



Universidade do Minho  
Escola de Engenharia

# Mestrado Integrado em Engenharia de Telecomunicações e Informática

## Redes de Computadores II

### Relatório II

David Faria

João Silva

Jorge Bastos

# Índice

1.Introdução.....	pg. 2
2. Implementação do Problema.....	pg. 3
3.Conclusão.....	pg. 20
4.Referências.....	pg. 21
5.Anexos.....	pg.22

# 1.Introdução

No âmbito da unidade curricular de Redes de Computadores 2, foi-nos proposto um segundo trabalho com os objetivos de Sintetizar e implementar soluções de encaminhamento intra e inter-domínio; Configuração de encaminhamento interno RIP e OSPF; Configuração de encaminhamento externo BGPv4; Definição de políticas de encaminhamento; Diagnóstico de problemas de encaminhamento intra e inter-domínio.

Sendo que se pretende emular no CORE vários tipos sistemas autónomos e interligá-los entre si. A interface gráfica CORE será usada para desenhar as topologias de rede e configurar as ligações, os endereços, bem como os protocolos de encaminhamento abordados e, ainda, os protocolos de segurança utilizados. Serão realizados vários testes às diferentes redes configuradas, no âmbito de perceber o correto ou incorreto funcionamento dos protocolos implementados.

O *Common Open Research Emulator* é uma ferramenta desenvolvida para a construção de redes virtuais. Na qualidade de emulador, o CORE constrói uma representação de uma rede que corre em tempo real e se comporta como se fosse real, ao contrário das ferramentas de simulação, onde modelos abstratos são usados para a representação das redes.

Este relatório pretende ilustrar as decisões que tomámos para a realização do trabalho, bem como as nossas perspetivas de resolução do mesmo.

## 2. Implementação do Problema

### 2.1. Conceitos Teóricos

#### 2.1.1. BGP – *Border Gateway Protocol* <sup>[1]</sup>

O BGP é um protocolo de Vetores de Caminho que, ao oposto do RIP e do OSPF, explorados um pouco no trabalho anterior, é um *Exterior Gateway Protocol* (EGP), protocolo este que veio substituir o próprio EGP. BGPv4 é atualmente o padrão normalizado de implementação do protocolo BGP na internet, que essencialmente funciona como uma interligação entre Sistemas Autónomos. O BGP atribui a estes sistemas um número identificador de 16-bits que varia entre 1 e 65535, sendo que a gama de valores entre 64512 e 65535 está reservada para uso privado, ou interno.

Neste contexto, o encaminhamento é feito com base em políticas que permitem aos *routers* fazer a decisão de qual o caminho a escolher, usando as mais variadas métricas, tais como distância, custo, etc.. Este protocolo é usado maioritariamente em circunstâncias onde existem múltiplas conexões para um Sistema Autónomo, dado que o seu maior benefício é o controlo de como o tráfego circula nos sistemas locais.

Para o protocolo funcionar, os *routers* BGP têm de formar relações com os seus vizinhos, ou *peers*, havendo dois tipos de relações fundamentais:

- iBGP – entre vizinhos dentro do mesmo Sistema Autónomo;
- eBGP – entre vizinhos ligados entre si, mas separados por Sistemas Autónomos diferentes.

Em geral, as conexões eBGP são estabelecidas sobre conexões ponto-a-ponto ou sobre redes locais. Se a ligação física é cortada, a sessão eBGP também o é, e todos os prefixos apreendidos por esta, são anunciados como apagados e removidos da tabela de encaminhamento do *router*. Na ordem inversa, as conexões iBGP são geralmente estabelecidas entre endereços lógicos, não associados a uma interface física particular. Isto permite, em caso de rutura da ligação física, se conservar a sessão iBGP ativa se uma ligação alternativa existir e se um protocolo de encaminhamento interno dinâmico for empregue (OSPF, por exemplo).

#### 2.1.2. Sistemas Autónomos <sup>[2]</sup>

Basicamente, a internet encontra-se dividida em domínios hierárquicos, aos quais damos o nome de Sistemas Autónomos. Um Sistema Autónomo, também denominado por AS (*Autonomous System*), é um conjunto de redes informáticas IP integradas na internet, cuja política de encaminhamento interna

(prioridade de encaminhamento, filtragem de rotas, etc.) é coerente, geralmente sob o controlo de uma única entidade/organização, tipicamente um ISP, ou Internet Service Provider.

Podemos dividir os sistemas autónomos em 3 categorias, as quais descrevemos de seguida:

- *Stub* – tem apenas uma conexão para outro AS, sendo que o tráfego de dados entre os domínios pode ser iniciado ou terminado nele mesmo. Os dados num *stub* são emitidos e são recebidos dados, mas sem haver trânsito de dados por ele mesmo;
- *Multihomed* – tem mais do que uma conexão para outros ASs, sendo que mantém as mesmas condições de tráfego que um *stub*, ou seja, não permite o tráfego transitório de dados por ele próprio;
- De Trânsito – é basicamente um AS *multihomed* que permite o tráfego transitório de dados.

