

TP4: Encaminhamento de Tráfego [Protocolo BGP]

João Guedes (PG47329), Paulo Sousa (PG47556), and Renata Teixeira (PG47603)

Universidade do Minho, Braga, Portugal
Mestrado em Engenharia Informática
Interligação de Redes IP
Grupo 9
16 de Maio de 2022

1 Descrição do Problema

Pretende-se implementar e testar um cenário de encaminhamento global envolvendo vários Sistemas Autónomos (ASs) interligados através do protocolo de encaminhamento externo BGP. Internamente os ASs utilizam diferentes protocolos de encaminhamento. O cenário a ser emulado na plataforma CORE está ilustrado na Figura 1, representando um potencial cenário existente na Internet envolvendo seis sistemas autónomos.

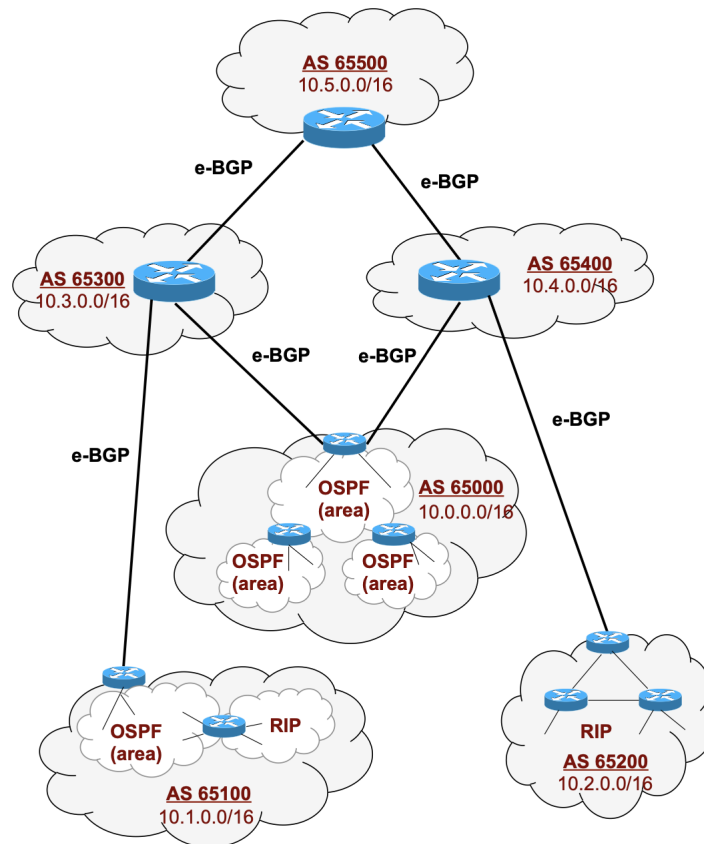


Figura 1: Esquema de Interligação entre vários AS e protocolos internos/externos utilizados

2 Descrição da Resolução do Problema

De forma a resolver o problema proposto que resulta na seguinte topologia da Figura 2, procedemos agora à descrição do trabalho realizado, de forma a justificar as principais decisões tomadas pelo grupo.

