 65005 65006 i
* 172.16.62.0/24  192.168.1.50     0             0 65002 65003 65006 i
*>                192.168.1.62     0             0 65005 65006 i
tux-rtr1#
```

Figure 4.5: Routing Table BGP

O BGP apresenta dois caminhos alternativos para cada AS, oferecendo prioridade ao caminho mais curto, i.e, que requer passar por menos AS. No caso de dois caminhos terem o mesmo comprimento, o BGP dá prioridade ao caminho que tem como *next Hop* o AS mais pequeno. Por exemplo, na figura 4.1, o AS65001 pode aceder ao AS65004 através quer do AS65002, quer do AS65005. Por predefinição vai escolher o AS65002.

No momento em que foi tirado o *print* com a configuração, o AS65002 não tinha ligação ao AS65004. Desse modo, apenas é fornecida uma rota possível para o AS6004 e AS6005, que corresponde à ligação direta entre o nosso AS65001 e AS6006. Se essa ligação estivesse ativa, as rotas para o AS6006 seriam:

- AS6001-AS6002-AS6004 (escolhida por ter um AS *identifier* mais pequeno)

- AS6001-AS6605-AS6005

Para confirmar a escolha dos caminhos, são feitos traceroutes para todas as redes:

```
[root@tux13:~# traceroute -n 172.16.21.1
traceroute to 172.16.21.1 (172.16.21.1), 30 hops max, 60 byte packets
 1 172.16.1.19 0.467 ms 0.492 ms 0.523 ms
 2 192.168.1.50 0.656 ms 0.826 ms 1.073 ms
 3 172.16.21.1 1.338 ms * *
[root@tux13:~# traceroute -n 172.16.31.1
traceroute to 172.16.31.1 (172.16.31.1), 30 hops max, 60 byte packets
 1 172.16.1.19 0.494 ms 0.510 ms 0.542 ms
 2 192.168.1.50 0.637 ms 0.833 ms 1.087 ms
 3 192.168.1.94 1.714 ms 4.365 ms 4.078 ms
 4 * * 172.16.31.1 3.143 ms
[root@tux13:~# traceroute -n 172.16.41.1
traceroute to 172.16.41.1 (172.16.41.1), 30 hops max, 60 byte packets
 1 172.16.1.19 0.285 ms 0.343 ms 0.368 ms
 2 192.168.1.62 0.543 ms 0.576 ms 0.668 ms
 3 192.168.1.181 0.697 ms 0.872 ms 1.123 ms
 4 172.16.41.1 1.528 ms * *
[root@tux13:~# traceroute -n 172.16.51.1
traceroute to 172.16.51.1 (172.16.51.1), 30 hops max, 60 byte packets
 1 172.16.1.19 0.186 ms 0.291 ms 0.318 ms
 2 192.168.1.62 0.509 ms 0.625 ms 0.653 ms
 3 172.16.51.1 0.590 ms 0.573 ms 0.556 ms
[root@tux13:~# traceroute -n 172.16.61.1
traceroute to 172.16.61.1 (172.16.61.1), 30 hops max, 60 byte packets
 1 172.16.1.19 0.361 ms 0.393 ms 0.424 ms
 2 192.168.1.62 0.552 ms 0.600 ms 0.695 ms
 3 192.168.1.226 0.771 ms 0.945 ms 1.201 ms
 4 172.16.61.1 1.611 ms * *
```

Figure 4.6: Traceroute para todas as redes do Tux13

Como se pode observar, um acesso a um *neighbor*, como o AS65002 é direto. O primeiro IP é o acesso ao ABR, neste caso o 172.16.1.19 na AS65001. O segundo é o IP do ABR no AS65002, na interface que liga diretamente ao AS65001, 172.16.1.50. O terceiro é o próprio IP da rede de destino.

No caso de um acesso a um AS que não seja vizinho, como é o caso do AS65003, temos que passar primeiro pelo AS65002 e só depois podemos ligar ao AS65003.

Todos estes caminhos são os indicados como preferenciais pelo BGP na figura 4.5.

4.3 Configuração para teste - *Highest AS*

O último teste consiste em dar prioridade a rotas pelo AS com o valor mais elevado. Como indicado anteriormente, o BGP faz precisamente o contrário por predefinição.

Relembrando o exemplo do acesso pelo AS65001 ao AS65004, queremos que seja feito pelo AS65005, e não pelo AS65004. No entanto, no momento da elaboração deste relatório, o AS65004 não tinha conectividade ao AS65004, pelo que este cenário não é possível demonstrar. **Nesta tipologia, não existe mais nenhum cenário para o AS6001 que esta regra possa ser testada.**

Consideramos então outro cenário que permite testar a metodologia: acesso ao AS65003. As duas rotas possíveis são:

- AS65001 -> AS65002 -> AS65003
- AS65001 -> AS65005 -> AS65006 -> AS65003

Como se pode observar, o segundo caminho tem mais *hops*, pelo que não vai ser o escolhido pelo BGP.