### 2.1.3. Segurança [3]

Em relação ao tema da segurança, trata-se de uma das ou até a mais importante preocupação quando criámos uma rede informática, que eventualmente, irá estar ligada à internet. Tornar uma rede segura significa tornar a troca de informação entre *routers* viável, evitando que informações tais como tabelas de encaminhamento estejam erradas, vindas de um *router* não funcional, ou deliberadamente malicioso, fenómeno este provocado quando um intruso quer efetivamente que dois ou mais *routers* recebam informações erradas, de forma a poder entrar na rede, altera-la e até mesmo destruí-la. Daí que se torna quase indispensável autenticar todo este processo de trocas de informações. Para tal existem 3 tipos:

- Autenticação Nula (*Null Authentication*) - Chamada de tipo 0 e não usa qualquer tipo de autenticação. Autenticação fraca, obviamente e é a predefinida;
- Autenticação por texto (*Plain Text Authentication*) - Chamada de tipo 1 e usa uma *password* em texto simples. Autenticação média;
- Autenticação por MD5 (*Message Digest 5 Authentication*) - Chamada de tipo 2 e usa uma *password* baseada no algoritmo MD5. As *passwords* tornam-se uma chave hexadecimal de 128 bits, com 32 dígitos. Autenticação forte.

## 2.2. Análise do Problema

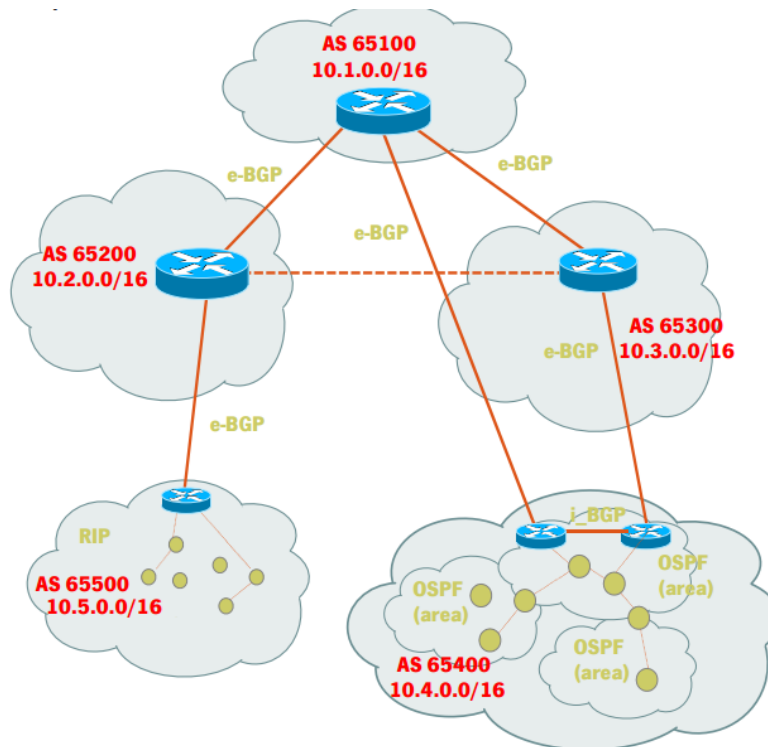


Figura 1 – Topologia proposta para o problema.

Este segundo trabalho prático consiste na gestão do encaminhamento em sistemas autônomos emulados, tendo em conta a topologia descrita pela Figura 1 e tendo como objetivos os seguintes parâmetros:

- Sintetizar e implementar soluções de encaminhamento intra e interdomínio;
- Configuração de encaminhamento interno RIP e OSPF;
- Configuração de encaminhamento externo BGPv4;
- Definição de políticas de encaminhamento;
- Diagnóstico de problemas de encaminhamento intra e interdomínio.

De seguida, passaremos a explicar a nossa abordagem à resolução do problema proposto.



### 2.3.3. Configuração dos protocolos de encaminhamento

#### 2.3.3.1. RIP

Na AS 65500 e, conforme é pedido no enunciado do projeto, tivemos que implementar um encaminhamento interno usando o protocolo dinâmico RIP. Para tal, fazemos uso dos seguintes comandos para configurar os *routers* RIP:

```
(config)# router rip
```

```
(config-router)# version 2
```

```
(config-router)# network 10.5.x.x/30
```

```
(config-router)# end
```

O primeiro comando configura o *router* como fazendo uso do protocolo RIP. A segunda linha de comandos indica que usamos o RIPv2. Esta decisão foi tomada apenas, e só, para facilitar a implementação do protocolo de segurança para este caso. Por último, o terceiro comando indica o endereço de rede da ligação para com outro *router* do género.

#### 2.3.3.2. OSPF

No AS 65400 foi-nos pedido para implementar o protocolo de encaminhamento interno dinâmico OSPF, discriminando 3 áreas OSPF. Para tal efeito os comandos usados para configurar os *routers* OSPF foram os seguintes:

```
(config)# router ospf
```

```
(config-router)# network 10.4.x.x/30 area x
```

```
(config-router)# end
```

À semelhança do RIP, o primeiro comando configura o protocolo e o comando *network* indica o endereço de rede de uma ligação com outro *router* OSPF, mas neste caso a área à qual esse router pertence tem que ser discriminada, pelo que depois da rede indicada é preciso indicar, também, a área à qual pertence dita ligação.

#### 2.3.3.3. BGP

Este protocolo tem de ser configurado nos *routers* de fronteira, ou *Autonomous Systems Border Routers* (ASBR), entre os diferentes sistemas autónomos. Dito isto, a configuração dos ASBRs é a que se segue:



```
(config-router)# end
```

Tabela 2 – Endereçamento da rede 10.5.0.0/16.

