

TP2: Encaminhamento de Tráfego [Protocolo RIP]

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

Universidade do Minho, Braga, Portugal
Mestrado em Engenharia Informática
Engenharia de Serviços em Rede
Grupo 9
30 de Março de 2022

- 1 **Questão 1 - Definir uma topologia de uma rede de interligação envolvendo vários *routers*, os respectivos *links*, e redes cliente. A topologia criada deverá incluir pelo menos duas situações distintas de redundância: i) na perspectiva de pelo menos um dos *routers* existem pelo menos dois caminhos alternativos com o mesmo custo mínimo para atingir uma determinada rede e ii) na perspectiva de pelo menos um dos *routers* existem pelo menos dois caminhos de custo diferente para atingir uma ou mais redes.**

Tendo em conta o proposto pelo enunciado definimos a seguinte topologia:

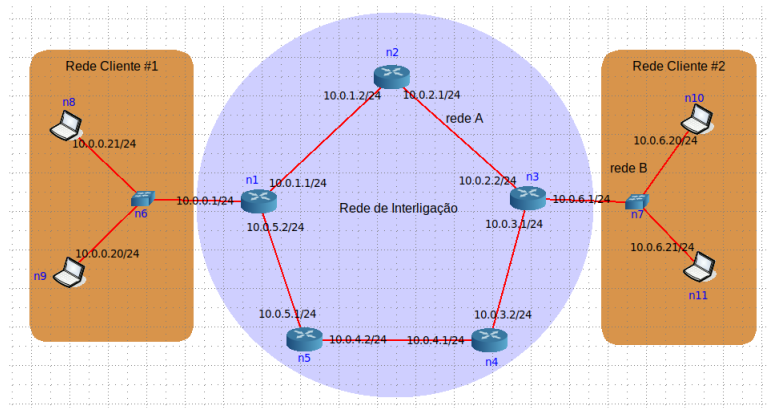


Figura 1: Topologia Criada

Esta topologia, como suposto, contém duas situações distintas de redundância: o *router* n5, por exemplo, tem dois caminhos alternativos com o mesmo custo mínimo para atingir a rede A; e no caso do *router* n5, por exemplo, existem dois caminhos de custo diferente para atingir a rede A.

- 2 **Questão 2 - Analise/comente a configuração dos endereços (IPv4) das interfaces de rede dos *routers*/*hosts* que foi efetuada pelo CORE.**

Como podemos observar na topologia ilustrada na Figura 1, cada dispositivo contém 1 IP para cada uma das suas ligações a outros *routers* e/ou *hosts*, como por exemplo o *router* n1 que contém

3 IPs - 10.0.0.1, 10.0.5.2 e 10.0.1.1, que se ligam, respetivamente aos dispositivos com IPs: 10.0.0.21 ou 10.0.0.20, 10.0.5.1 e 10.0.1.2; a partir desta verificação constata-se que ligações diretas entre dois ou mais dispositivos são consideradas redes, como se verifica atentando nas máscaras de rede dos endereços.

- 3 **Questão 3 - Configure, na interface gráfica, todos os *routers* da topologia para usarem o protocolo RIP para difusão de todas as redes de interligação e redes clientes.**

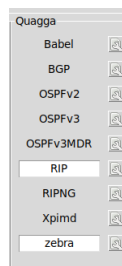


Figura 2: Configuração de cada *router*

Abrindo a configuração de cada *router* onde se pode seleccionar os protocolos usados pelo dispositivo, mantivemos a seleção no Zebra, sendo este um pacote de software de *routing* que fornece serviços de *routing* baseados em TCP/IP com suporte a protocolos de *routing* como RIP, OSPF e BGP, e desativamos os protocolos indesejados. Verificamos que os *routers* utilizam OSPFv2 e OSPFv3, sendo estes protocolos que não tencionamos usar no contexto deste problema, estes são então desativados e, por sua vez, ativamos o protocolo RIP em todos routers de forma a estes usarem este protocolo para difusão de todas as redes de interligação e redes clientes.

