

# Trabalho TP2: Encaminhamento de Tráfego [Protocolo RIP]

Beatriz Almeida<sup>[53693]</sup>, Miguel Gomes<sup>[54153]</sup>, and Rui Armada<sup>[50737]</sup>

Universidade do Minho, Braga 4710 - 057, Portugal  
Rua da universidade, Braga, Portugal  
<https://www.uminho.pt/PT>

**Abstract.** Abstract aqui

**Keywords:** CORE · RIP · Routing

## 1 Introdução

Este projeto foi-nos atribuído como uma introdução prática aos processos de configuração dos protocolos de encaminhamento RIP. Devido a esta natureza do trabalho foram-nos propostas algumas tarefas que tivémos de resolver utilizando os conhecimentos teóricos que fomos acumulando com o decorrer das aulas teóricas, assim como a experiência que já temos de anos anteriores para dar uso à ferramenta do CORE.

Tivémos ainda de apurar as nossas capacidades de pesquisa/(auto)aprendizagem para configuração de protocolos de encaminhamento de tráfego em redes IP, uma vez que nem todas as tarefas podiam ser resolvidas apenas com os conhecimentos pré-adquiridos.

## 2 Tarefas

- 2.1 Definir uma topologia de uma rede de interligação envolvendo vários routers, os respectivos links, e redes cliente.**

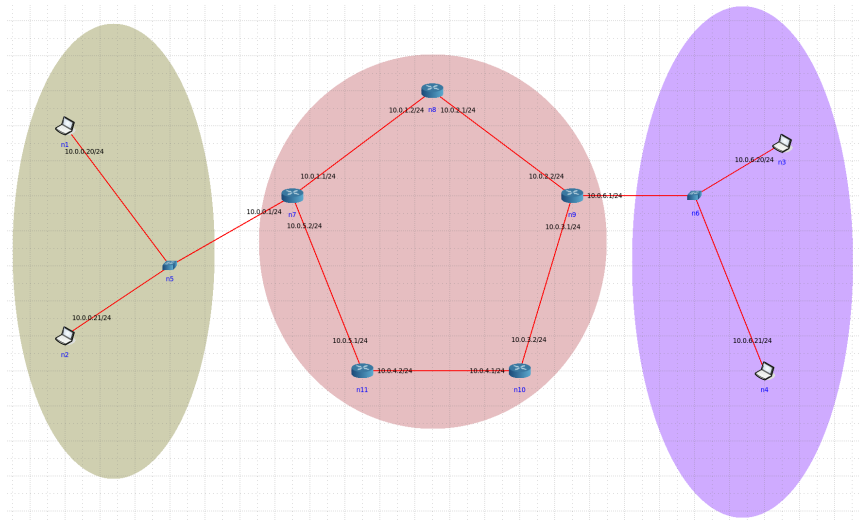


Fig. 1: Topologia desenvolvida

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

Com base na Figura 1, é possível afirmar que existem 7 sub-redes de Classe A distintas, cada uma delas contendo a sua faixa de valores de IP dentro desta classe. Para 7 sub-redes, é necessário pelo menos usar 3 bits para criar o número necessário de sub-redes. Com 3 bits, é necessário ter  $2^3 = 8$  sub-redes disponíveis, existindo espaço para futuras expansões da rede.

A máscara de sub-rede para 3 bits emprestados seria 255.255.255.224, o que corresponde a uma máscara de 255.255.255.11100000 em binário. Então, a máscara de sub-rede que se usaria para dividir a rede em 7 sub-redes seria 255.255.255.224.

- 2.3** 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.

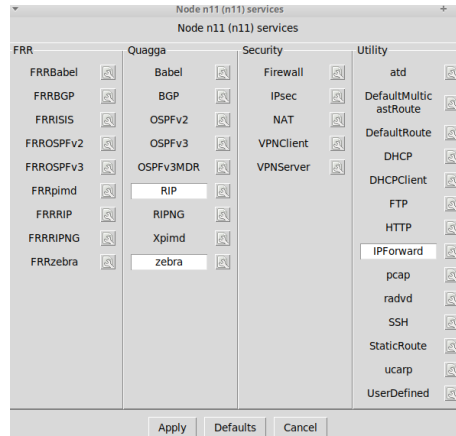


Fig. 2: Configuração do router

O objetivo era fazer com que todos os routers da topologia usassem o protocolo RIP. Para atingir esse objetivo, as opções OSPFv2 e OSPFv3 foram desativadas e, em seguida, a opção RIP foi ativada.

