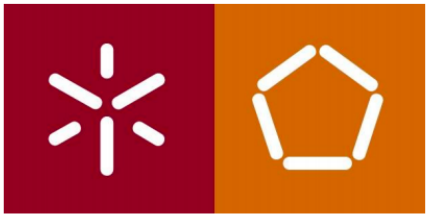


Universidade do Minho

Mestrado Integrado em Engenharia Informática

22/10/2019

Redes de Computadores
TP2: Protocolo IPv4

PL4 Grupo 9:
António Gonçalves (A85516)
João Araújo (A84306)
João Silva (A81761)

Contents

1 Parte 1

1.1 Questão 1 - Comportamento do traceroute na topologia CORE

A topologia CORE que nos foi pedida para implementar neste parte do trabalho foi a seguinte:

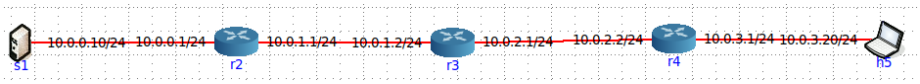


Figure 1: Topologia CORE.

Alínea a. -Active o wireshark ou o tcpdump no pc s1. Numa shell de s1, execute o comando traceroute -I para o endereço IP do host h5.

```
root@s1:/tmp/pycore.38519/s1.conf# traceroute -I 10.0.3.20
traceroute to 10.0.3.20 (10.0.3.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.036 ms  0.010 ms  0.007 ms
 2  10.0.1.2 (10.0.1.2)  0.019 ms  0.011 ms  0.010 ms
 3  10.0.2.2 (10.0.2.2)  0.022 ms  0.014 ms  0.015 ms
 4  10.0.3.20 (10.0.3.20)  0.050 ms  0.021 ms  0.017 ms
```

Figure 2: Execução do traceroute -I 10.0.3.20.

Alínea b. -Registe e analise o tráfego ICMP enviado por s1 e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.

Após executar o comando "traceroute -I 10.0.2.10" a partir do host s1, são devolvidos 4 tempos relativos aos 4 pacotes enviados para o host h5. Como se observa, o tráfego correspondente a enviar do s1 para o host h5 corresponde a 4 datagramas.

```
133 78.098260907  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=1/256, ttl=1 (no response found!)
134 78.098280144  10.0.0.1   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
135 78.098295217  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=2/512, ttl=1 (no response found!)
136 78.098301147  10.0.0.1   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
137 78.098306273  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=3/768, ttl=1 (no response found!)
138 78.098310801  10.0.0.1   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
139 78.098316343  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=4/1024, ttl=2 (no response found!)
140 78.098332525  10.0.1.2   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
141 78.098338175  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=5/1280, ttl=2 (no response found!)
142 78.098346225  10.0.1.2   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
143 78.098350939  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=6/1536, ttl=2 (no response found!)
144 78.098358510  10.0.1.2   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
145 78.098364083  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=7/1792, ttl=3 (no response found!)
146 78.098383998  10.0.2.2   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
147 78.098389143  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=8/2048, ttl=3 (no response found!)
148 78.098406639  10.0.2.2   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
149 78.098405805  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=9/2304, ttl=3 (no response found!)
150 78.098419019  10.0.2.2   10.0.0.10  ICMP  102 Time-to-live exceeded (Time to live exceeded in transit)
151 78.098423958  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=10/2560, ttl=4 (reply in 152)
152 78.098470510  10.0.3.20  10.0.0.10  ICMP  74 Echo (ping) reply    id=0x0025, seq=10/2560, ttl=61 (request in 151)
153 78.098482308  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=11/2816, ttl=4 (reply in 154)
154 78.098498866  10.0.3.20  10.0.0.10  ICMP  74 Echo (ping) reply    id=0x0025, seq=11/2816, ttl=61 (request in 153)
155 78.098504283  10.0.0.10  10.0.3.20  ICMP  74 Echo (ping) request  id=0x0025, seq=12/3072, ttl=4 (reply in 156)
156 78.098518433  10.0.3.20  10.0.0.10  ICMP  74 Echo (ping) reply    id=0x0025, seq=12/3072, ttl=61 (request in 155)
```

Figure 3: Tráfego ICMP enviado por s1 e tráfego recebido como resposta.

Alínea c. -Qual deve ser o valor inicial mínimo do campo TTL para alcançar o destino h5? Verifique na prática que a sua resposta está correta.