- 4 **Questão 4 - Ative a topologia e proceda a testes de conectividade (e.g. ping, etc.) entre os diversos equipamentos. Verifique também as rotas utilizadas pelo tráfego que circula na rede.**

```
root@n8:/tmp/pycore_42783/v6.conf# traceroute 10.0.5.1
traceroute to 10.0.5.1 (10.0.5.1), 30 hops max, 60 byte packets
 1 10.0.0.1 (10.0.0.1) 0.031 ms 0.006 ms 0.003 ms
 2 10.0.0.2 (10.0.0.2) 0.003 ms 0.006 ms 0.006 ms
 3 10.0.5.1 (10.0.5.1) 0.031 ms 0.006 ms 0.007 ms
root@n8:/tmp/pycore_42783/v6.conf#
```

Figura 3: *Traceroute* n8 para n10

Para verificar a conectividade, executámos um **traceroute** do n8 para o n10, verificando também por onde é feita a sua rota.

```

n1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 10.0.0.0/24 is directly connected, eth0
C>* 10.0.1.0/24 is directly connected, eth1
R>* 10.0.2.0/24 [120/2] via 10.0.1.2, eth1, 00:00:55
R>* 10.0.3.0/24 [120/5] via 10.0.1.2, eth1, 00:00:55
R>* 10.0.4.0/24 [120/2] via 10.0.5.1, eth2, 00:00:55
C>* 10.0.5.0/24 is directly connected, eth2
R>* 10.0.6.0/24 [120/3] via 10.0.1.2, eth1, 00:00:55
C>* 127.0.0.0/8 is directly connected, lo
n1#

```

Figura 4: Show ip route - n1

Para complementar, fomos também verificar quais as rotas presentes na tabela de *routing* do n1, verificando que para chegar ao n10 (10.0.6.20), o próximo salto seria o 10.0.1.2, pela interface eth1.

5 Questão 5 -

i) Visualize e explique as tabelas de *routing* que foram estabelecidas pelos *routers* da rede de interligação. [nota: selecione e explique em detalhe uma tabela de *routing* específica que considere relevante para esse efeito]

```

n2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* 10.0.0.0/24 [120/2] via 10.0.1.1, eth0, 00:03:00
C>* 10.0.1.0/24 is directly connected, eth0
C>* 10.0.2.0/24 is directly connected, eth1
R>* 10.0.3.0/24 [120/2] via 10.0.2.2, eth1, 00:03:00
R>* 10.0.4.0/24 [120/3] via 10.0.1.1, eth0, 00:02:58
R>* 10.0.5.0/24 [120/2] via 10.0.1.1, eth0, 00:03:00
R>* 10.0.6.0/24 [120/2] via 10.0.2.2, eth1, 00:03:00
C>* 127.0.0.0/8 is directly connected, lo
n2#

```

Figura 5: Tabela de routing

Existem 3 rotas que estão conectadas diretamente. Duas delas são interfaces para a rede na qual o *router* está, com IP 10.0.1.0/24 e 10.0.2.0/24 e um é o *localhost* com IP 127.0.0.0/8.

Além disso, existem 5 rotas que usam o protocolo RIP, três delas utilizam o interface de saída *eth0* com IP 10.0.0.0, 10.0.4.0 e 10.0.5.0 e duas delas utilizam a interface de saída *eth1*, com IP 10.0.3.0 e 10.0.6.0.

ii) Visualize e explique as tabelas de *routing* dos *hosts* das redes clientes. [nota: explique em detalhe uma tabela de *routing* específica de um dos *hosts*]

```
n1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 10.0.0.0/24 is directly connected, eth0
C>* 10.0.1.0/24 is directly connected, eth1
R>* 10.0.2.0/24 [120/2] via 10.0.1.2, eth1, 00:00:02
R>* 10.0.3.0/24 [120/3] via 10.0.1.2, eth1, 00:00:02
R>* 10.0.4.0/24 [120/2] via 10.0.5.1, eth2, 00:00:02
C>* 10.0.5.0/24 is directly connected, eth2
R>* 10.0.6.0/24 [120/3] via 10.0.1.2, eth1, 00:00:02
C>* 127.0.0.0/8 is directly connected, lo
n1#
```

Figura 6: Tabela de routing

Existem 4 rotas que estão conectadas diretamente. Três delas são interfaces para a rede na qual o *router* está, com IP 10.0.0.0/24, 10.0.1.0/24 e 10.0.5.0/24 e um é o *localhost* com IP 127.0.0.0/8.

Além disso, existem 4 rotas que usam o protocolo RIP, três delas utilizam a interface de saída *eth1* com IP 10.0.2.0, 10.0.3.0 e 10.0.6.0 e uma delas utilizam a interface de saída *eth2*, com IP 10.0.4.0.