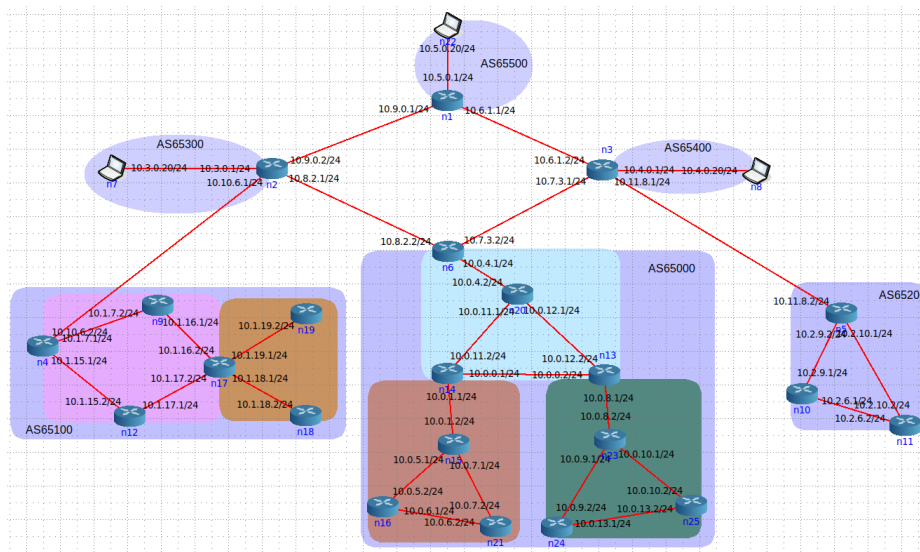


Figura 2: Topologia Resultante

2.1 Conexões entre AS

Pelo enunciado verificamos que existem 6 AS: 65000, 65100, 65200, 65300, 65400, 65500. Em que:

AS 65200

- Sistema autónomo *stub* - mantém relações de *peering* BGP exterior com um sistema autónomo vizinho AS 65400, como é possível verificar pela topologia da Figura 2.
- Usa internamente endereços IPv4 da gama 10.2.0.0/16, como verificado na topologia, tendo este requisito sido tratado de forma manual pelo grupo na configuração dos *routers*.
- Realiza encaminhamento através do protocolo RIP, utilizando rotas por defeito para atingir os outros sistemas autónomos.

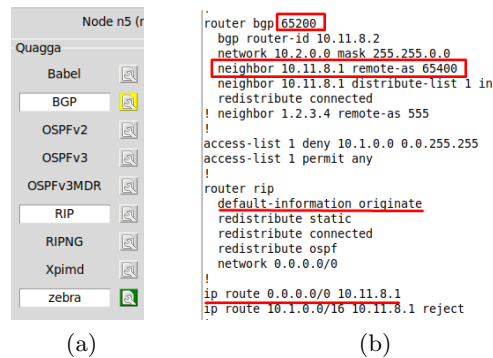


Figura 3: Configuração do *Router* n5 - AS 65200

Tendo em conta a informação relativa a este AS, conseguimos observar a configuração do *router* n5, pertencente a este sistema, tendo ativados os protocolos RIP e BGP, como verificado em a) da Figura 3, e tendo como rota por defeito o *router* n3 e como único sistema autónomo vizinho AS 65400, como se pode verificar em b).

AS 65100

- Sistema autónomo *stub* - mantém relações de *peering* BGP exterior com um sistema autónomo vizinho AS 65300, como é possível verificar pela topologia da Figura 2.
- Usa internamente endereços IPv4 da gama 10.1.0.0/16, como verificado na topologia, tendo este requisito sido tratado de forma manual pelo grupo na configuração dos *routers*.
- Internamente utiliza o protocolo de encaminhamento OSPF. Adicionalmente, existem também algumas redes mais antigas que operam segundo o protocolo RIP. Para conectividade entre redes/equipamentos RIP/OSPF recorre-se a processos de redistribuição de rotas. Internamente são usadas rotas por defeito para atingir os outros sistemas autónomos.