Endereço de rede	Endereço de difusão	Máscara de rede	Gama de Endereços
10.5.0.0	10.5.0.3	255.255.255.252	De 10.5.0.1 até 10.5.0.2
10.5.0.4	10.5.0.7	255.255.255.252	De 10.5.0.5 até 10.5.0.6
10.5.0.8	10.5.0.11	255.255.255.252	De 10.5.0.9 até 10.5.0.10
10.5.0.12	10.5.0.15	255.255.255.252	De 10.5.0.13 até 10.5.0.14
10.5.1.0	10.5.1.127	255.255.255.0	De 10.5.1.1 até 10.5.1.126
10.5.2.0	10.5.2.127	255.255.255.0	De 10.5.2.1 até 10.5.2.126
10.5.3.0	10.5.3.127	255.255.255.0	De 10.5.3.1 até 10.5.3.126

Como foi dito anteriormente, este AS *stub* é aquele que implementa o RIP, como tal decidimos colocar alguns terminais nos nós 8, 9 e 10, respetivamente, para testar o processo de convergência do algoritmo RIP. É importante referir que o nó 1 é também um ASBR, como tal vamos analisar mais detalhadamente como foi configurado:

```
!
key chain secure
key 1
key-string projeto
!
interface eth0
ip address 10.5.0.1/30
ip rip authentication key-chain secure
!
interface eth1
ip address 10.5.0.5/30
ip rip authentication key-chain secure
!
interface eth2
ip address 192.168.1.1/30
!
router rip
version 2
network 10.5.0.0/30
network 10.5.0.4/30
redistribute bgp
!
```

Figura 4 – zebra do router n1.

Como foi referido anteriormente, fazemos uso do RIPv2 para podermos implementar o protocolo de segurança através de texto simples. Para tal, e como podemos ver, usamos um “key chain” com o nome secure, no qual definimos a *password* projeto e, passamos a informação através das interfaces que conectam aos outros *routers* RIP de que a autenticação das mensagens RIP é esta.

De referir que para os ASBRs dentro dum sistema autónomo onde está implementado algum tipo de protocolo de encaminhamento dinâmico, é importante fazer o comando “redistribute bgp” para os routers RIP ou OSPF aprenderem as rotas BGP, de modo a poderem enviar tráfego para fora do AS.

Para comprovar o bom funcionamento do RIP, a imagem seguinte mostra a tabela de encaminhamento do router RIP no nó 3, no qual ele adquire conhecimento das redes internas e externas fornecidas pelo BGP e ele próprio é o melhor caminho para alguns destinos, tais como a rede 10.5.3.0/24.



#### 2.3.4.3. AS 65100



Figura 7 – Topologia do AS 65100.

Para o sistema de tráfego AS 65100, o endereçamento usado, foi o seguinte:

Tabela 4 – Endereçamento da rede 10.1.0.0/16.

Endereço de rede	Endereço de difusão	Máscara de rede	Gama de Endereços
10.1.0.0	10.1.0.3	255.255.255.252	De 10.1.0.1 até 10.1.0.2

#### 2.3.4.4. AS 65300

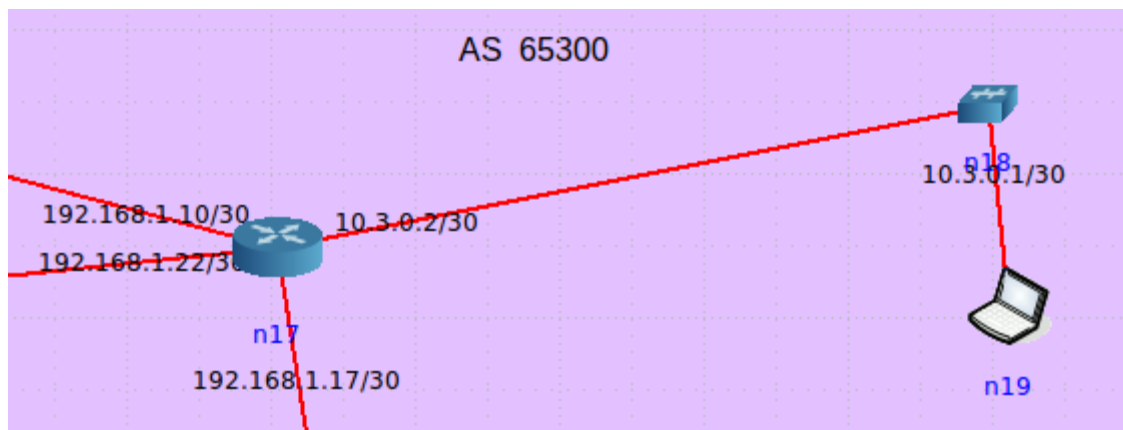


Figura 8 – Topologia do AS 65300.

O endereçamento referente ao AS 65300 é descrito na tabela seguinte:

Tabela 5 – Endereçamento da rede 10.3.0.0/16.

Endereço de rede	Endereço de difusão	Máscara de rede	Gama de Endereços
10.3.0.0	10.3.0.3	255.255.255.252	De 10.3.0.1 até 10.3.0.2