O valor mínimo do campo TTL deverá ser 4 para poder alcançar o h5, como podemos ver pela imagem da pergunta anterior.

Alínea d. -Qual o valor médio do tempo de ida-e-volta (Round-Trip Time) obtido?

O tempo médio será: $(0.050 + 0.021 + 0.017)/3 \approx 0,029(3)$

1.2 Questão 2 - Traceroute na máquina nativa e criação de datagramas IP

Alínea a. -Qual é o endereço IP da interface ativa do seu computador?

Figure 4: IP da interface nativa

O IP do interface ativa do PC é 192.168.100.196

Alínea b. -Qual é o valor do campo protocolo? O que identifica?

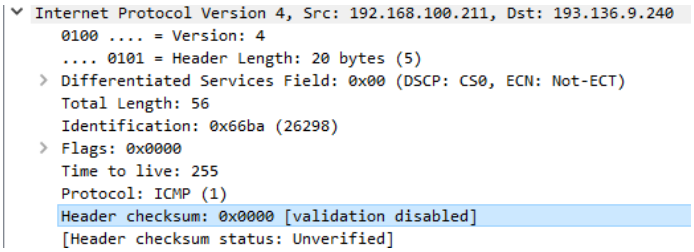


Figure 5: Informações sobre o cabeçalho IPv4.

Como podemos observar, o valor do protocolo é 1 e este identifica o protocolo ICMP.

Alínea c. -Quantos bytes tem o cabeçalho IP(v4)? Quantos bytes tem o campo de dados (payload) do datagrama? Como se calcula o tamanho do payload?

O cabeçalho IP(v4) tem 20 bytes e o campo de dados 56 bytes(como podemos observar na Fig.5). Assim, o tamanho do payload será: 56 - 20 = 36 bytes

Alínea d. -O datagrama IP foi fragmentado? Justifique.

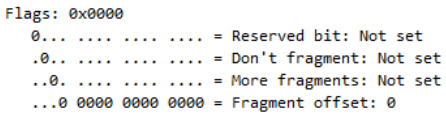


Figure 6: Flags da mensagem.

Podemos observar na imagem a cima que esta mensagem não foi fragmentada, uma vez que o *Fragment offset* está a 0 e a flag *More fragment* está como Not set.

Alínea e. -Ordene os pacotes capturados de acordo com o endereço IP fonte (e.g., selecionando o cabeçalho da coluna Source), e analise a sequência de tráfego ICMP gerado a partir do endereço IP atribuído à interface da sua máquina. Para a sequência de mensagens ICMP enviadas pelo seu computador, indique que campos do cabeçalho IP variam de pacote para pacote.

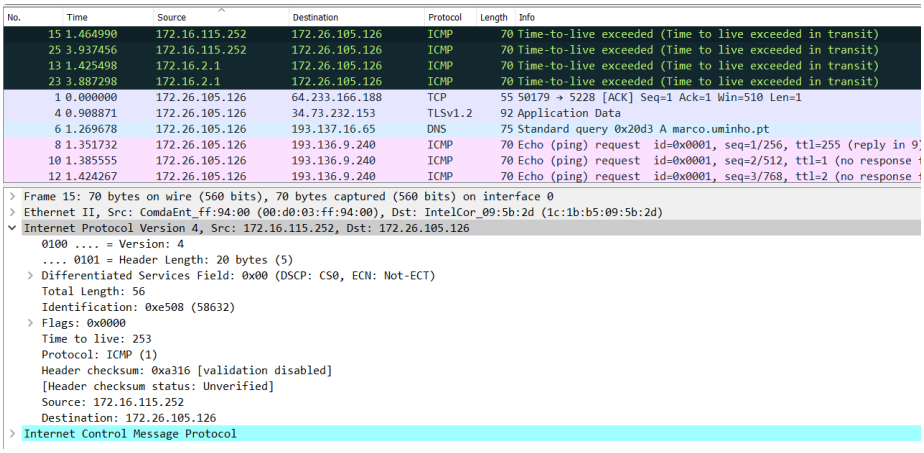


Figure 7: Pacotes ordenados de acordo com o IP da fonte.