- 2.4** 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@n3:/tmp/pycore.42185/n3.conf# ping 10.0.0.20
PING 10.0.0.20 (10.0.0.20) 56(84) bytes of data:
64 bytes from 10.0.0.20: icmp_seq=1 ttl=61 time=0,055 ms
64 bytes from 10.0.0.20: icmp_seq=2 ttl=61 time=0,054 ms
64 bytes from 10.0.0.20: icmp_seq=3 ttl=61 time=0,050 ms
64 bytes from 10.0.0.20: icmp_seq=4 ttl=61 time=0,073 ms
^C
[1]+  Stopped                  ping 10.0.0.20
root@n3:/tmp/pycore.42185/n3.conf#

```

(a) Ping de N4 para N1.

```

root@n3:/tmp/pycore.42185/n3.conf# traceroute 10.0.6.1
traceroute to 10.0.6.1 (10.0.6.1), 30 hops max, 60 byte packets
 1 10.0.6.1 (10.0.6.1) 0,096 ms 0,005 ms 0,003 ms
root@n3:/tmp/pycore.42185/n3.conf# traceroute 10.0.2.1
traceroute to 10.0.2.1 (10.0.2.1), 30 hops max, 60 byte packets
 1 10.0.6.1 (10.0.6.1) 0,088 ms 0,006 ms 0,003 ms
 2 10.0.2.1 (10.0.2.1) 0,014 ms 0,006 ms 0,005 ms
root@n3:/tmp/pycore.42185/n3.conf#

```

(b) Traceroute de N9 e N8, respectivamente

Fig. 3: Testes de conectividade

## 2.5

```
n7# 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, 00:00:43
R>* 10.0.4.0/24 [120/2] via 10.0.5.1, eth2, 00:00:43
C>* 10.0.5.0/24 is directly connected, eth2
R>* 10.0.6.0/24 [120/3] via 10.0.1.2, eth1, 00:00:39
C>* 127.0.0.0/8 is directly connected, lo
n7#
```

(a) Tabela de Routing do Router N7

```
n8# 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, 00:01:19
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, 00:01:19
R>* 10.0.4.0/24 [120/3] via 10.0.2.2, eth1, 00:01:19
R>* 10.0.5.0/24 [120/2] via 10.0.1.1, eth0, 00:01:19
R>* 10.0.6.0/24 [120/2] via 10.0.2.2, eth1, 00:01:19
C>* 127.0.0.0/8 is directly connected, lo
n8#
```

(b) Tabela de Routing do Router N8

```
n9# 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, 00:01:39
R>* 10.0.1.0/24 [120/2] via 10.0.2.1, eth0, 00:01:40
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, 00:01:39
R>* 10.0.5.0/24 [120/3] via 10.0.2.1, eth0, 00:01:39
C>* 10.0.6.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
n9#
```

(c) Tabela de Routing do Router N9

```
n10# 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 10.0.4.2, eth1, 00:02:02
R>* 10.0.1.0/24 [120/3] via 10.0.3.1, eth0, 00:02:03
R>* 10.0.2.0/24 [120/2] via 10.0.3.1, eth0, 00:02:03
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, 00:02:02
R>* 10.0.6.0/24 [120/2] via 10.0.3.1, eth0, 00:02:03
C>* 127.0.0.0/8 is directly connected, lo
n10#
```

(d) Tabela de Routing do Router N10

```
n11# 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 10.0.5.2, eth1, 00:02:32
R>* 10.0.1.0/24 [120/2] via 10.0.5.2, eth1, 00:02:32
R>* 10.0.2.0/24 [120/3] via 10.0.5.2, eth1, 00:02:32
R>* 10.0.3.0/24 [120/2] via 10.0.4.1, eth0, 00:02:32
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, 00:02:32
C>* 127.0.0.0/8 is directly connected, lo
n11#
```

(e) Tabela de Routing do Router N11

Fig. 4: Tabelas de Routing

i) **Visualize e explique as tabelas de routing que foram estabelecidas pelos routers da rede de interligação.** Como podemos ver na Figura 4a, as tabelas de *routing* são compostas por dois tipos de entradas: entradas derivadas de interfaces conectadas, como a sub-rede 10.0.0.0/24 estar conectada diretamente através da interface `eth0`; e entradas preenchidas usando o protocolo RIP, como é o caso da sub-rede 10.0.2.0/24 que apenas é acessível através de um salto por 10.0.1.2 que se encontra na interface `eth1`.

ii) **Visualize e explique as tabelas de routing dos hosts das redes clientes** Quanto às tabelas de *hosts*, podemos ver que estas apenas contém duas entradas que representam a rede local (0.0.0.0) e o *Gateway* (10.0.x.0) da rede.