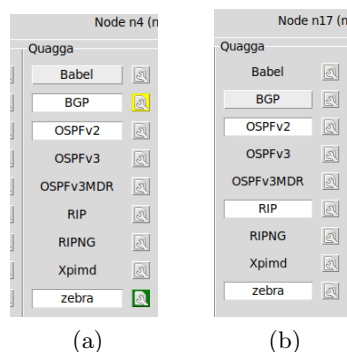


Figura 4: Configuração dos Protocolos para n4 e n17 - AS 65100

De forma a respeitar os requisitos para este sistema, o *router* n17, do AS 65100, foi configurado com os protocolos OSPF e RIP (Figura 4 b)) sendo um *router* que utiliza os dois protocolos, como se verifica na Figura 2.

```

!
router bgp 65100
  bgp router-id 10.10.6.2
  network 10.1.0.0 mask 255.255.0.0
  neighbor 10.10.6.1 remote-as 65300
  neighbor 10.10.6.1 distribute-list 1 in
  redistribute connected
  ! neighbor 1.2.3.4 remote-as 555
!
access-list 1 deny 10.2.0.0 0.0.255.255
access-list 1 permit any
!
router ospf
  default-information originate
  router-id 10.1.15.1
  network 10.1.7.1/24 area 0
  network 10.1.15.1/24 area 0
!
ip route 0.0.0.0/0 10.10.6.1
ip route 10.2.0.0/16 10.10.6.1 reject
!

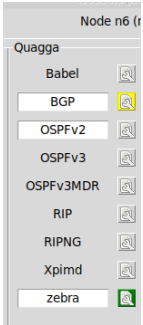
```

Figura 5: Configuração de Rotas por Defeito de n4

Já o *router* n4, também pertencente ao AS 65100, é configurado com os protocolos OSPF e BGP, como comprovado na Figura 4 a), visto que é um *router* que comunica com outros AS, e com a sua própria rede, e, por sua vez, tem uma rota por defeito para n2 e um único vizinho AS 65300 - Figura 5.

AS 65000

- Sistema autónomo *multihomed* - mantém relações de *peering* BGP exterior com dois sistema autónomos vizinhos AS 65300 e AS 65400, como é possível verificar pela topologia da Figura 2.
- Usa internamente endereços IPv4 da gama 10.0.0.0/16, como verificado na topologia, tendo este requisito sido tratado de forma manual pelo grupo na configuração dos *routers*.
- Internamente utiliza o protocolo de encaminhamento OSPF. São usadas rotas por defeito para atingir os outros sistemas autónomos.



(a)

```

router bgp 65000
  bgp router-id 10.8.2.2
  network 10.0.0.0 mask 255.255.0.0
  neighbor 10.8.2.1 remote-as 65300
  neighbor 10.8.2.1 filter-list 1 out
  neighbor 10.7.3.1 remote-as 65400
  neighbor 10.7.3.1 filter-list 1 out
  redistribute connected
  ! neighbor 1.2.3.4 remote-as 555
!

```

(b)

Figura 6: Configuração do *Router* n6 - AS 65000

Novamente se pode verificar pela Figura 6, que o *router* n6, pertencente a este AS, foi configurado com os protocolos OSPF e BGP (Figura 6 a)), tendo, para além disso, como vizinhos os AS 65300 e 65400 - (Figura 6 b)).

```

router ospf
  default-information originate
  router-id 10.0.4.2
  network 10.0.4.2/24 area 0
  network 10.0.11.1/24 area 0
  network 10.0.12.1/24 area 0
!
ip route 0.0.0.0/0 10.0.4.1
!

```

Figura 7: Configuração de Rotas por Defeito de n20

Também se verifica, na Figura 7, que o *router* n20 do AS 65000 tem configurado apenas o protocolo OSPF, pois não necessita de protocolos de encaminhamento externos, e rota por defeito para n6.

AS 65300, AS 65400 e AS 65500

- São essencialmente sistemas autónomos de trânsito. Não sendo estritamente necessário configurar um protocolo de encaminhamento interno em cada um deles.

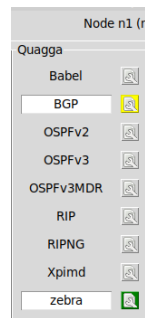


Figura 8: Configuração do *Router* n1 - AS 65500

Como especificado acima, não é estritamente necessário configurar um protocolo de encaminhamento interno, sendo que, para o *router* n1 do AS 65500, Figura 8, é apenas configurado o protocolo BGP.

Por fim, como notas adicionais nesta secção, podemos afirmar, que para que o tráfego de uma rede de um certo AS possa chegar a outra rede de outro AS, é necessário haver uma rota por defeito, e que para além das rotas por defeito apresentadas, é o protocolo BGP que trata de todas as outras. É, também, importante salientar que os *routers* serem configurados com *default-information originate*, permite que sejam comunicadas as rotas internas pelo BGP.