```
root@tux13:~# traceroute -n 172.16.31.1
traceroute to 172.16.31.1 (172.16.31.1), 30 hops max, 60 byte packets
 1  172.16.1.19  0.494 ms  0.510 ms  0.542 ms
 2  192.168.1.50  0.637 ms  0.833 ms  1.087 ms
 3  192.168.1.94  1.714 ms  4.365 ms  4.078 ms
 4  * * 172.16.31.1  3.143 ms
```

Figure 4.7: Rota para o AS65003

Para se forçar este caminho, temos que aplicar uma penalização ao AS65002. Isto é obtido com a seguinte configuração [7]:

```
route-map OUT permit 10
  set as-path prepend 65001 65001 65001
```

Figure 4.8: Route-map OUT

Através desta configuração, estamos a adicionar 3 *extra hops*, de forma a aumentar o custo do rota pelo AS65002. O comando `neighbor 192.168.1.50 route-map OUT in` adiciona estes *hops* para tráfego interno, i.e., que está a sair do AS65001 para o AS65002. Deste modo, o caminho do AS65002 para o AS65001 não tem nenhum *hop* extra. Para tal, era preciso adicionar o comando paralelo `neighbor 192.168.1.50 route-map OUT out`. Este comando adiciona *extra hops* para o tráfego externo, i.e., o nosso AS não considera esse *extra hops* mas os restantes consideram.

```

router bgp 65001
  no synchronization
  bgp log-neighbor-changes
  network 172.16.11.0 mask 255.255.255.0
  network 172.16.12.0 mask 255.255.255.0
  neighbor 192.168.1.50 remote-as 65002
  neighbor 192.168.1.50 route-map OUT in
  neighbor 192.168.1.62 remote-as 65005
  no auto-summary

```

Figure 4.9: Associar o route-map OUT ao AS65002

```

* 172.16.31.0/24 192.168.1.50 0 65001 65001 65001 65002 65003 i
*> 192.168.1.62 0 65005 65006 65006 65003 i

```

Figure 4.10: Novas rotas para o AS65003

O AS65006 adicionou também um *hop* extra para acesso feitos do AS65003. No entanto, o custo continua a ser menor, sendo que a nova rota preferencial é pelo AS65005.

O traceroute confirma este cenário:

```

[root@tux13:~# traceroute 172.16.31.1
traceroute to 172.16.31.1 (172.16.31.1), 30 hops max, 60 byte packets
 1 tux-rtr1.netlab.fe.up.pt (172.16.1.19) 0.360 ms 0.393 ms 0.424 ms
 2 192.168.1.62 (192.168.1.62) 0.690 ms 0.718 ms 0.747 ms
 3 192.168.1.226 (192.168.1.226) 0.797 ms 0.995 ms 1.245 ms
 4 192.168.1.145 (192.168.1.145) 1.660 ms 3.255 ms 2.965 ms
 5 172.16.31.1 (172.16.31.1) 2.637 ms 2.055 ms 1.801 ms
root@tux13:~#

```

Figure 4.11: Novo traceroute para o AS65005

Como se pode observar, o caminho escolhido é agora pelo AS65005 (172.168.1.62).

Nota: No final do trabalho, foi possível tirar um *print* que já fornece duas rotas para cada AS. No entanto, os restantes grupos já tinham configurado *prepends*, pelo que o sistema já não está equilibrado. No entanto, ainda é possível observar os caminhos.