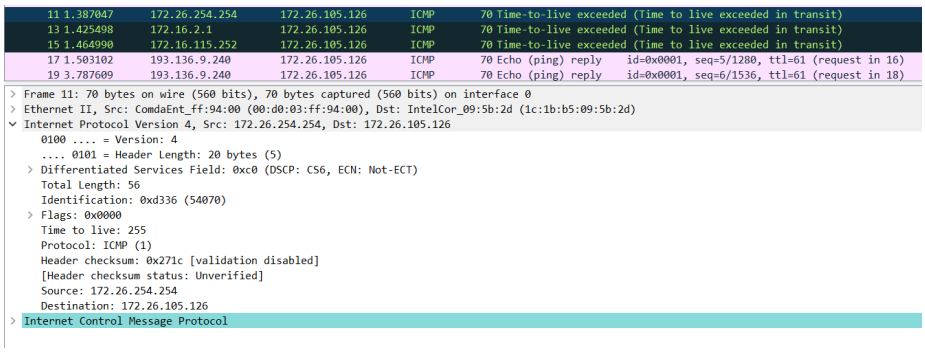
Os campos que variam são o campo de identificação e o TTL(Time to Live).

Alínea f. -Observa algum padrão nos valores do campo de Identificação do datagrama IP e TTL?

Os valores de identificação dos pacotes consecutivos têm um valor de diferença(ordem crescente). Os valores de TTL também são sequenciais, até chegar ao valor 4.

Alínea g. -Ordene o tráfego capturado por endereço destino e encontre a série de respostas ICMP TTL exceeded enviadas ao seu computador. Qual é o valor do campo TTL? Esse valor permanece constante para todas as mensagens de resposta ICMP TTL exceeded enviados ao seu host? Porquê?

O valor do TTL inicial vai diminuindo em 1 valor sempre que é mandada outra resposta.



1.3 Questão 3 - Análise da fragmentação de pacotes de IP

Alínea a- Localize a primeira mensagem ICMP. Porque é que houve necessidade de fragmentar o pacote inicial?

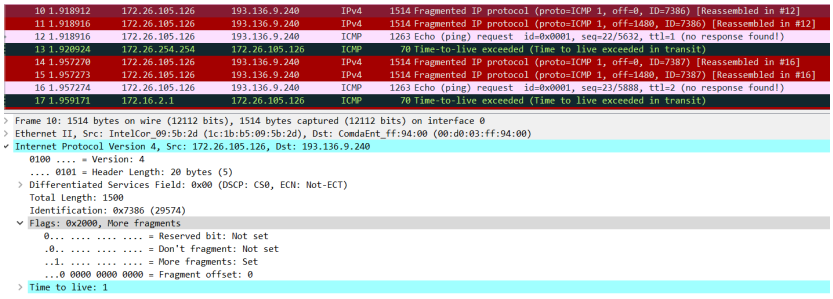


Figure 8: Primeira mensagem ICMP.

Houve necessidade de fragmentar o pacote inicial uma vez que o novo tamanho do pacote (neste caso, 4209) é demasiado grande para ser enviado num só (tamanho Maximo de um fragmento = 1500 bytes).

Alínea b- Imprima o primeiro fragmento do datagrama IP segmentado. Que informação no cabeçalho indica que o datagrama foi fragmentado? Que informação no cabeçalho IP indica que se trata do primeiro fragmento?Qual é o tamanho deste datagrama IP?

Como podemos observar na imagem anterior, o *Fragment offset* tem o valor de 0 e a flag *More fragments* como Set, o que nos permite concluir que o datagrama foi fragmentado. Com a imagem seguinte conseguimos observar o tamanho dos fragmentos que fazem parte do datagrama, sendo o tamanho do primeiro fragmento 1480 bytes.

```
[3 IPv4 Fragments (4189 bytes): #10(1480), #11(1480), #12(1229)]
   [Frame: 10, payload: 0-1479 (1480 bytes)]
   [Frame: 11, payload: 1480-2959 (1480 bytes)]
   [Frame: 12, payload: 2960-4188 (1229 bytes)]
   [Fragment count: 3]
   [Reassembled IPv4 length: 4189]
   [Reassembled IPv4 data: 080091a2000100162020202020202020202020202020202020…]
```

Figure 9: Os fragmentos da mensagem.

Alínea c- Imprima o segundo fragmento do datagrama IP original. Que informação do cabeçalho IP indica que não se trata do 1º fragmento? Há mais fragmentos? O que nos permite afirmar isso?