### 2.3.4.5. AS 65400

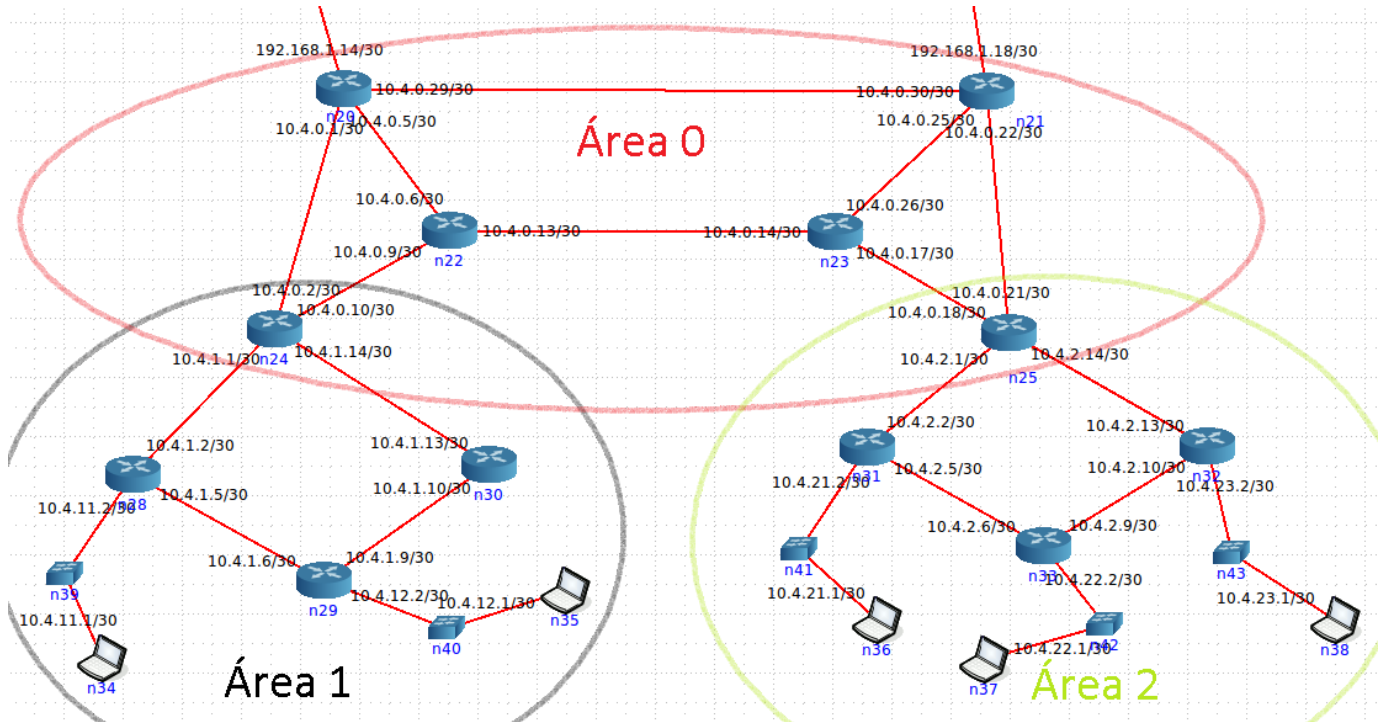


Figura 9 – Topologia do AS 65400, com identificação das diferentes áreas OSPF.

Tendo em conta que tínhamos que definir 3 áreas para implementar o protocolo OSPF, o endereçamento nesse respetivos blocos foi feito de acordo com a tabela seguinte.

Tabela 6 – Endereçamento das áreas OSPF da rede 10.4.0.0/16.

Número de Área	Endereço de rede	Endereço de difusão	Máscara de rede	Gama de Endereços
0	10.4.0.0	10.4.0.3	255.255.255.252	De 10.4.0.1 até 10.4.0.2
	10.4.0.4	10.4.0.7	255.255.255.252	De 10.4.0.5 até 10.4.0.6
	10.4.0.8	10.4.0.11	255.255.255.252	De 10.4.0.9 até 10.4.0.10
	10.4.0.12	10.4.0.15	255.255.255.252	De 10.4.0.13 até 10.4.0.14
	10.4.0.16	10.4.0.19	255.255.255.252	De 10.4.0.17 até 10.4.0.18
	10.4.0.20	10.4.0.23	255.255.255.252	De 10.4.0.21 até 10.4.0.22
1	10.4.1.0	10.4.1.3	255.255.255.252	De 10.4.0.1 até 10.4.0.2
	10.4.1.4	10.4.1.7	255.255.255.252	De 10.4.0.5 até 10.4.0.6

	10.4.1.8	10.4.1.11	255.255.255.252	De 10.4.0.9 até 10.4.0.10
	10.4.1.12	10.4.1.15	255.255.255.252	De 10.4.0.13 até 10.4.0.14
	10.4.11.0	10.4.11.3	255.255.255.252	De 10.4.11.1 até 10.4.11.2
	10.4.12.0	10.4.12.3	255.255.255.252	De 10.4.12.1 até 10.4.12.2
2	10.4.2.0	10.4.2.3	255.255.255.252	De 10.4.2.1 até 10.4.2.2
	10.4.2.4	10.4.2.7	255.255.255.252	De 10.4.2.5 até 10.4.2.6
	10.4.2.8	10.4.2.11	255.255.255.252	De 10.4.2.9 até 10.4.2.10
	10.4.2.12	10.4.2.15	255.255.255.252	De 10.4.2.13 até 10.4.2.14
	10.4.21.0	10.4.21.3	255.255.255.252	De 10.4.21.1 até 10.4.21.2
	10.4.22.0	10.4.22.3	255.255.255.252	De 10.4.22.1 até 10.4.22.2
	10.4.23.0	10.4.23.3	255.255.255.252	De 10.4.23.1 até 10.4.23.2

O AS 65400 é o único AS da topologia que é *multihomed* e, como tal, tem o iBGP configurado nos seus ASBRs, que comporta-se basicamente da mesma maneira que o eBGP mostrado anteriormente.