```

root@n1:/tmp/pycore.33319/n1.conf# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0,0,0,0 10,0,0,1 0,0,0,0 UG 0 0 0 eth0
10,0,0,0 0,0,0,0 255,255,255,0 U 0 0 0 eth0
root@n1:/tmp/pycore.33319/n1.conf#

```

(a) Tabela de Hosts N1

```

root@n2:/tmp/pycore.33319/n2.conf# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0,0,0,0 10,0,0,1 0,0,0,0 UG 0 0 0 eth0
10,0,0,0 0,0,0,0 255,255,255,0 U 0 0 0 eth0
root@n2:/tmp/pycore.33319/n2.conf#

```

(b) Tabela de Hosts N2

```

root@n3:/tmp/pycore.33319/n3.conf# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0,0,0,0 10,0,6,1 0,0,0,0 UG 0 0 0 eth0
10,0,6,0 0,0,0,0 255,255,255,0 U 0 0 0 eth0
root@n3:/tmp/pycore.33319/n3.conf#

```

(c) Tabela de Hosts N3

```

root@n4:/tmp/pycore.33319/n4.conf# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0,0,0,0 10,0,6,1 0,0,0,0 UG 0 0 0 eth0
10,0,6,0 0,0,0,0 255,255,255,0 U 0 0 0 eth0
root@n4:/tmp/pycore.33319/n4.conf#

```

(d) Tabela de Hosts N4

Fig. 5: Tabelas de Hosts

## 2.6 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?

O comando a ser utilizado deve ser:

```
timers basic {update} {timeout} {garbage}
```

Onde:

- **{update} (default: 30s)**: Define após quantos segundos o processo do RIP deve ser 'acordado' para enviar uma mensagem de resposta não solicitada a todos os vizinhos do *router*, que contém a sua tabela de *routing* completa;
- **{timeout} (default: 180s)**: Define após quantos segundos uma determinada rota deve ser marcada como já não sendo válida; no entanto, esta rota é retida na tabela de *routing* durante um curto intervalo de tempo, de modo a que todos os seus vizinhos possam ser notificados sobre a mudança;
- **{garbage} (default: 120s)**: Define a partir de quantos segundos uma rota marcada como inválida deve ser removida da tabela de *routing*.

Este comando foi retirado da documentação online do Quagga [1].

## 2.7 Através da consola de configuração (vtysh) dos routers desative um (ou mais) links/interfaces de interligação por forma a originar duas situações distintas. 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.

i) apesar da(s) interface(s) desativadas todos os routers/redes da topologia podem manter conectividade Primeiramente, decidimos fazer um *traceroute* inicial do Host *n1* para o Host *n3* de forma a saber que caminho estava a ser utilizado para entregar os pacotes ao destino.

```

root@n1:/tmp/pycore.45117/n1.conf# traceroute 10.0.6.20
traceroute to 10.0.6.20 (10.0.6.20), 30 hops max, 60 byte packets
 1 10.0.0.1 (10.0.0.1) 0.058 ms 0.004 ms 0.004 ms
 2 10.0.1.2 (10.0.1.2) 0.017 ms 0.006 ms 0.005 ms
 3 10.0.2.2 (10.0.2.2) 0.017 ms 0.007 ms 0.007 ms
 4 10.0.6.20 (10.0.6.20) 0.031 ms 0.010 ms 0.010 ms
root@n1:/tmp/pycore.45117/n1.conf# █

```

Fig. 6: Traceroute Inicial do Host N1 para o Host N3

Após confirmar que a rota tomada, decidimos remover o *link* entre os Routers *n7* e *n8*, o que calculámos que fosse criar uma mudança de rota, onde os pacotes teriam de passar pelos Routers *n11*, *n10* e *n9*, por essa ordem, para chegar ao destino (*n3*).

Para tal, começámos por abrir um terminal **vttysh** no Router *n7* e utilizámos o comando **show int** para verificar qual das interfaces do router é que fazia a ligação com o router *n8*.