6 Questão 6 - Suponha que pretendia alterar o intervalo de tempo segundo o qual são gerados os *updates* enviados pelo protocolo RIP. Que comando utilizaria para o efeito?

Para alterar o intervalo de tempo segundo o qual são gerados *updates* pelo protocolo RIP, os comandos a utilizar primeiramente seriam "*configure terminal*" para entrar em primeiro modo de configuração, seguido de um "*router rip*" para entrar em modo de configuração do protocolo RIP, e finalmente o "*timers basic <<update timer>> <<invalid timer>> <<Holddown timer>> <<Flush timer>>*". Neste último comando, o **Update timer** determina o intervalo de tempo segundo o qual o router envia a informação de rota, o **Invalid Timer** determina o tempo que demora uma entrada até ser colocada em "hold-down state", o **Holddown timer** que mantém o "hold-down state" e o **Flush timer** que é um determina o tempo para remover entradas RIP da tabela de encaminhamento [1].

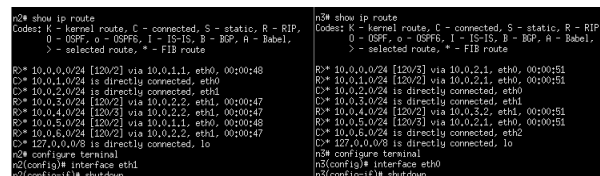
7 Questão 7 - Através da consola de configuração (vtysh) dos *routers* desative um (ou mais) links/interfaces de interligação por forma a originar duas situações distintas:

- apesar da(s) interface(s) desativadas todos os *routers*/redes da topologia podem manter conectividade;
- alguns dos *routers*/redes ficam sem caminhos disponíveis para atingir outros *routers*/redes da topologia.

Para cada uma das situações explique o que observou em relação às alterações das tabelas de *routing* dos equipamentos e ao tempo de propagação da informação de *routing*. [nota: para cada caso deverá ser também selecionada e explicada em detalhe uma tabela de *routing* específica que considere relevante para esse efeito]

7.1 i)

Inicialmente foi desativado o link entre n2 e n3 (correspondendo à rede 10.0.2.0), verificando qual a interface correspondente tanto ao n2, como ao n3, desativando-a no final, como mostra a figura 7



```
n2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

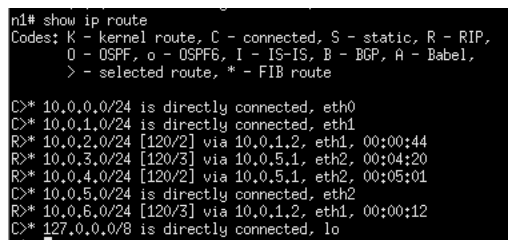
R* 10.0.0.0/24 [120/2] via 10.0.1.1, eth0, 00:00:48
C* 10.0.1.0/24 is directly connected, eth0
C* 10.0.2.0/24 is directly connected, eth1
R* 10.0.5.0/24 [120/2] via 10.0.2.2, eth1, 00:00:47
R* 10.0.4.0/24 [120/3] via 10.0.2.2, eth1, 00:00:47
R* 10.0.5.0/24 [120/2] via 10.0.1.1, eth0, 00:00:48
R* 10.0.5.0/24 [120/2] via 10.0.2.2, eth1, 00:00:47
C* 127.0.0.0/8 is directly connected, lo
n2# configure terminal
n2(config)# interface eth1
n2(config-if)# shutdown

n3# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R* 10.0.0.0/24 [120/3] via 10.0.2.1, eth0, 00:00:51
R* 10.0.1.0/24 [120/2] via 10.0.2.1, eth0, 00:00:51
C* 10.0.2.0/24 is directly connected, eth0
C* 10.0.5.0/24 is directly connected, eth1
R* 10.0.4.0/24 [120/2] via 10.0.3.2, eth1, 00:00:51
R* 10.0.5.0/24 [120/2] via 10.0.2.1, eth0, 00:00:51
C* 10.0.5.0/24 is directly connected, eth2
C* 127.0.0.0/8 is directly connected, lo
n3# configure terminal
n3(config)# interface eth0
n3(config-if)# shutdown
```