Para além do mais, o serviço OSPF encontra-se dividido em 3 áreas que pela figura acima se encontram reveladas. Tendo em conta tal facto, vamos fazer a análise da tabela do OSPF do *router* de fronteira entre a área 0 e a área 1 no nó 24.

```

===== OSPF network routing table =====
N  10.4.0.0/30      [10] area: 0.0.0.0
                        directly attached to eth0
N  10.4.0.4/30      [20] area: 0.0.0.0
                        via 10.4.0.1, eth0
                        via 10.4.0.9, eth1
N  10.4.0.8/30      [10] area: 0.0.0.0
                        directly attached to eth1
N  10.4.0.12/30     [20] area: 0.0.0.0
                        via 10.4.0.9, eth1
N  10.4.0.16/30     [30] area: 0.0.0.0
                        via 10.4.0.9, eth1
N  10.4.0.20/30     [40] area: 0.0.0.0
                        via 10.4.0.9, eth1
N  10.4.0.24/30     [30] area: 0.0.0.0
                        via 10.4.0.9, eth1
N  10.4.1.0/30      [10] area: 0.0.0.1
                        directly attached to eth2
N  10.4.1.4/30      [20] area: 0.0.0.1
                        via 10.4.1.2, eth2
N  10.4.1.8/30      [20] area: 0.0.0.1
                        via 10.4.1.13, eth3
N  10.4.1.12/30     [10] area: 0.0.0.1

```

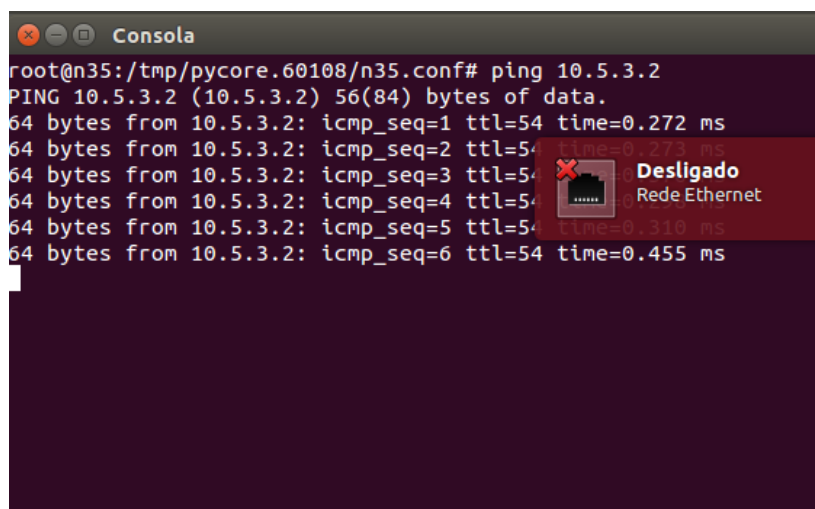
Figura 10 – Comando “show ip ospf route” para mostra a tabela de encaminhamento do OSPF no nó 24

Como podemos comprovar, ele consegue fazer a distinção entre os caminhos da área 0 (0.0.0.0) e da área 1 (0.0.0.1), sabendo os melhores caminhos para chegar aos restantes pontos das áreas que não se encontram diretamente conectados.

A segurança no OSPF, à imagem do RIP, é conseguida através das interfaces dos router propagarem as *passwords* de segurança umas às outras estando em concordância, mas no caso do OSPF é ainda mais simples, basta fazer uso da seguinte linha de código, em que o campo PASSWORD é preenchido com a palavra passe que desejemos propagar:

```
ip ospf authentication-key PASSWORD
```

Posto isto, vamos fazemos uma verificação mais abrangente da topologia a ver se conseguíamos comunicar com sucesso com qualquer outro AS da topologia, nomeadamente o AS 65500, mais propriamente o sistema terminal contendo o endereço publico, 10.5.3.2, cujo *ping* pode ser visto na figura seguinte.



```
root@n35:/tmp/pycore.60108/n35.conf# ping 10.5.3.2
PING 10.5.3.2 (10.5.3.2) 56(84) bytes of data.
64 bytes from 10.5.3.2: icmp_seq=1 ttl=54 time=0.272 ms
64 bytes from 10.5.3.2: icmp_seq=2 ttl=54 time=0.273 ms
64 bytes from 10.5.3.2: icmp_seq=3 ttl=54 time=0.273 ms
64 bytes from 10.5.3.2: icmp_seq=4 ttl=54 time=0.310 ms
64 bytes from 10.5.3.2: icmp_seq=5 ttl=54 time=0.310 ms
64 bytes from 10.5.3.2: icmp_seq=6 ttl=54 time=0.455 ms
```

Figura 11 – ping efetuado entre o terminal em AS 65400 no nó 35 e o terminal em AS 65500 no nó 10.

## 2.3.5. Políticas de Encaminhamento Externas

As relações entre os sistemas autónomos são estabelecidas fazendo uso do protocolo BGP, já explicado anteriormente, para que possa preencher os requisitos do problema e estabelecer as ligações e políticas desejadas.

### 2.3.5.1. Relação entre AS65200 e AS65300

Como referido no enunciado deste projeto, estes dois sistemas autónomos têm um contrato de parceria entre eles que lhes permite encaminhar diretamente o tráfego das suas redes internas. Além do mais, estes dois AS's funcionam como sistemas de tráfego, ou seja, encaminham tráfego proveniente de outros sistemas autónomos.

Para que tal aconteça, ambos anunciam aos outros sistemas autónomos com os quais estabelecem ligação, os anúncios recebidos. Se a ligação entre AS 65200 e o AS 65100 se quebrar, o AS

65200 deverá poder usar o AS 65300 para encaminhar tráfego para o AS 65400. De igual modo se a ligação entre o AS 65300 e o AS 65100 se quebrar, o AS 65300 deverá usar o AS 65200 para encaminhar tráfego para o AS 65500.

Os comandos utilizados para implementar esta relação encontram-se de seguida.