```

n7# show int
% Unknown command.
n7# show int
Interface eth0 is up, line protocol detection is disabled
  index 9 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:00:00:aa:00:02
  inet 10.0.0.1/24
  inet6 2001::1/64
  inet6 fe80::200:ff:feaa:2/64
Interface eth1 is up, line protocol detection is disabled
  index 12 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:00:00:aa:00:03
  inet 10.0.1.1/24
  inet6 2001:1::1/64
  inet6 fe80::200:ff:feaa:3/64
Interface eth2 is up, line protocol detection is disabled
  index 34 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:00:00:aa:00:0c
  inet 10.0.5.2/24
  inet6 2001:5::2/64
  inet6 fe80::200:ff:feaa:c/64
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
n7# █

```

Fig. 7: Verificação das Interfaces do Router *n7*

Como podemos ver acima esta ligação é feita através da interface **eth1**.

De seguida, escrevemos os seguintes comandos, para desativar a interface escolhida e para ver a nova configuração atual:

```
n7# conf t
n7(config)# int eth1
n7(config-if)# shutdown
n7(config-if)# exit
n7(config)# exit
n7# write t
Building configuration...

Current configuration:
!
!
service integrated-vtysh-config
!
interface eth0
 ip address 10.0.0.1/24
!
interface eth1
 ip address 10.0.1.1/24
 shutdown
!
interface eth2
 ip address 10.0.5.2/24
!
router rip
 redistribute connected
 redistribute static
 network 0.0.0.0/0
!
ip forwarding
ipv6 forwarding
!
line vty
!
end
n7# ^C
```

Fig. 8: Desativação da Interface *eth1* do Router *n7*

Resta-nos, portanto, copiar essa nova configuração e copiá-la para o ficheiro de configuração do serviço **zebra** (que é necessário porque o CORE não permite efetuar o comando **mem write**). Temos, então, que primeiro desligar a sessão atual, para podermos aceder à configuração dos serviços no router *n7* e alterar a configuração do **zebra**.

Por fim, voltamos a correr a sessão, já com as novas configurações guardadas e abrimos um terminal **vttysh** no router *n7* para correr novamente o comando **show int** e verificar o estado atual da interface **eth1**.

```

n7# show int
Interface eth0 is up, line protocol detection is disabled
  index 53 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:00:00:aa:00:02
  inet 10.0.0.1/24
  inet6 2001::1/64
  inet6 fe80::200:ff:feaa:2/64
Interface eth1 is down
  index 56 metric 1 mtu 1500
  flags: <BROADCAST,MULTICAST>
  HWaddr: 00:00:00:aa:00:03
  inet 10.0.1.1/24
  inet6 2001:1::1/64
Interface eth2 is up, line protocol detection is disabled
  index 78 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 00:00:00:aa:00:0c
  inet 10.0.5.2/24
  inet6 2001:5::2/64
  inet6 fe80::200:ff:feaa:c/64
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
n7#
n7# 

```

Fig. 9: Confirmação de que a interface *eth1* foi desativada com sucesso

Finalmente, podemos ver como a perda deste *link* afetou a rota que os pacotes tomaram para chegar ao destino:

```

root@n1:/tmp/pycore.45117/n1.conf# traceroute 10.0.6.20
traceroute to 10.0.6.20 (10.0.6.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.059 ms  0.004 ms  0.004 ms
 2  10.0.5.1 (10.0.5.1)  0.018 ms  0.006 ms  0.005 ms
 3  10.0.4.1 (10.0.4.1)  0.016 ms  0.007 ms  0.006 ms
 4  10.0.3.1 (10.0.3.1)  0.016 ms  0.009 ms  0.008 ms
 5  10.0.6.20 (10.0.6.20)  0.034 ms  0.010 ms  0.010 ms
root@n1:/tmp/pycore.45117/n1.conf# 

```

Fig. 10: Traceroute Final do Host N1 para o Host N3

Tal como previsto, o caminho foi alterado e passou-se a utilizar a rota que passa pelo Router *n11*.

ii) **alguns dos routers/redes ficam sem caminhos disponíveis para atingir outros routers/redes da topologia.** Para esta fase decidimos aproveitar a configuração da fase anterior, o que nos fez



decidir que o próximo *link* a desativar deveria ser o *link* que faz a ligação entre o Router *n7* e o Router *n11*, isolando, assim, a sub-rede da esquerda (ver Topologia).

Seguimos os mesmos passos da fase anterior relativos a verificar as interfaces pertencentes ao Router *n7*, à desativação da ligação escolhida (neste caso foi a interface **eth2**) e à cópia da nova configuração para a configuração do serviço **zebra**.

Após um rápido reinício da sessão, confirmamos que as alterações foram concluídas com sucesso:

```
n7# show int
Interface eth0 is up, line protocol detection is disabled
  index 97 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Hwaddr: 00:00:00:aa:00:02
  inet 10.0.0.1/24
  inet6 2001::1/64
  inet6 fe80::200:ff:feaa:2/64
Interface eth1 is down
  index 100 metric 1 mtu 1500
  flags: <BROADCAST,MULTICAST>
  Hwaddr: 00:00:00:aa:00:03
  inet 10.0.1.1/24
  inet6 2001:1::1/64
Interface eth2 is down
  index 122 metric 1 mtu 1500
  flags: <BROADCAST,MULTICAST>
  Hwaddr: 00:00:00:aa:00:0c
  inet 10.0.5.2/24
  inet6 2001:5::2/64
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
n7#
```

Fig. 11: Confirmação de que ambas as interfaces *eth1* e *eth2* foram desativadas.

Devido à perda deste *link* é impossível estabelecer uma ligação entre os Hosts *n1* e *n3*, pelo que o resultado de um comando **ping** entre eles resulta no seguinte:

```

root@n1:/tmp/pycore.45117/n1.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
From 10.0.0.1 icmp_seq=4 Destination Net Unreachable
^C
--- 10.0.6.20 ping statistics ---
13 packets transmitted, 0 received, +4 errors, 100% packet loss, time 12282ms
root@n1:/tmp/pycore.45117/n1.conf# █

```

Fig. 12: Falha a fazer um *ping* do Host *n1* para o Host *n3*

## 2.8 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 número de saltos, mas sim por um outro caminho alternativo

Para configurar um dos routers de modo que o tráfego na rede use uma rota alternativa com um número superior de saltos é necessário utilizar o comando "**offset-list**"<sup>1</sup>, este comando permite alterar a métrica para uma rota específica. Foi efetuado um *traceroute* inicial do host *n1* para o host *n4* com os endereços IP 10.0.0.20 e 10.0.6.21, respetivamente.

```

n1.conf > traceroute 10.0.6.21
traceroute to 10.0.6.21 (10.0.6.21), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.098 ms  0.028 ms  0.027 ms
 2  10.0.1.2 (10.0.1.2)  0.071 ms  0.043 ms  0.043 ms
 3  10.0.2.2 (10.0.2.2)  0.082 ms  0.063 ms  0.063 ms
 4  10.0.6.21 (10.0.6.21)  0.102 ms  0.080 ms  0.080 ms
n1.conf >

```

Fig. 13: Traceroute Inicial entre *n1* e *n4*.

Para alterar a rota *preferencial* foi escolhido o router *n7* por conter três "*interfaces*" com ligações e uma possível rota alternativa. Entrando no modo de configuração, o primeiro passo foi visualizar a *IP Route Table* do router.

<sup>1</sup> RIP-Metric-Manipulation Configure RIP Route Metric

```
n7# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface
```

	Network	Next Hop	Metric	From	Tag	Time
C(i)	10.0.0.0/24	0.0.0.0	1	self	0	
C(i)	10.0.1.0/24	0.0.0.0	1	self	0	
R(n)	10.0.2.0/24	10.0.1.2	2	10.0.1.2	0	02:46
R(n)	10.0.3.0/24	10.0.1.2	3	10.0.1.2	0	02:46
R(n)	10.0.4.0/24	10.0.5.1	2	10.0.5.1	0	02:46
C(i)	10.0.5.0/24	0.0.0.0	1	self	0	
R(n)	10.0.6.0/24	10.0.1.2	3	10.0.1.2	0	02:46
R(n)	10.0.7.0/24	10.0.5.1	2	10.0.5.1	0	02:46
R(n)	10.0.8.0/24	10.0.5.1	3	10.0.5.1	0	02:46

```
n7#
```

Fig. 14: Original RIP IP Tables

Para o comando *"offset-list"* funcionar foi necessário criar uma *"access-list"* para alterar a métrica dessa rota. O comando usado foi *"access-list 1 permit any"* que configura o router para permitir todo o tráfego IP, sem quaisquer restrições de origem ou destino.

Procedeu-se depois a alteração da métrica das rotas com o comando *"offset-list 1 out 5"*. Este comando indica ao router para aumentar o valor da métrica nas rotas correspondentes à lista de acesso 1 com 5, à medida que essas são anunciadas para fora (*"out"*).