Figura 7: Comandos para desativar link entre n2 e n3

Posteriormente à desativação deste link, fizemos uma comparação tanto da tabela de *routing* presente em n1, dada pelas figuras 8 e 9, como de um trace route do n8 para o n10, dada pela imagem 9.



```
n1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 10.0.0.0/24 is directly connected, eth0
C>* 10.0.1.0/24 is directly connected, eth1
R>* 10.0.2.0/24 [120/2] via 10.0.1.2, eth1, 00:00:44
R>* 10.0.3.0/24 [120/3] via 10.0.5.1, eth2, 00:04:20
R>* 10.0.4.0/24 [120/2] via 10.0.5.1, eth2, 00:05:01
C>* 10.0.5.0/24 is directly connected, eth2
R>* 10.0.6.0/24 [120/3] via 10.0.1.2, eth1, 00:00:12
C>* 127.0.0.0/8 is directly connected, lo
```

Figura 8: Tabela de *routing* antes da desativação do link



```
n1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 10.0.0.0/24 is directly connected, eth0
C>* 10.0.1.0/24 is directly connected, eth1
R>* 10.0.3.0/24 [120/3] via 10.0.5.1, eth2, 00:00:49
R>* 10.0.4.0/24 [120/2] via 10.0.5.1, eth2, 00:01:30
C>* 10.0.5.0/24 is directly connected, eth2
R>* 10.0.6.0/24 [120/4] via 10.0.5.1, eth2, 00:00:49
C>* 127.0.0.0/8 is directly connected, lo
n1#
```

Figura 9: Tabela de *routing* depois da desativação do link

Em relação à tabela de *routing*, antes e depois, conseguimos notar facilmente que desapareceu a informação da rota até 10.0.2.0 (link que foi desativado) e, além disso, houve mudança na rota que tem como destino a rede 10.0.6.0, que passou de ir pela interface eth1, para ir pela interface eth2. Isto deve-se ao facto da rota utilizada anteriormente, ter sido comprometida pela desativação de um link, não conseguindo mais chegar ao seu destino; tendo de optar pela outra rota disponível, que é feita através da interface eth2.

```

root@n8:/tmp/pycore.42783/n8.conf# traceroute 10.0.6.20
traceroute to 10.0.6.20 (10.0.6.20): 30 hops max, 50 byte packets
 1 10.0.0.1 (10.0.0.1) 0.034 ns 0.004 ns 0.003 ns
 2 10.0.1.2 (10.0.1.2) 0.010 ns 0.006 ns 0.005 ns
 3 10.0.2.2 (10.0.2.2) 0.002 ns 0.007 ns 0.007 ns
 4 10.0.6.20 (10.0.6.20) 0.022 ns 0.010 ns 0.009 ns
root@n8:/tmp/pycore.42783/n8.conf# traceroute 10.0.6.20
traceroute to 10.0.6.20 (10.0.6.20): 30 hops max, 50 byte packets
 1 10.0.0.1 (10.0.0.1) 0.017 ns 0.004 ns 0.003 ns
 2 10.0.3.1 (10.0.3.1) 0.010 ns 0.006 ns 0.006 ns
 3 10.0.4.1 (10.0.4.1) 0.010 ns 0.006 ns 0.006 ns
 4 10.0.3.1 (10.0.3.1) 0.013 ns 0.008 ns 0.008 ns
 5 10.0.6.20 (10.0.6.20) 0.015 ns 0.011 ns 0.011 ns
root@n8:/tmp/pycore.42783/n8.conf#

```

Figura 10: traceroute do n8 para o n10

Relativamente à figura 10, conseguimos notar claramente uma diferença na rota em que é utilizada, passando até de uma rota de 4 saltos, para uma de 5.

7.2 ii)

De seguida, decidimos desativar o link entre n4 e n5, fazendo por isso com que não existisse rota entre a **rede Cliente #1** e a **rede Cliente #2**. Assim começámos por desativar o link, como mostra a figura 11, verificando a inexistência de conectividade através da verificação da tabela de *routing* em n1, e de um ping feito por n8 até n10 (figuras 12 e 13).

```

n1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* 10.0.0.0/24 [120/3] via 10.0.4.2, eth1, 00:18:45
R>* 10.0.1.0/24 [120/3] via 10.0.4.2, eth1, 00:17:34
C>* 10.0.3.0/24 is directly connected, eth0
C>* 10.0.4.0/24 is directly connected, eth1
R>* 10.0.5.0/24 [120/2] via 10.0.4.2, eth1, 00:18:45
R>* 10.0.6.0/24 [120/2] via 10.0.3.1, eth0, 00:18:45
C>* 127.0.0.0/8 is directly connected, lo
n1# configure terminal
n1(config)# interface eth1
n1(config-if)# shutdown
n1(config-if)#