*Router n11 (AS65200):*

...

*neighbor 192.168.1.22 route-map setlocalin in*

*!*

*ip as-path access-list 1 permit 65400^\$*

*!*

*route-map setlocalin permit 10*

*match as-path 1*

*set local-preference 200*

*route-map setlocalin permit 20*

*set local-preference 150*

*!*

*Router n17 (AS65300):*

...

*neighbor 192.168.1.21 route-map setlocalin in*

*!*

*ip as-path access-list 1 permit 65500^\$*

*!*

*route-map setlocalin permit 10*

*match as-path 1*

*set local-preference 200*

*!*

*route-map setlocalin permit 20*

*set local-preference 150*

*!*

Como pode ser facilmente observado, foram criadas condições para que pudéssemos implementar o exigido no enunciado. Na secção das Tabelas de Encaminhamento será mais perceptível as restrições aplicadas aqui descritas.

## 2.3.6. Tabelas de Encaminhamento

As restrições exigidas no enunciado do projeto, foram implementadas recorrendo ao código descrito na secção dos Anexos. Nesta secção pretende-se apresentar as tabelas de encaminhamento dos *routers* de fronteira de todos os sistemas autónomos.

Na Figura 12 observámos a tabela de encaminhamento do *router* de fronteira do sistema autónomo 65500.



#### 2.3.6.1. AS65500 (router n1)

```
Terminal
```

```
BGP table version is 0, local router ID is 10.5.0.1  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
               r RIB-failure, S Stale, R Removed  
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.1.0.0/16	192.168.1.2			0	65200 65300 65100 i
*> 10.2.0.0/16	192.168.1.2	0		0	65200 i
*> 10.3.0.0/30	192.168.1.2			0	65200 65300 i
*> 10.4.0.0/16	192.168.1.2			0	65200 65300 65400 i
*> 10.5.0.0/16	0.0.0.0	0		32768	i

```
Total number of prefixes 5  
  
~  
~  
~ 10.5.0.3/30  
~ 10.5.0.2/30  
~ 10.5.0.1/30  
~ 10.5.0.0/30 n3  
~ 10.4.11.2/24  
  
(END)
```

Figura 12 – Tabela de Encaminhamento do *router* de fronteira do sistema autónomo 65500.

Este *router* não tem qualquer restrição em relação à sua política de encaminhamento. Podemos porém concluir que faz ligação direta com o AS65200.

Na Figura 13 encontra-se a tabela de encaminhamento do *router* de fronteira do sistema autónomo 65200.

#### 2.3.6.2. AS65200 (router n11)

```

Terminal
Hello, this is Quagga (version 0.99.22.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

n11# show ip bgp
BGP table version is 0, local router ID is 10.2.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop           Metric LocPrf Weight Path
*> 10.1.0.0/16    192.168.1.6         0             0 65100 i
*> 10.2.0.0/16    0.0.0.0             0             32768 i
* 10.3.0.0/30     192.168.1.6         0             0 65100 65300 i
*> 10.4.0.0/16    192.168.1.22        0            150      0 65300 i
* 10.5.0.0/16     192.168.1.1         0             0 65100 65400 i
*> 10.6.0.0/16    192.168.1.1         0             0 65300 65400 i
*> 10.7.0.0/16    192.168.1.1         0             0 65500 i

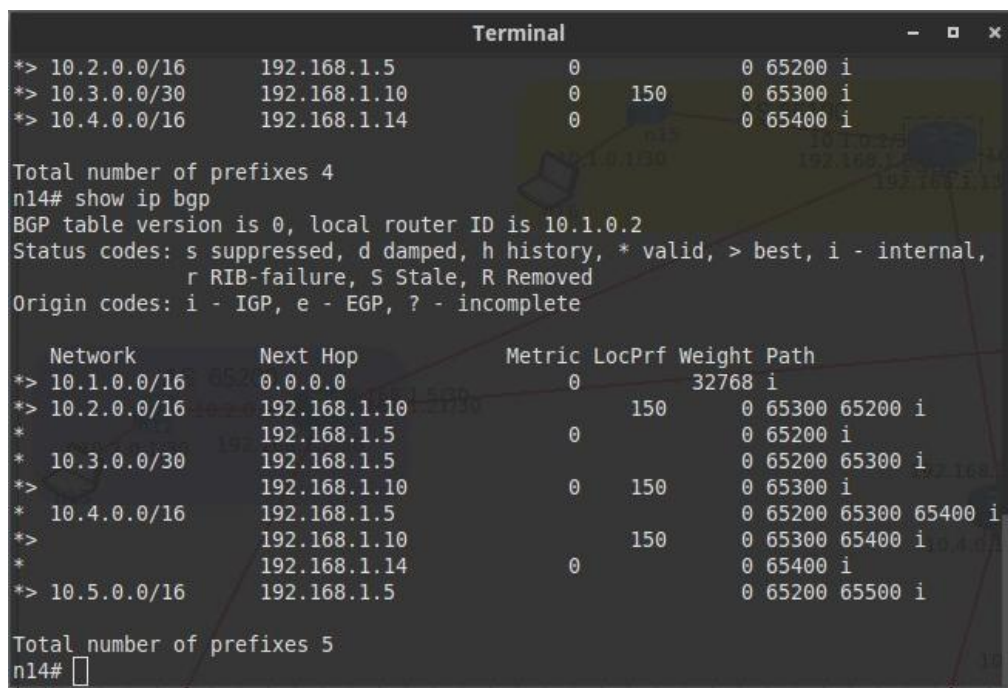
Total number of prefixes 5
(END)

```

Figura 13 – Tabela de encaminhamento do router n11 (AS200).

Podemos verificar que o tráfego com destino ao AS65300 tem uma ligação direta pela interface 192.168.1.22, tal como pedido no enunciado. Verificámos também neste caso que se a ligação entre o AS65200 e o AS65100 quebrar, o AS200 utiliza o AS300 para encaminhar tráfego para o AS400.

### 2.3.6.3. AS65100 (router n14)