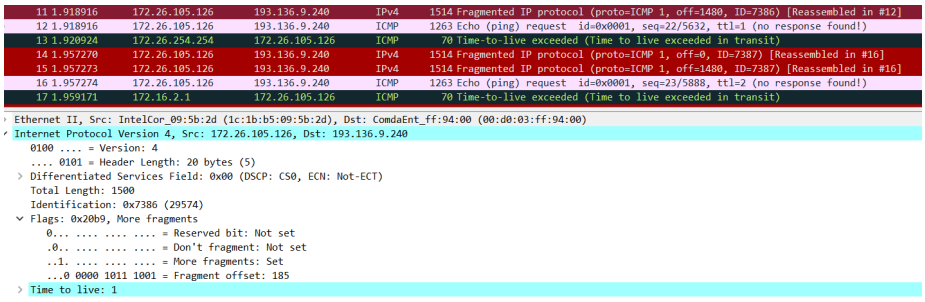


Figure 10: Segundo fragmento do datagrama IP original.

Podemos dizer que este fragmento não é o primeiro uma vez que o seu *Fragment offset* é diferente de 0. Para além disso, conseguimos afirmar que há mais fragmentos uma vez que a flag *More fragments* está Set.

Alínea d- Quantos fragmentos foram criados a partir do datagrama original? Como se detecta o último fragmento correspondente ao datagrama original?

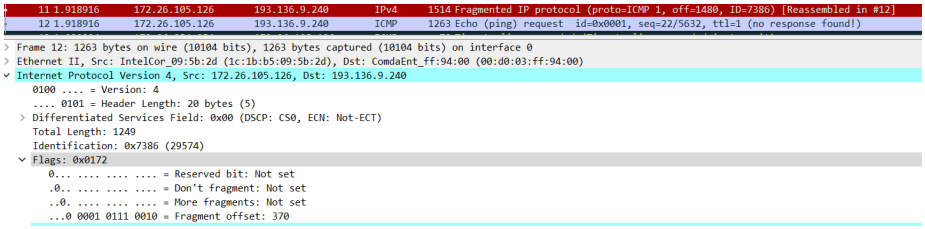


Figure 11: Informações sobre o último fragmento do datagrama.

Foram criados 3 fragmentos ao todo a partir do datagrama original. Podemos afirmar que o datagrama presente na imagem é o último já que a flag *More Fragments* está Not set e o *Fragment offset* tem um valor diferente de 0.

Alínea e- Indique, resumindo, os campos que mudam no cabeçalho IP entre os diferentes fragmentos, e explique a forma como essa informação permite reconstruir o datagrama original.

Resumindo, os campos do cabeçalho IP que mudam entre os vários fragmentos são o *Fragment offset* e a Flag *More Fragments*.

O *Fragment offset* permite-nos saber que parte da mensagem está a ser enviada, servindo para reconstruir a mensagem original. Assim, quando um dos fragmentos é tratado, o *Fragment offset* vai ser alterado para o início do próximo fragmento.

A Flag *More fragments* indica se existem ou não outros fragmentos que ainda não foram lidos, impedindo que alguns fragmentos sejam perdidos no processo.

2 Parte 2

2.1 Questão 1 - Endereçamento e Encaminhamento IP

2.1.1 - Atenda aos endereços IP atribuídos automaticamente pelo CORE aos diversos equipamentos da topologia.

Alínea a. - Indique que endereços IP e máscaras de rede foram atribuídos pelo CORE a cada equipamento. Para simplificar, pode incluir uma imagem que ilustre de forma clara a topologia definida e o endereçamento usado.

Na imaguem seguinte podemos ver os endereços IP atribuidos aos vários equipamentos, Podemos tambem observar que a Máscara utilizada é a 255.255.255.0 (ou /24 notação CIDR).