2.2 Áreas

- Internamente o sistema autónomo 65000 utiliza o protocolo de encaminhamento OSPF, estruturado em áreas (com duas áreas, além da área 0, e com três *routers* no mínimo em cada área). Internamente são usadas rotas por defeito para atingir os outros sistemas autónomos.

De seguida, com a conexão feita, são criadas as 3 áreas no AS 65000, sendo que a área 0 é dada pela zona azul, a área 1 pela laranja/castanha e a área 2 pela verde (Figura 2). A área de

backbone necessita de ficar no "meio". Na seguinte figura demonstram-se as configurações no zebra dos *routers* de maior influência.

<pre>router ospf n14 router-id 10.0.11.2 network 10.0.11.2/24 area 0 network 10.0.0.1/24 area 0 network 10.0.1.1/24 area 1</pre>	<pre>router ospf n13 router-id 10.0.12.2 network 10.0.12.2/24 area 0 network 10.0.0.2/24 area 0 network 10.0.8.1/24 area 2</pre>
<pre>router ospf n15 router-id 10.0.1.2 network 10.0.1.2/24 area 1 network 10.0.5.1/24 area 1 network 10.0.7.1/24 area 1</pre>	<pre>router ospf n23 router-id 10.0.8.2 network 10.0.8.2/24 area 2 network 10.0.9.1/24 area 2 network 10.0.10.1/24 area 2</pre>

Figura 9: Configurações de *Routers* de Diferentes Áreas

Consegue-se constatar através da Figura 9 que os *routers* n14 e n13 são *routers* externos, pertencendo a duas áreas, com duas interfaces dentro da área 0 e uma que se conecta às áreas 1 e 2, respetivamente. Já os *routers* n15 e n23 são *routers* internos, que pertencem à área 1 e 2, respetivamente.

2.3 AS65000 não é um Sistema Autónomo de Trânsito

- O sistema autónomo AS 65000 é um sistema *multihomed* mas não um sistema autónomo de trânsito. Ou seja, mesmo que as ligações entre os sistemas autónomos vizinhos AS 65300 e AS 65400 (via AS 65500) falhem, estes não devem ter possibilidade de encaminhar tráfego através do sistema autónomo AS 65000.

<pre>router bgp 65000 bgp router-id 10.8.2.2 network 10.0.0.0 mask 255.255.0.0 neighbor 10.8.2.1 remote-as 65300 neighbor 10.8.2.1 filter-list 1 out neighbor 10.7.3.1 remote-as 65400 neighbor 10.7.3.1 filter-list 1 out redistribute connected ! neighbor 1.2.3.4 remote-as 555 !</pre>	<pre>router ospf router-id 10.8.2.2 network 10.8.2.2/24 area 0 network 10.7.3.2/24 area 0 network 10.0.4.1/24 area 0 ! ip as-path access-list 1 permit ^\$!</pre>
(a)	(b)

Figura 10: Configuração do *Router* n6 - AS 65000

Como podemos observar através da Figura 10, que demonstra as configurações do *router* n6 pertencente ao AS 65000, conseguimos assegurar que este sistema autónomo só irá anunciar prefixos do seu próprio AS [1].

Figura 11: Tabelas de Encaminhamento

Como podemos observar pela tabela de encaminhamento de n6, este *router* tem conhecimento de n2 e de n3. Além disso, podemos observar pelas tabelas de encaminhamento de n3 e n2 que nenhum deles tem um *next hop* que seja o n6. Assim, prova-se que o sistema AS 65000 a que o n6 pertence não é um sistema autónomo de trânsito.