```

```

n5# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* 10.0.0.0/24 [120/2] via 10.0.5.2, eth1, 00:18:50
R>* 10.0.1.0/24 [120/2] via 10.0.5.2, eth1, 00:18:50
R>* 10.0.3.0/24 [120/2] via 10.0.4.1, eth0, 00:18:50
C>* 10.0.4.0/24 is directly connected, eth0
C>* 10.0.5.0/24 is directly connected, eth1
R>* 10.0.6.0/24 [120/3] via 10.0.4.1, eth0, 00:18:48
C>* 127.0.0.0/8 is directly connected, lo
n5# configure terminal
n5(config)# interface eth0
n5(config-if)# shutdown
n5(config-if)#

```

Figura 11: Comandos para desativar link entre n4 e n5

```

n1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 10.0.0.0/24 is directly connected, eth0
C>* 10.0.1.0/24 is directly connected, eth1
C>* 10.0.5.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
n1#

```

Figura 12: Tabela de *routing* depois da desativação dos 2 links

```

root@n8:/tmp/pycore.42783/n8.conf# ping 10.0.6.20
PING 10.0.6.20 (10.0.6.20) 56(84) bytes of data:
From 10.0.0.1 icmp_seq=1 Destination Net Unreachable
From 10.0.0.1 icmp_seq=2 Destination Net Unreachable
From 10.0.0.1 icmp_seq=3 Destination Net Unreachable
^C
--- 10.0.6.20 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2037ms
root@n8:/tmp/pycore.42783/n8.conf#

```

Figura 13: Ping feito por n8 até n10

8 Configure um dos *routers* da sua topologia por forma a que o tráfego que ele transmite para uma determinada rede destino não passe pelo caminho com um menor numero de saltos, mas sim por um outro caminho alternativo [sugestão: nas aulas teóricas foi mencionado um comando específico que permite resolver esta questão]

Primeiramente, tivemos de criar uma *access list* para posteriormente ser feita uma alteração da métrica. Esta alteração foi feita entrando em modo configuração do RIP, e alterando a métrica de default para 7, no nosso caso, como mostram as figuras 14 e 15.

```

n2# configure terminal
n2(config)# access-list
<1-99> IP standard access list
<100-199> IP extended access list
<1300-1999> IP standard access list (expanded range)
<2000-2699> IP extended access list (expanded range)
WORD IP zebra access-list
n2(config)# access-list 1
remark Access list entry comment
deny Specify packets to reject
permit Specify packets to forward
n2(config)# access-list 1 permit
A,B,C,D Address to match
any Any source host
host A single host address
n2(config)# access-list 1 permit any
<cr>
n2(config)# access-list 1 permit any

```

Figura 14: Criação da Access List

```

n2(config)# router rip
n2(config-router)# offset-list
WORD Access-list name
n2(config-router)# offset-list 1
in For incoming updates
out For outgoing updates
n2(config-router)# offset-list 1 out
<0-16> Metric value
n2(config-router)# offset-list 1 out 7
<cr>
IFNAME Interface to match
n2(config-router)# offset-list 1 out 7 eth1
<cr>
n2(config-router)# offset-list 1 out 7 eth1

```

Figura 15: Alteração da métrica

Seguidamente verificámos a tabela de *routing* de n3, e executámos um *traceroute* do n10, para o n8, confirmando que da rede Cliente #2 para a rede Cliente #1, a rota utilizada não é a mesma após a modificação da métrica (figuras 16 e 17).

```

n3# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* 10.0.0.0/24 [120/4] via 10.0.3.2, eth1, 00:01:20
R>* 10.0.1.0/24 [120/4] via 10.0.3.2, eth1, 00:01:20
C>* 10.0.2.0/24 is directly connected, eth0
C>* 10.0.3.0/24 is directly connected, eth1
R>* 10.0.4.0/24 [120/2] via 10.0.3.2, eth1, 00:38:07
R>* 10.0.5.0/24 [120/3] via 10.0.3.2, eth1, 00:07:59
C>* 10.0.6.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
n3#