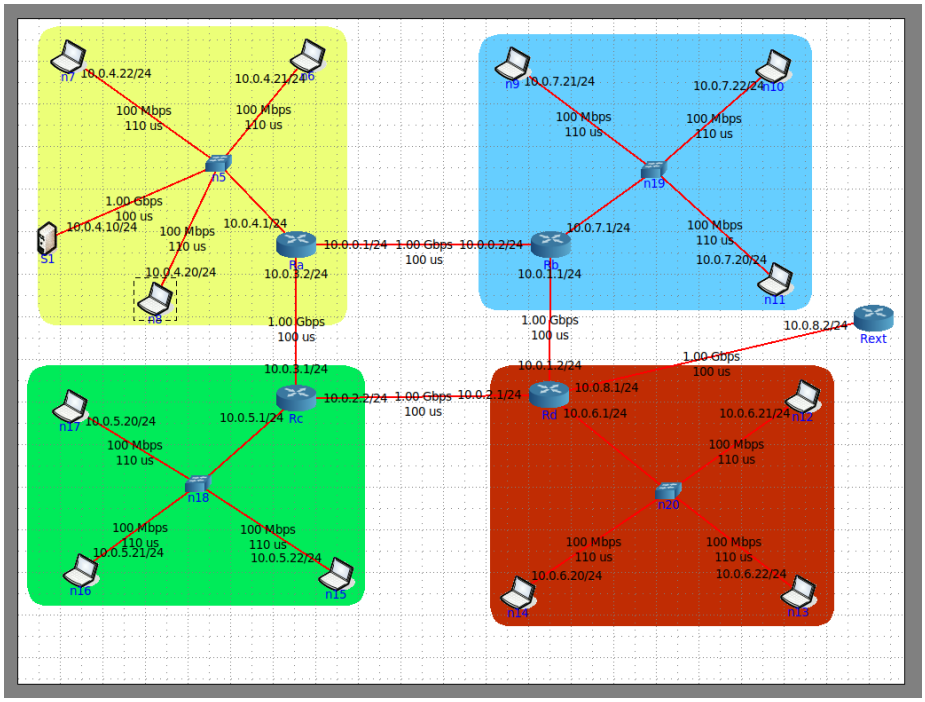


Figure 12: Topologia da Parte 2 do trabalho.

Alínea b. - Trata-se de endereços públicos ou privados? Porquê?

Uma vez que os IPs dos vários equipamentos se encontram no intervalo de 10.0.0.0 e 10.255.255.255, podemos concluir que são privados.

Alínea c. - Por que razão não é atribuído um endereço IP aos switches?
Não são atribuidos IPs aos switches uma vez que estes são switches de Ethernet.

Alínea d. - Usando o comando ping certifique-se que existe conectividade IP entre os laptops dos vários departamentos e o servidor do departamento A (basta certificar-se da conectividade de um laptop por departamento).

As seguintes imagens comprovam que o servidor S1 está ligado a todos os outros laptops, uma vez que os pacotes enviados conseguem chegar ao destino.

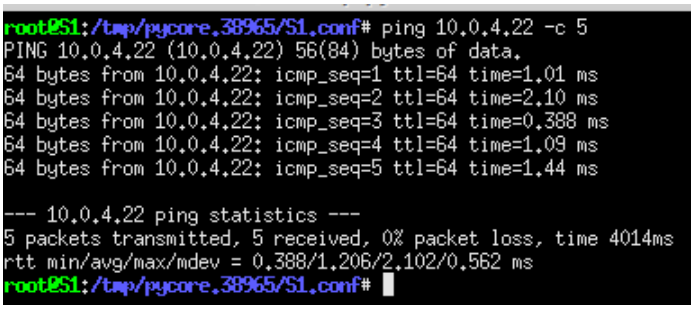


Figure 13: Conectividade entre S1 e um laptop do departamento A.

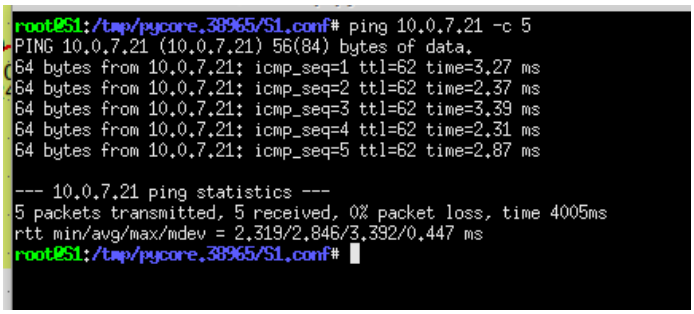


Figure 14: Conectividade entre S1 e um laptop do departamento B.

Figure 15: Conectividade entre S1 e um laptop do departamento C.