2.4 AS 65200 e AS 65100 Têm Conectividade Assegurada com Todos os Outros ASs exceto Um com o Outro

- O AS 65200 (redes 10.2.0.0/16) tem conectividade assegurada com todos os outros ASs, com exceção do AS 65100 com o qual os administradores do AS 65200 decidiram não ter conectividade.
- O AS 65100 (redes 10.1.0.0/16) tem conectividade assegurada com todos os outros ASs, com exceção do AS 65200 com o qual os administradores do AS 65100 decidiram não ter conectividade.

```
router bgp 65100
  bgp router-id 10.10.6.2
  network 10.1.0.0 mask 255.255.0.0
  neighbor 10.10.6.1 remote-as 65300
  neighbor 10.10.6.1 distribute-list 1 in
  redistribute connected
  ! neighbor 1.2.3.4 remote-as 555
  !
  access-list 1 deny 10.2.0.0 0.0.255.255
  access-list 1 permit any
  !
  router ospf
  default-information originate
  router-id 10.1.15.1
  network 10.1.7.1/24 area 0
  network 10.1.15.1/24 area 0
  !
  ip route 0.0.0.0/0 10.10.6.1
  ip route 10.2.0.0/16 10.10.6.1 reject
  !
```

(a) Configuração do *router* n4

```
router bgp 65200
  bgp router-id 10.11.8.2
  network 10.2.0.0 mask 255.255.0.0
  neighbor 10.11.8.1 remote-as 65400
  neighbor 10.11.8.1 distribute-list 1 in
  redistribute connected
  ! neighbor 1.2.3.4 remote-as 555
  !
  access-list 1 deny 10.1.0.0 0.0.255.255
  access-list 1 permit any
  !
  router rip
  default-information originate
  redistribute static
  redistribute connected
  redistribute ospf
  network 0.0.0.0/0
  !
  ip route 0.0.0.0/0 10.11.8.1
  ip route 10.1.0.0/16 10.11.8.1 reject
  !
```

(b) Configuração do *router* n5

Figura 12

É possível verificar pela Figura 12 que nos *routers* n4 e n5 pertencentes aos AS 65100 e 65200, respetivamente, é feito o comando "*access-list 1 deny*" para os IPs IPv4 da gama do AS 65200 (redes 10.2.0.0/16) e 65100 (redes 10.1.0.0/16), respetivamente. Permitindo acesso a todos os outros AS com o comando "*access-list 1 permit any*".

Fazemos, para além disso, o comando "*ip route (...) reject*" onde rejeitamos o tráfego proveniente do AS 65300 para o caso do n4 e proveniente do AS 65400 para o caso do n5.

Desta forma, asseguramos que o AS 65200 tem conectividade assegurada com todos os outros ASs, com exceção do AS 65100; e que o AS 65100 tem conectividade assegurada com todos os outros ASs, com exceção do AS 65200, como proposto pelo enunciado.

2.5 Testes de Conectividade

Pelas figuras 13, 14 e 15 apresentadas abaixo, conseguimos demonstrar o cumprimento de todos os requisitos apresentados no enunciado deste trabalho.

```
root@n10:/tmp/pycore.36727/n10.conf# ping 10.1.15.2
PING 10.1.15.2 (10.1.15.2) 56(84) bytes of data:
From 10.2.9.2 icmp_seq=1 Destination Host Unreachable
From 10.2.9.2 icmp_seq=2 Destination Host Unreachable
From 10.2.9.2 icmp_seq=3 Destination Host Unreachable
^C
--- 10.1.15.2 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2028ms
root@n10:/tmp/pycore.36727/n10.conf#
```

(a) Ping de n10 para n12

```
root@n12:/tmp/pycore.36727/n12.conf# ping 10.2.9.1
PING 10.2.9.1 (10.2.9.1) 56(84) bytes of data:
From 10.1.15.1 icmp_seq=1 Destination Host Unreachable
From 10.1.15.1 icmp_seq=2 Destination Host Unreachable
From 10.1.15.1 icmp_seq=3 Destination Host Unreachable
^C
--- 10.2.9.1 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2046ms
root@n12:/tmp/pycore.36727/n12.conf#
```