```

Figura 16: Tabela de routing de n3

```

root@n10:/tmp/pycore.40637/n10.conf# traceroute 10.0.0.21
traceroute to 10.0.0.21 (10.0.0.21), 30 hops max, 60 byte packets
 1 10.0.6.1 (10.0.6.1) 0.022 ms 0.004 ms 0.003 ms
 2 10.0.3.2 (10.0.3.2) 0.012 ms 0.006 ms 0.006 ms
 3 10.0.4.2 (10.0.4.2) 0.013 ms 0.007 ms 0.007 ms
 4 10.0.1.1 (10.0.1.1) 0.016 ms 0.008 ms 0.008 ms
 5 10.0.0.21 (10.0.0.21) 0.015 ms 0.011 ms 0.011 ms
root@n10:/tmp/pycore.40637/n10.conf#

```

Figura 17: Traceroute n10 para n8

Pelas imagens, conseguimos verificar que todo o tráfego dirigido a *10.0.0.0* (rede Cliente #1), é feito através da interface eth1, podendo confirmar que isto acontece realmente através do *traceroute*, onde conseguimos realmente ver por onde o tráfego passa.

- 9 Assuma que a rede de interligação que definiu passará a estar ligada a uma outra rede externa através do *router* x (ver Figura 16). Apresente e explique o(s) comando(s) que usaria na sua rede de interligação para que todo o tráfego dirigido a redes externas saísse pelo *router* n5.

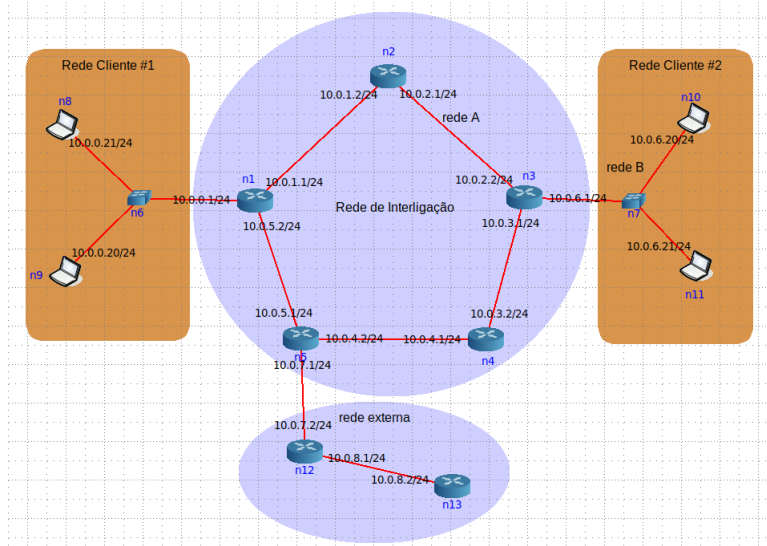


Figura 18: Topologia modificada com uma rede externa

Para que todo o tráfego dirigido a redes externas passe pelo *router* n5, é preciso definir uma rota por defeito em todos os *routers* para que, não só o n5, mas todos os outros "saibam" por onde têm de encaminhar o tráfego. Isto é, o n3 se quiser enviar tráfego para uma rede externa, tem de "saber" que esse encaminhamento é feito através do n5. Assim, os comandos que o grupo usou para permitir a comunicação com uma rede externa, foi o comando para a criação de uma rota por defeito no *router* n5 ("ip route <<IP destination prefix>> <<IP destination prefix mask>> <<IP gateway address>>"), e finalmente o comando para anunciar as rotas RIP ("default-information originate"), como mostra a figura 19.

Depois da execução destes comandos, o grupo decidiu verificar se as rotas foram anunciadas corretamente examinando a tabela de *routing* do *router* n2 (figura 20).


```

n5# configure terminal
n5(config)# ip route 0.0.0.0 0.0.0.0 10.0.7.2
n5(config)# exit
n5# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 10.0.7.2, eth2
R>* 10.0.0.0/24 [120/2] via 10.0.5.1, eth1, 00:00:23
R>* 10.0.1.0/24 [120/2] via 10.0.5.1, eth1, 00:00:23
R>* 10.0.2.0/24 [120/3] via 10.0.5.1, eth1, 00:00:23
R>* 10.0.3.0/24 [120/2] via 10.0.4.1, eth0, 00:00:23
C>* 10.0.4.0/24 is directly connected, eth0
C>* 10.0.5.0/24 is directly connected, eth1
R>* 10.0.6.0/24 [120/3] via 10.0.4.1, eth0, 00:00:22
C>* 10.0.7.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
n5# configure terminal
n5(config)# router rip
n5(config-router)# default-information originate
n5(config-router)#