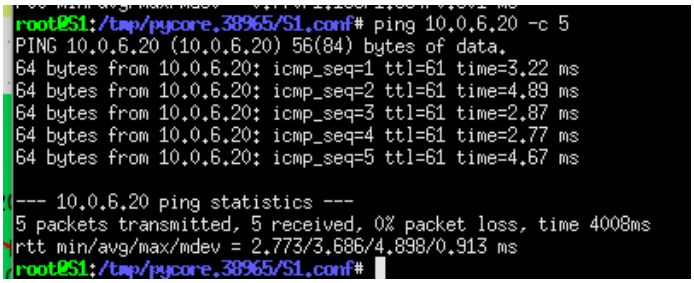


Figure 16: Conectividade entre S1 e um laptop do departamento D.

Alinea e. - Verifique se existe conectividade IP do router de acesso Rext para o servidor S1.

Todos os pacotes enviados foram recebidos, logo existe conectividade OP do router Rext e o servidor S1.

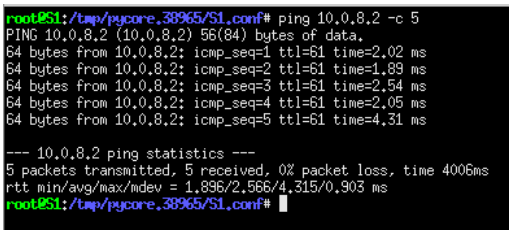


Figure 17: Conectividade entre o Router Rext e o servidor S1.

2.1.2 - Para o router e um laptop do departamento B:

Alinea a. - Execute o comando netstat –rn por forma a poder consultar a tabela de encaminhamento unicast (IPv4). Inclua no seu relatório as tabelas de encaminhamento obtidas; interprete as várias entradas de cada tabela. Se necessário, consulte o manual respetivo (man netstat).

```
root@n9:/tmp/pycore.38965/n9.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.7.1        0.0.0.0         UG        0 0          0 eth0
10.0.7.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
root@n9:/tmp/pycore.38965/n9.conf#
```

Figure 18: Resultado do comando netstat -rn para um laptop.

Para o laptop existem só duas rotas: a default, que tem como destino o router Rb (0.0.0.0), e a outra tem como destino o departamento B (10.0.7.0).

```
root@Rb:/tmp/pycore.38965/Rb.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.0.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.2.0        10.0.1.2        255.255.255.0   UG        0 0          0 eth1
10.0.3.0        10.0.0.1        255.255.255.0   UG        0 0          0 eth0
10.0.4.0        10.0.0.1        255.255.255.0   UG        0 0          0 eth0
10.0.5.0        10.0.0.1        255.255.255.0   UG        0 0          0 eth0
10.0.6.0        10.0.1.2        255.255.255.0   UG        0 0          0 eth1
10.0.7.0        0.0.0.0         255.255.255.0   U         0 0          0 eth2
10.0.8.0        10.0.1.2        255.255.255.0   UG        0 0          0 eth1
```

Figure 19: Resultado do comando netstat -rn para um router.

Para o Router Rb existem várias entradas, como por exemplo, um pacote que tenha de chegar a 10.0.2.0 terá obrigatoriamente de passar por 10.0.1.2.

Alinea b. - Diga, justificando, se está a ser usado encaminhamento estático ou dinâmico (sugestão: analise que processos estão a correr em cada sistema).

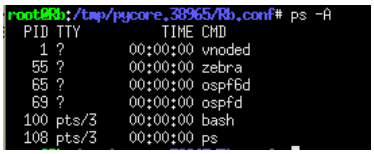


Figure 20: Processos a decorrer no Router Rb.

Entre os Routers está a ser utilizado encaminhamento dinâmico, uma vez que existem vários caminhos possíveis (podemos observar o protocolo ospfd, que permite ao pacote seguir vários caminhos diferentes quando não lhes é possível seguir a rota predefinida).

Figure 21: Processos a decorrer no Laptop n9.

Nos laptops, está a ser utilizado encaminhamento estático, uma vez que só existe um rota possível. Caso não seja possível seguir esse caminho, o pacote é descartado.

Alinea c. - Admita que, por questões administrativas, a rota por defeito (0.0.0.0 ou default) deve ser retirada definitivamente da tabela de encaminhamento do servidor S1 localizado no departamento A. Use o comando route delete para o efeito. Que implicação tem esta medida para os utilizadores da empresa que acedem ao servidor? Justifique.