```
n7# configure t
n7(config)# router rip
n7(config-router)# offset-list 1 out 5
n7(config-router)# exit
n7(config)# access-list 1 permit any
n7(config)# exit
n7#
```

Fig. 15: Commando executados no router *n7*

Após aplicar a nova configuração, o mesmo *traceroute* foi executado, resultando desta vez em mais um salto. Verificou-se também uma alteração na *IP Route Table* do router na linha **R(n) 10.0.3.0/24 10.0.1.2 3 10.0.1.2 0 02:46** que passou a ter o valor de **R(n) 10.0.3.0/24 10.0.5.1 3 10.0.5.1 0 03:00**

```

n1.conf > traceroute 10.0.6.21
traceroute to 10.0.6.21 (10.0.6.21), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.128 ms  0.027 ms  0.025 ms
 2  10.0.5.1 (10.0.5.1)  0.055 ms  0.042 ms  0.039 ms
 3  10.0.4.1 (10.0.4.1)  0.069 ms  0.057 ms  0.058 ms
 4  10.0.2.2 (10.0.2.2)  0.086 ms  0.065 ms  0.065 ms
 5  10.0.6.21 (10.0.6.21)  0.137 ms  0.105 ms  0.119 ms
n1.conf >

```

Fig. 16: Traceroute depois de definir a nova rota.

```

n7# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

```

	Network	Next Hop	Metric	From	Tag	Time
C(i)	10.0.0.0/24	0.0.0.0	1	self	0	
C(i)	10.0.1.0/24	0.0.0.0	1	self	0	
R(n)	10.0.2.0/24	10.0.1.2	2	10.0.1.2	0	03:00
R(n)	10.0.3.0/24	10.0.5.1	3	10.0.5.1	0	03:00
R(n)	10.0.4.0/24	10.0.5.1	2	10.0.5.1	0	03:00
C(i)	10.0.5.0/24	0.0.0.0	1	self	0	
R(n)	10.0.6.0/24	10.0.1.2	3	10.0.1.2	0	03:00
R(n)	10.0.7.0/24	10.0.5.1	2	10.0.5.1	0	03:00
R(n)	10.0.8.0/24	10.0.5.1	3	10.0.5.1	0	03:00

```

n7#

```

Fig. 17: RIP IP Table alterada.

## 2.9 Apresente e explique o(s) comando(s) que usaria na sua rede de interligação para que todo o tráfego dirigido a redes externas saísse por um router x

Para esta e a próxima secção foi necessário alterar a topologia de modo a simular redes externas. Para esse efeito colocou-se um novo router *n12* ligado ao router *n11*.

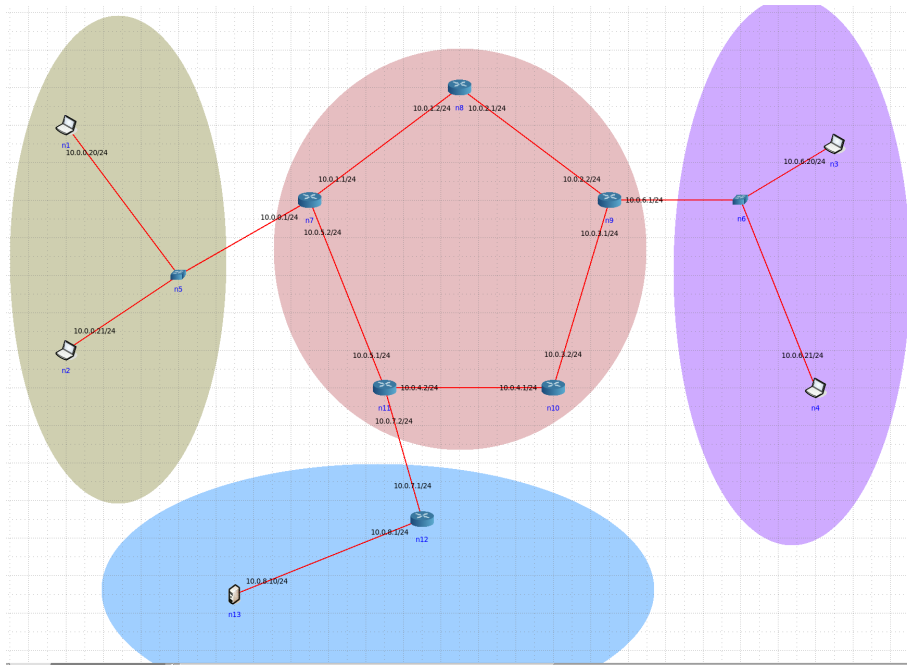


Fig. 18: Nova Topologia.

Para colocar o *router x 11* como um Gateway para qualquer tráfego dirigido a redes externas foi executado o comando **"ip route 0.0.0.0 0.0.0.0 10.0.7.1"**. Este comando configura o router a enviar todo o tráfego que não possuía uma rota na sua *RIP IP Table* para o endereço IP 10.0.7.1.