```

Figura 19: Comandos para a inserção de rota por defeito no router n5

```

n2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

R>* 0.0.0.0/0 [120/3] via 10.0.1.1, eth0, 00:01:07
R>* 10.0.0.0/24 [120/2] via 10.0.1.1, eth0, 00:01:24
C>* 10.0.1.0/24 is directly connected, eth0
C>* 10.0.2.0/24 is directly connected, eth1
R>* 10.0.3.0/24 [120/2] via 10.0.2.2, eth1, 00:01:24
R>* 10.0.4.0/24 [120/3] via 10.0.2.2, eth1, 00:01:24
R>* 10.0.5.0/24 [120/2] via 10.0.1.1, eth0, 00:01:24
R>* 10.0.6.0/24 [120/2] via 10.0.2.2, eth1, 00:01:24
R>* 10.0.7.0/24 [120/3] via 10.0.1.1, eth0, 00:01:24
C>* 127.0.0.0/8 is directly connected, lo
n2#

```

Figura 20: Tabela de *routing* do router n2

- 10 Após os procedimentos efetuados na questão 9, defina um cenário de rede que lhe permita verificar, e.g. através dos utilitários *ping* e *traceroute*, que os *routers* da topologia que definiu conseguem efetivamente conectividade a endereços externos (i.e. localizados em Outras Redes na Figura 16) através do *router* n5

Após os passos feitos na pergunta 8, as redes cliente #1 e #2 conseguiam enviar tráfego para o exterior, mas não o conseguiam receber, não sendo possível executar *pings* ou *traceroutes*, por exemplo. Assim, é preciso fazer o mesmo que foi feito para o *router* n5, no *router* n12, não esquecendo que a rede externa utiliza OSPF (figura 21).

Após da execução desses comandos, as redes clientes conseguem finalmente comunicar com a rede externa, como mostra a figura 22.

```

n12# configure terminal
n12(config)# ip route 0.0.0.0 0.0.0.0 10.0.7.1
n12(config)# exit
n12# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, o - OSPF6, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 10.0.7.1, eth0
o 10.0.7.0/24 [110/1] is directly connected, eth0, 00:05:29
O 10.0.7.0/24 [110/10] is directly connected, eth0, 00:05:29
C>* 10.0.7.0/24 is directly connected, eth0
o 10.0.8.0/24 [110/1] is directly connected, eth1, 00:05:29
O 10.0.8.0/24 [110/10] is directly connected, eth1, 00:05:29
C>* 10.0.8.0/24 is directly connected, eth1
C>* 127.0.0.0/8 is directly connected, lo
n12# configure terminal
n12(config)# router ospf
n12(config-router)# default-information originate
n12(config-router)#

```

Figura 21: Comandos para a inserção de rota por defeito no *router* n12

```

root@n8:/tmp/pycore.40637/n8.conf# ping 10.0.7.2 n12
PING 10.0.7.2 (10.0.7.2) 56(84) bytes of data.
64 bytes from 10.0.7.2: icmp_seq=1 ttl=62 time=0.035 ms
64 bytes from 10.0.7.2: icmp_seq=2 ttl=62 time=0.041 ms
64 bytes from 10.0.7.2: icmp_seq=3 ttl=62 time=0.111 ms
^C
--- 10.0.7.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 0.035/0.062/0.111/0.034 ms
root@n8:/tmp/pycore.40637/n8.conf# ping 10.0.8.2 n13
PING 10.0.8.2 (10.0.8.2) 56(84) bytes of data.
64 bytes from 10.0.8.2: icmp_seq=1 ttl=61 time=0.042 ms
64 bytes from 10.0.8.2: icmp_seq=2 ttl=61 time=0.046 ms
64 bytes from 10.0.8.2: icmp_seq=3 ttl=61 time=0.058 ms
^C
--- 10.0.8.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.042/0.048/0.058/0.006 ms
root@n8:/tmp/pycore.40637/n8.conf#

```

Figura 22: *Pings* executados para ambos os *routers* na rede externa

Referências

1. N Study, <https://www.n-study.com/en/iprouting/rip-timer/>.