(b) Ping de n12 para n10

Figura 13

```
root@n8:/tmp/pycore.36727/n8.conf# ping 10.5.0.20
PING 10.5.0.20 (10.5.0.20) 56(84) bytes of data:
64 bytes from 10.5.0.20: icmp_seq=1 ttl=62 time=0.061 ms
64 bytes from 10.5.0.20: icmp_seq=2 ttl=62 time=0.044 ms
^C
root@n8:/tmp/pycore.36727/n8.conf# ping 10.3.0.20
PING 10.3.0.20 (10.3.0.20) 56(84) bytes of data:
64 bytes from 10.3.0.20: icmp_seq=1 ttl=61 time=0.036 ms
64 bytes from 10.3.0.20: icmp_seq=2 ttl=61 time=0.047 ms
64 bytes from 10.3.0.20: icmp_seq=3 ttl=61 time=0.052 ms
^C
root@n8:/tmp/pycore.36727/n8.conf# ping 10.0.6.2
PING 10.0.6.2 (10.0.6.2) 56(84) bytes of data:
64 bytes from 10.0.6.2: icmp_seq=1 ttl=59 time=0.104 ms
64 bytes from 10.0.6.2: icmp_seq=2 ttl=59 time=0.061 ms
^C
root@n8:/tmp/pycore.36727/n8.conf# ping 10.0.10.2
PING 10.0.10.2 (10.0.10.2) 56(84) bytes of data:
64 bytes from 10.0.10.2: icmp_seq=1 ttl=59 time=0.074 ms
64 bytes from 10.0.10.2: icmp_seq=2 ttl=59 time=0.057 ms
^C
```

Figura 14: Ping de n8 para n22, n7, n21 e n25

```
root@n7:/tmp/pycore.36727/n7.conf# traceroute 10.2.9.1
traceroute to 10.2.9.1 (10.2.9.1), 30 hops max, 60 byte packets
 1 10.3.0.1 (10.3.0.1) 0.021 ms 0.005 ms 0.004 ms
 2 10.9.0.1 (10.9.0.1) 0.011 ms 0.006 ms 0.006 ms
 3 10.6.1.2 (10.6.1.2) 0.013 ms 0.008 ms 0.007 ms
 4 10.11.8.2 (10.11.8.2) 0.014 ms 0.009 ms 0.009 ms
 5 10.2.9.1 (10.2.9.1) 0.016 ms 0.011 ms 0.011 ms
```

(a) Traceroute de n7 para n10

```
root@n12:/tmp/pycore.36727/n12.conf# traceroute 10.0.5.1
traceroute to 10.0.5.1 (10.0.5.1), 30 hops max, 60 byte packets
 1 10.1.15.1 (10.1.15.1) 0.023 ms 0.006 ms 0.006 ms
 2 10.10.6.1 (10.10.6.1) 0.012 ms 0.006 ms 0.006 ms
 3 10.8.2.2 (10.8.2.2) 0.014 ms 0.007 ms 0.008 ms
 4 10.0.4.2 (10.0.4.2) 0.014 ms 0.009 ms 0.008 ms
 5 10.0.11.2 (10.0.11.2) 0.015 ms 0.011 ms 0.010 ms
 6 10.0.5.1 (10.0.5.1) 0.019 ms 0.019 ms 0.013 ms
root@n12:/tmp/pycore.36727/n12.conf#
```

(b) Traceroute de n12 para n15

Figura 15

3 Conclusão

Como sabido pelo grupo a área de *routing* é uma área de alguma complexidade técnica, e sendo que este TP agrega vários protocolos de *routing* que muitas das vezes acarreta bastante adversidades no *debug* de situações atípicas, este trabalho acabou por levantar algumas dificuldades no desenrolar da sua elaboração.

Não obstante das diversas dificuldades enfrentadas durante este projeto, consideramos que o proposto pelo enunciado foi cumprido, através da dedicação do grupo, e que nos foi possível consolidar conhecimento lecionado em aulas teóricas sobre protocolos de encaminhamento internos/externos, e desenvolver experiências em processos de configuração de protocolos RIP e OSPF, incluindo processos de redistribuição de rotas e em processos de configuração do protocolo externo BGP, incluindo condicionamento/filtragem de rotas BGP.

Referências

1. BGP Prevent Transit AS, <https://networklessons.com/bgp/bgp-prevent-transit-as>.