```
n11# configure t
n11(config)# ip route 0.0.0.0 0.0.0.0 10.0.7.1
n11(config)# exit
n11#
```

Fig. 19: Comando Executado para adicionar a rota.

```
n11# 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

S>* 0.0.0.0/0 [1/0] via 10.0.7.1, eth2
R>* 10.0.0.0/24 [120/7] via 10.0.5.2, eth1, 00:14:23
R>* 10.0.1.0/24 [120/4] via 10.0.4.1, eth0, 00:14:23
R>* 10.0.2.0/24 [120/3] via 10.0.4.1, eth0, 00:14:23
R>* 10.0.3.0/24 [120/2] via 10.0.4.1, eth0, 00:14:23
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, 00:14:23
C>* 10.0.7.0/24 is directly connected, eth2
R>* 10.0.8.0/24 [120/2] via 10.0.7.1, eth2, 00:14:22
C>* 127.0.0.0/8 is directly connected, lo
n11#
```

Fig. 20: IP RIP Table

- 2.10** 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 através do router x.

Para confirmar se a rota foi bem anunciada verificamos se na tabela de routing de outro router existia a nova entrada com a rota definida previamente.

```

Hello, this is Quagga (version 0.99.21mr2.2).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

n9# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From          Tag Time
R(n) 0.0.0.0/0         10.0.3.2          3 10.0.3.2           0 02:56
R(n) 10.0.0.0/24       10.0.2.1          3 10.0.2.1           0 02:56
R(n) 10.0.1.0/24       10.0.2.1          2 10.0.2.1           0 02:56
C(i) 10.0.2.0/24       0.0.0.0           1 self              0
C(i) 10.0.3.0/24       0.0.0.0           1 self              0
R(n) 10.0.4.0/24       10.0.3.2          2 10.0.3.2           0 02:56
R(n) 10.0.5.0/24       10.0.2.1          3 10.0.2.1           0 02:56
C(i) 10.0.6.0/24       0.0.0.0           1 self              0
R(n) 10.0.7.0/24       10.0.3.2          3 10.0.3.2           0 02:56
R(n) 10.0.8.0/24       10.0.3.2          4 10.0.3.2           0 02:56
n9#

```

Fig. 21: IP RIP Table do router *n9*

Para haver conectividade entre as redes externas e a rede principal foi necessário repetir o processo realizado na questão anterior no router *n12* mas com o IP do router *n11*, 10.0.7.2.

```

n12# conf t
n12(config)# ip route 0.0.0.0 0.0.0.0 10.0.7.2
n12(config)# exit
n12# 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

S>* 0.0.0.0/0 [1/0] via 10.0.7.2, eth0
R>* 10.0.0.0/24 [120/8] via 10.0.7.2, eth0, 00:12:56
R>* 10.0.1.0/24 [120/5] via 10.0.7.2, eth0, 00:12:56
R>* 10.0.2.0/24 [120/4] via 10.0.7.2, eth0, 00:12:56
R>* 10.0.3.0/24 [120/3] via 10.0.7.2, eth0, 00:12:56
R>* 10.0.4.0/24 [120/2] via 10.0.7.2, eth0, 00:12:56
R>* 10.0.5.0/24 [120/2] via 10.0.7.2, eth0, 00:12:56
R>* 10.0.6.0/24 [120/4] via 10.0.7.2, eth0, 00:12:56
C>* 10.0.7.0/24 is directly connected, eth0
C>* 10.0.8.0/24 is directly connected, eth1
C>* 127.0.0.0/8 is directly connected, lo
n12#

```

Fig. 22: Execução do comando no router *n12* e visualização da tabela de routing

Para confirmar a conectividade entre as redes externas e a rede interna foi realizado um **ping** e um **traceroute** entre os hosts *n1* e *n13* com os IPS 10.0.0.20 e 10.0.8.10 respetivamente.