```

Terminal
*> 10.2.0.0/16      192.168.1.5          0          0 65200 i
*> 10.3.0.0/30      192.168.1.10         0      150    0 65300 i
*> 10.4.0.0/16      192.168.1.14         0          0 65400 i

Total number of prefixes 4
n14# show ip bgp
BGP table version is 0, local router ID is 10.1.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 10.1.0.0/16     0.0.0.0          0           32768 i
*> 10.2.0.0/16     192.168.1.10     0      150    0 65300 65200 i
* 10.2.0.0/16     192.168.1.5      0           0 65200 i
* 10.3.0.0/30     192.168.1.5      0           0 65200 65300 i
*> 10.4.0.0/16     192.168.1.10     0      150    0 65300 i
* 10.4.0.0/16     192.168.1.5      0           0 65200 65300 65400 i
*> 10.5.0.0/16     192.168.1.10     0      150    0 65300 65400 i
* 10.5.0.0/16     192.168.1.14     0           0 65400 i
*> 10.5.0.0/16     192.168.1.5      0           0 65200 65500 i

Total number of prefixes 5
n14#

```

Figura 14 – Tabela de encaminhamento do router n14 (AS65100).

Ao efetuarmos a análise a esta tabela de encaminhamento podemos concluir que todas as restrições funcionam como o esperado. O tráfego do AS65100 para o AS65500 utiliza o AS65200, uma vez que não pode usar o AS65300 para tal efeito. Já o tráfego deste sistema autónomo para o sistema autónomo 65400 utiliza o AS65300, tal como o pedido no enunciado. Vendo isto, podemos concluir que as restrições estão a funcionar corretamente. É de notar que este AS é o ISP do AS65200, do AS65300 e do AS65400, uma vez que aceita as suas rotas e as anuncia.

2.3.6.4. AS65300 (router n17)

```
Terminal
*> 10.5.0.0/16      192.168.1.21      150      0 65200 65500 i

Total number of prefixes 5

n17# show ip bgp
BGP table version is 0, local router ID is 10.3.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
* 10.1.0.0/16     192.168.1.18          0         0 65400 65100 i
*>                192.168.1.21      150         0 65200 65100 i
*                 192.168.1.9          0         0 65100 i
*> 10.2.0.0/16     192.168.1.21          0      150      0 65200 i
*                 192.168.1.9          0         0 65100 65200 i
*> 10.3.0.0/30     0.0.0.0          0      32768 i
* 10.4.0.0/16     192.168.1.9          0         0 65100 65400 i
*>                192.168.1.18          0         0 65400 i
* 10.5.0.0/16     192.168.1.9          0         0 65100 65200 65500 i
*>                192.168.1.21      150         0 65200 65500 i

Total number of prefixes 5
(END)
```

Figura 15 – Tabela de encaminhamento do router n17 (AS65300).

Na análise a este à tabela de encaminhamento deste router de fronteira, é facilmente observável que estabelece uma ligação direta com o sistema autónomo AS65200 e que se a ligação que possui com o router de fronteira do sistema autónomo 65100 se quebrar, consegue utilizar o AS65200 para encaminhar tráfego para o AS65500. Este sistema autónomo é o ISP do AS65400.

2.3.6.5. AS65400 (router n20 e router n21)

Este sistema autónomo é um sistema *multihomed* (anteriormente descrito) e possui dois ASBRs, pelo que é necessário apresentar ambas as tabelas de encaminhamento. Na Figura16 encontramos a tabela de encaminhamento do router 20 e na Figura17 a tabela de encaminhamento do router 21 (o que se encontra do lado esquerdo da topologia).



### 3. Conclusão

Ao longo da realização deste projeto, foi possível aprimorar conhecimentos que já tinham ficado do primeiro trabalho, bem como aprender de uma forma mais dinâmica o protocolo BGP e a sua diversidade.

As políticas de encaminhamento e o seu funcionamento e implementação, como objetivo deste trabalho, permitiram um estudo mais aprofundado da questão do encaminhamento e, consequentemente, uma melhor percepção das técnicas atualmente existentes, sendo que foi a parte que causou maiores adversidades à correta realização do mesmo.

Numa análise posterior, muitos dos objetivos propostos foram cumpridos, tais como o do sistema autónomo *Stub* e *Multihomed* (este com algumas falhas menores) e algumas políticas de encaminhamento externo funcionaram, problema esse que tentamos corrigir neste relatório, mas no geral, continuamos a acreditar que conseguimos atingir medianamente os objetivos requeridos e ultrapassar alguns obstáculos que sem algum esforço e dedicação, não seriam de todo possíveis.

## 4. Referências

[1]: "BGPExpert.com What is BGP?", <http://www.bgpexpert.com/what.php>. Visitado a 15 de Junho.

[2]: "Autonomous System numbers - FAQs", <https://www.apnic.net/services/services-apnic-provides/helpdesk/faqs/asn-faqs>. Visitado a 15 de Junho.

[3]: "Sample Configuration for Authentication in OSPF",  
<http://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13697-25.html>. Visitado a 15 de Junho.

[4]: Nicolau, M.J., Acetatos das aulas de Redes de Computadores 1 e 2.

[5]: Manual Quagga, <http://www.quagga.net/>

## 5. Anexos

### 5.1. Código correspondente aos routers de fronteira dos sistemas autónomos.

#### AS500