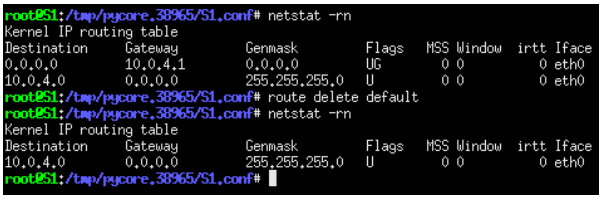


Figure 22: Resultado do comando netstat -rn antes e depois de remover a rota *default*.

Ao remover a rota default, estamos a impossibilitar o servidor S1 de se ligar a equipamentos fora do seu departamento.

Alinea d. - Adicione as rotas estáticas necessárias para restaurar a conectividade para o servidor S1 por forma a contornar a restrição imposta na alínea c). Utilize para o efeito o comando route add e registe os comandos que usou.

Será necessário adicionar rotas do servidor S1 até aos routers dos outros departamentos, para que seja possível haver partilha de informação entre eles.

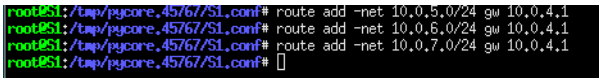


Figure 23: Adicionar novas rotas entre S1 e os restantes routers.

Alinea e. - Teste a nova política de encaminhamento garantindo que o servidor está novamente acessível utilizando para o efeito o comando ping. Registe a nova tabela de encaminhamento do servidor.

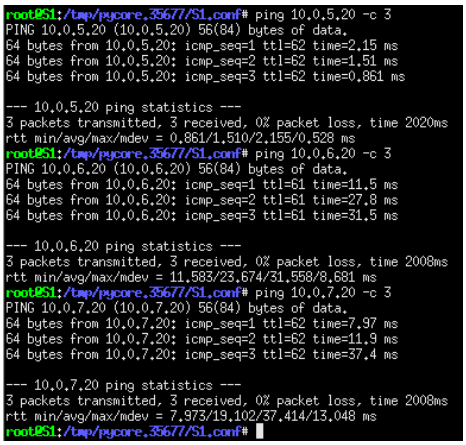


Figure 24: Envio de mensagens do servidor S1 para laptops do departamento C, D e B.

Como podemos observar, após adicionar as rotas para os diversos rotes dos departamento, os seus laptops passaram a ser acessíveis ao servidor S1.

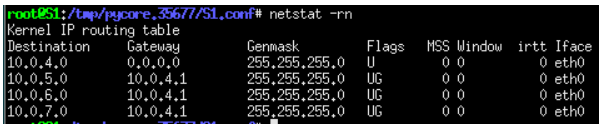


Figure 25: Rotas que foram adicionadas 1 S1.

2.2 Questão 2 - Definição de Sub-redes

Alinea a. - Considere que dispõe apenas do endereço de rede IP 172.yyx.32.0/20, em que "yy" são os dígitos correspondendo ao seu número de grupo (Gyy) e "x" é o dígito correspondente ao seu turno prático (PLx). Defina um novo esquema de endereçamento para as redes dos departamentos (mantendo a rede de acesso e core inalteradas) e atribua endereços às interfaces dos vários sistemas envolvidos. Deve justificar as opções usadas.

Endereço de IP: 172.94.32.0/20. Uma vez que existem 4 departamentos, será necessário definir pelo menos 4 sub-redes. Dividindo o endereço de IP, definimos $2^n - 2$ sub-redes à medida que andamos n bits para a direita. Escolhendo usar 3 bits, ficamos com $2^3 - 2 = 6$ endereços disponíveis.

Atribuição de endereços:

Sub-rede 1: 000 | default
Sub-rede 2: 001 | 172.94.32.0/23 | Departamento A
Sub-rede 3: 010 | 172.94.34.0/23 | livre
Sub-rede 4: 011 | 172.94.36.0/23 | Departamento B
Sub-rede 5: 100 | 172.94.38.0/23 | Departamento C
Sub-rede 6: 101 | 172.94.40.0/23 | livre
Sub-rede 7: 110 | 172.94.42.0/23 | Departamento D
Sub-rede 8: 111 | broadcast