```

n1.conf > ping 10.0.8.10
PING 10.0.8.10 (10.0.8.10) 56(84) bytes of data.
64 bytes from 10.0.8.10: icmp_seq=1 ttl=61 time=0.130 ms
64 bytes from 10.0.8.10: icmp_seq=2 ttl=61 time=0.147 ms
64 bytes from 10.0.8.10: icmp_seq=3 ttl=61 time=0.159 ms
64 bytes from 10.0.8.10: icmp_seq=4 ttl=61 time=0.145 ms
^C
--- 10.0.8.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3041ms
rtt min/avg/max/mdev = 0.130/0.145/0.159/0.010 ms
n1.conf >

```

Fig. 23: Ping *n1* a *n13*



```

n1.conf > traceroute 10.0.8.10
traceroute to 10.0.8.10 (10.0.8.10), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.044 ms  0.010 ms  0.010 ms
 2  10.0.5.1 (10.0.5.1)  0.026 ms  0.018 ms  0.018 ms
 3  10.0.7.1 (10.0.7.1)  0.033 ms  0.024 ms  0.024 ms
 4  10.0.8.10 (10.0.8.10)  0.040 ms  0.032 ms  0.032 ms
n1.conf >

```

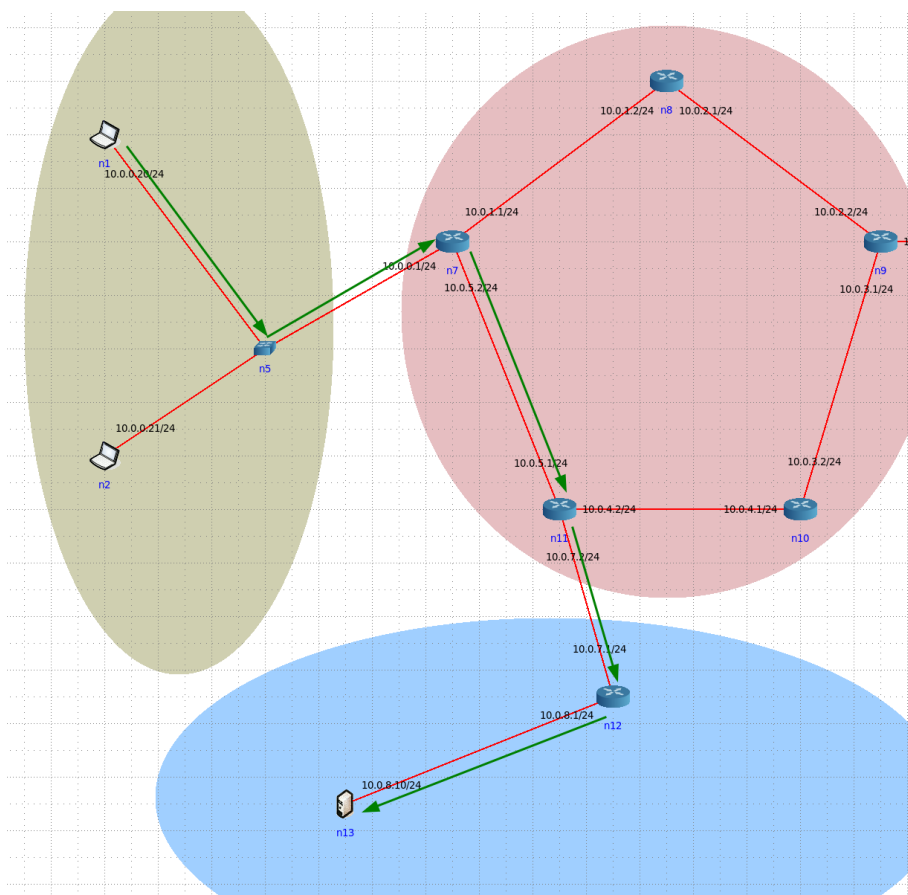
Fig. 24: Traceroute  $n1$  a  $n13$ 

Fig. 25: Caminho percorrido no traceroute

## References

1. Quagga documentation. (n.d.). Quagga. Retrieved March 19, 2024, from <https://www.nongnu.org/quagga/docs/quagga.html>
2. Configure RIP Route Metric (n.d.) Retrieved March 19, 2024, from <https://www.geeksforgeeks.org/configuring-rip-route-metric-offset-lists-in-cisco/>