```

tux-rtr1#sh ip bgp
BGP table version is 29, local router ID is 192.168.1.61
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 172.16.11.0/24  0.0.0.0              0         32768 i
*> 172.16.12.0/24  0.0.0.0              0         32768 i
* 172.16.21.0/24  192.168.1.62          0          0 65005 65006 65003 65002 i
*>                192.168.1.50          0          0 65001 65001 65001 65002 i
* 172.16.22.0/24  192.168.1.62          0          0 65005 65006 65006 65003 65002 i
*>                192.168.1.50          0          0 65001 65001 65001 65002 i
*> 172.16.31.0/24  192.168.1.62          0          0 65005 65006 65006 65003 i
*                192.168.1.50          0          0 65001 65001 65001 65002 65003 65003 i
*> 172.16.32.0/24  192.168.1.62          0          0 65005 65006 65006 65003 i
*                192.168.1.50          0          0 65001 65001 65001 65002 65003 65003 i
*> 172.16.41.0/24  192.168.1.62          0          0 65005 65004 i
*                192.168.1.50          0          0 65001 65001 65001 65002 65004 i
*> 172.16.42.0/24  192.168.1.62          0          0 65005 65004 i
*                192.168.1.50          0          0 65001 65001 65001 65002 65004 i
*> 172.16.51.0/24  192.168.1.62          0          0 65005 i
*                192.168.1.50          0          0 65001 65001 65001 65002 65004 65005 i
*> 172.16.52.0/24  192.168.1.62          0          0 65005 i
*                192.168.1.50          0          0 65001 65001 65001 65002 65004 65005 i
* 172.16.61.0/24  192.168.1.50          0          0 65001 65001 65001 65002 65004 65005 65006 i
*>                192.168.1.62          0          0 65005 65006 i
*> 172.16.62.0/24  192.168.1.50          0          0 65001 65001 65001 65002 65004 65005 65006 i
*>                192.168.1.62          0          0 65005 65006 i

```

Figure 4.12: Rotas para os diferentes AS

5 Conclusão

Neste trabalho criamos uma tipologia de rede onde nos foi possível implementar o protocolo de *routing* interno **OSPF**. Avaliamos as diferentes rotas e os pesos atribuídos a cada uma. Testamos também um cenário de falha de uma ligação, onde o protocolo foi capaz de se adaptar e criar uma nova rota por outro caminho.

Numa segunda fase, colaboramos com os restantes colegas para implementar o protocolo de *routing* externo **BGP**. Com implementação do OSPF na nossa bancada criamos efetivamente um **Autonomous System**. As restantes bancadas também tinham a sua AS.

Foram criados 6 AS dentro da sala, onde em cada bancada foi configurado um ABR a correr o protocolo BGP. Foi possível observar os diferentes caminhos para as redes dentro de cada AS. Consequentemente, configuramos os routers de modo a modificar as rotas escolhidas pelo BGP para o tráfego externo.

Em suma, adquirimos conhecimentos a nível do funcionamento de protocolos de *routing*, assim como na sua implementação numa configuração de rede laboratorial. Também foi possível configurar protocolos de *routing* numa SDN, transformando os computadores em routers.

Bibliografia

- [1] *OSPF Version 2 Documentation*. IETF - RFC 2328. [Visitado 05-2021]. URL: \url{https://www.ietf.org/rfc/rfc2328.txt}.
- [2] *A Border Gateway Protocol 4 (BGP-4)*. IETF - RFC 1771. [Visitado 05-2021]. URL: \url{https://www.ietf.org/rfc/rfc1771.txt?number=1771}.
- [3] *Quagga Homepage*. Quagga. [Visitado 05-2021]. URL: \url{https://www.quagga.net/}.
- [4] *Chapter 7. Configuring Routing*. Linux Wireless Access Point HOWTO. [Visitado 05-2021]. URL: \url{http://oob.freeshell.org/nzwireless/routing.html}.
- [5] *IP Routing: OSPF Configuration Guide*. Cisco Documentation. [Visitado 05-2021]. URL: \url{https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html}.
- [6] *IP Routing: BGP Configuration Guide*. Cisco Documentation. [Visitado 05-2021]. URL: \url{https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/configuring-a-basic-bgp-network.html}.
- [7] *IP Routing: BGP Configuration Guide, Chapter: BGP Route-Map Continue*. Cisco Documentation. [Visitado 05-2021]. URL: \url{https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/bgp-route-map-continue.html}.