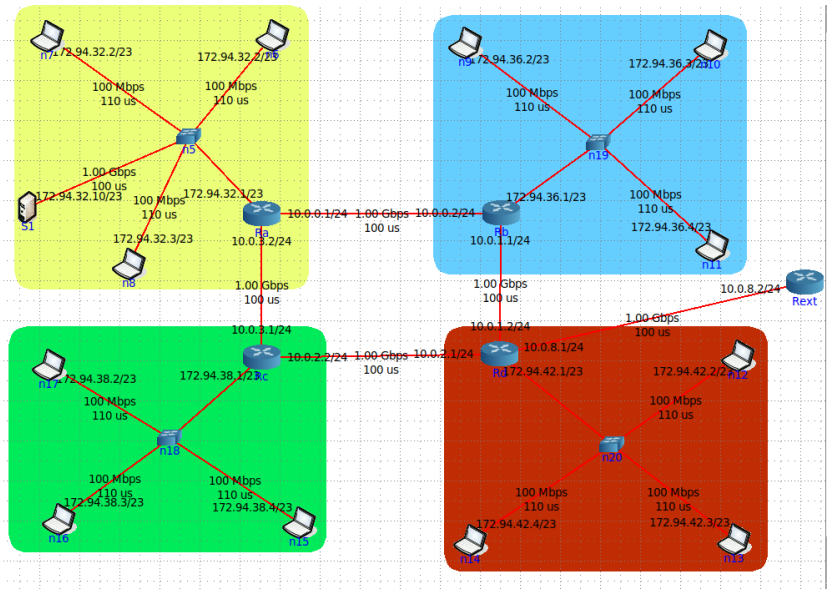


Figure 26: Novos IPs das ligações entre routers e Laptops.

Alinea b. - Qual a máscara de rede que usou (em notação decimal)? Quantos interfaces IP pode interligar em cada departamento? Justifique.

A máscara de rede que foi usada foi /23 (255.255.254.0). O número de interfaces IP que se podem interligar em cada departamento é $2^n - 2$ sendo $n = 32 - 23$,ou seja, $2^9 - 2 = 510$.

Alinea c. - Garanta e verifique que a conectividade IP entre as várias redes locais da organização MIEI-RC é mantida. Explique como procedeu.

```
root@n7:/tmp/pycore.44273/n7.conf# ping 172.94.38.2  -c 3
PING 172.94.38.2 (172.94.38.2) 56(84) bytes of data.
64 bytes from 172.94.38.2: icmp_seq=1 ttl=62 time=3.26 ms
64 bytes from 172.94.38.2: icmp_seq=2 ttl=62 time=69.8 ms
64 bytes from 172.94.38.2: icmp_seq=3 ttl=62 time=11.7 ms

--- 172.94.38.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 3.265/28.306/69.868/29.594 ms
root@n7:/tmp/pycore.44273/n7.conf# ping 172.94.36.2  -c 3
PING 172.94.36.2 (172.94.36.2) 56(84) bytes of data.
64 bytes from 172.94.36.2: icmp_seq=1 ttl=62 time=1.63 ms
64 bytes from 172.94.36.2: icmp_seq=2 ttl=62 time=0.911 ms
64 bytes from 172.94.36.2: icmp_seq=3 ttl=62 time=1.44 ms

--- 172.94.36.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.911/1.330/1.630/0.305 ms
root@n7:/tmp/pycore.44273/n7.conf# ping 172.94.42.2  -c 3
PING 172.94.42.2 (172.94.42.2) 56(84) bytes of data.
64 bytes from 172.94.42.2: icmp_seq=1 ttl=61 time=2.47 ms
64 bytes from 172.94.42.2: icmp_seq=2 ttl=61 time=2.08 ms
64 bytes from 172.94.42.2: icmp_seq=3 ttl=61 time=3.12 ms

--- 172.94.42.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.081/2.558/3.123/0.431 ms
root@n7:/tmp/pycore.44273/n7.conf#
```

Figure 27: Envio de pacotes de um laptop do departamento A para vários de outros departamentos.

Podemos observar que existe conectividade entre os vários laptops dos departamentos, uma vez que todos os pacotes enviados foram recebidos sem qualquer problema.

3 Conclusão

Em suma, na primeira parte do trabalho, através de uma topologia CORE, analisou-se o protocolo IPv4, mais propriamente o seu comportamento e o tráfego ICMP recebido.

Já na segunda parte, aprofundamos o estudo dos protocolos IPv4, no que toca ao endereçamento e encaminhamento IP. Utilizando novamente uma topologia CORE, analisamos o encaminhamento entre quatro redes diferentes, isto somado à manipulação de IPs (subnetting).

Posto isto, este trabalho permitiu-nos aprofundar os conhecimentos, não só, em relação,ao funcionamento do CORE e às tipologias deste, mas também aos protocolos IPv4.