```
!  
key chain secure  
key 1  
key-string projeto  
!  
interface eth0  
ip address 10.5.0.1/30  
ip rip authentication key-chan secure  
!  
interface eth1  
ip address 10.5.0.5/30  
ip rip authentication key-chan secure  
!  
interface eth2  
ip address 192.168.1.1/30  
!  
router rip  
version 2  
network 10.5.0.0/30  
network 10.5.0.4/30  
redistribute bgp  
!  
router bgp 65500  
bgp router-id 10.5.0.1  
network 10.5.0.0/16  
neighbor 192.168.1.2 remote-as 65200  
!
```

#### AS200

```
interface eth0  
ip address 192.168.1.2/30  
!  
interface eth1  
ip address 10.2.0.2/30  
!  
interface eth2  
ip address 192.168.1.5/30
```

```

!
interface eth3
ip address 192.168.1.21/30
!
router bgp 65200
bgp router-id 10.2.0.2
network 10.2.0.0/16
neighbor 192.168.1.1 remote-as 65500
neighbor 192.168.1.6 remote-as 65100
neighbor 192.168.1.22 remote-as 65300

neighbor 192.168.1.22 route-map setlocalin in
!
ip as-path access-list 1 permit 65400^$
!
route-map setlocalin permit 10
match as-path 1
set local-preference 200

route-map setlocalin permit 20
set local-preference 150
!

```

## AS100

```

interface eth0
ip address 10.1.0.2/30
!
interface eth1
ip address 192.168.1.6/30
!
interface eth2
ip address 192.168.1.9/30
!
interface eth3
ip address 192.168.1.13/30
!
router bgp 65100
bgp router-id 10.1.0.2
network 10.1.0.0/16
neighbor 192.168.1.5 remote-as 65200
neighbor 192.168.1.14 remote-as 65400
neighbor 192.168.1.10 remote-as 65300

neighbor 192.168.1.10 route-map setlocalin in
neighbor 192.168.1.5 route-map caminho in
!

```



```

ip as-path access-list 1 permit 65400^$
ip as-path access-list 2 permit 65500^$
!
route-map setlocalin permit 10
match as-path 1
set local-preference 200

route-map setlocalin permit 20
set local-preference 150

route-map caminho permit 10
match as-path 2
set local-preference 200

route-map setlocalin permit 20
set local-preference 150
!

```

## AS 300

```

interface eth0
ip address 10.3.0.2/30
!
interface eth1
ip address 192.168.1.10/30
!
interface eth2
ip address 192.168.1.17/30
!
interface eth3
ip address 192.168.1.22/30
!
router bgp 65300
bgp router-id 10.3.0.2
network 10.3.0.0/30

neighbor 192.168.1.9 remote-as 65100
neighbor 192.168.1.18 remote-as 65400
neighbor 192.168.1.21 remote-as 65200

neighbor 192.168.1.21 route-map setlocalin in
!
ip as-path access-list 1 permit 65500^$
!
route-map setlocalin permit 10
match as-path 1
set local-preference 200

```

```
route-map setlocalin permit 20
set local-preference 150
!
```

## AS400 router n20

```
interface eth0
ip address 10.4.0.1/30
ip ospf authentication-key projeto
!
interface eth1
ip address 10.4.0.5/30
ip ospf authentication-key projeto
!
interface eth2
ip address 10.4.0.29/30
ip ospf authentication-key projeto
!
interface eth3
ip address 192.168.1.14/30
!
router bgp 65400
bgp router-id 10.4.0.1
network 10.4.0.0/16

neighbor 192.168.1.13 remote-as 65100
neighbor 10.4.0.30 remote-as 65400
neighbor 10.4.0.30 next-hop-self

neighbor 10.4.0.30 route-map setlocalin in
neighbor 192.168.1.13 filter-list 1 out
!
ip as-path access-list 1 permit 65200^$
ip as-path access-list 2 permit 65500^$
ip as-path access-list 1 deny ^$
!
route-map setlocalin permit 10
match as-path 1
set local-preference 200

route-map setlocalin permit 10
match as-path 2
set local-preference 200

route-map setlocalin permit 20
```

*set local-preference 150*

*!*

*router ospf*

*network 10.4.0.0/30 area 0*

*network 10.4.0.4/30 area 0*

*redistribute bgp*

*area 0 authentication*

*!*

## **AS400 router n21**

*interface eth0*

*ip address 10.4.0.22/30*

*ip ospf authentication-key projeto*

*!*

*interface eth1*

*ip address 10.4.0.25/30*

*ip ospf authentication-key projeto*

*!*

*interface eth2*

*ip address 10.4.0.30/30*

*ip ospf authentication-key projeto*

*!*

*interface eth3*

*ip address 192.168.1.18/30*

*!*

*router bgp 65400*

*bgp router-id 10.4.0.30*

*network 10.4.0.0/16*

*neighbor 192.168.1.17 remote-as 65300*

*neighbor 10.4.0.29 remote-as 65400*

*neighbor 10.4.0.29 next-hop-self*

*neighbor 10.4.0.29 route-map setlocalin in*

*!*

*ip as-path access-list 1 permit 65100^\$*

*!*

*route-map setlocalin permit 10*

*match as-path 1*

*set local-preference 200*

*route-map setlocalin permit 20*

*set local-preference 150*

```
!  
router ospf  
network 10.4.0.20/30 area 0  
network 10.4.0.24/30 area 0  
redistribute bgp  
area 0 authentication  
!
```