TP2: Protocolo IP

Adriana Gonçalves, Eduardo Semanas, and Leonardo Neri

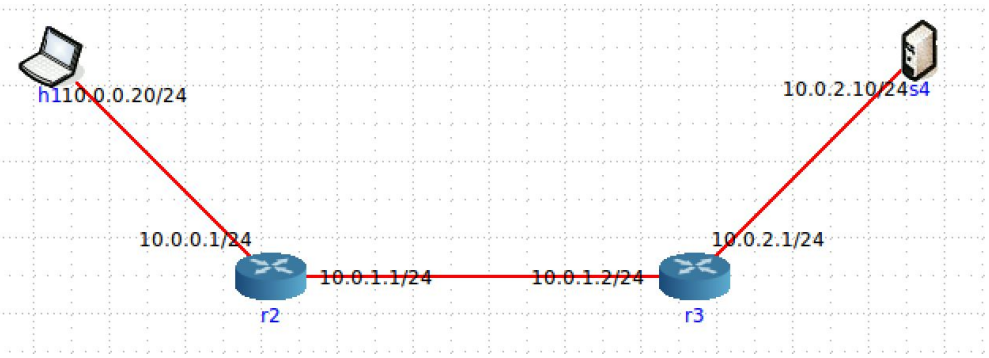
University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {ae4481,a75536,a80056}@alunos.uminho.pt

Introdução Este trabalho realizado no âmbito da unidade curricular de Redes de Computadores pretende explorar e compreender o funcionamento do protocolo IP. Para tal, iremos recorrer ao auxílio dos programas CORE e WIRESHARK de forma a responder às questões apresentadas sobre as diversas vertentes do IP tais como pacotes de dados, endereçamento e encaminhamento IP e a sua fragmentação.

Resolução das questões:

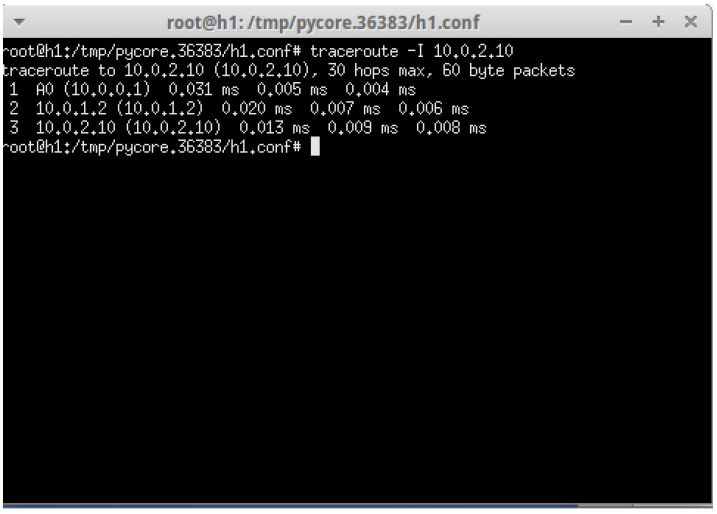
PARTE 1

1. Definimos a topologia da seguinte forma:



1.a) Ative o wireshark ou o tcpdump no pc h1. Numa shell de h1, execute o comando traceroute -I para o IP do host s4.

Depois de executado o wireshark e o comando traceroute -I para o endereço do host s4 obtivemos o seguinte:



1.b) Registe e analise o tráfego ICMP enviado por h1 e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.

Com o TTL=1 o package chega ao router r2 que retorna uma mensagem de erro com o Time to live exceeded. Com TTL=2 acontece o mesmo, mas desta vez o package atinge o router r3. A partir do momento em que TTL=3 nunca mais acontece a interrupção do package ao destino.

No.	Time	Source	Destination	Protocol	Length	Info
4	10.021733	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
5	11.619013	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=1/256, ttl=1
6	11.619036	10.0.0.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
7	11.619041	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=2/512, ttl=1
8	11.619044	10.0.0.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
9	11.619047	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=3/768, ttl=1
10	11.619049	10.0.0.1	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
11	11.619052	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=4/1024, ttl=2
12	11.619062	10.0.1.2	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
13	11.619065	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=5/1280, ttl=2
14	11.619069	10.0.1.2	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
15	11.619071	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=6/1536, ttl=2
16	11.619077	10.0.1.2	10.0.0.20	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
17	11.619080	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=7/1792, ttl=3
18	11.619091	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0047, seq=7/1792, ttl=62
19	11.619094	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=8/2048, ttl=3
20	11.619101	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0047, seq=8/2048, ttl=62
21	11.619103	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=9/2304, ttl=3
22	11.619110	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0047, seq=9/2304, ttl=62
23	11.619112	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=10/2560, ttl=4
24	11.619118	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0047, seq=10/2560, ttl=62
25	11.619120	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=11/2816, ttl=4
26	11.619127	10.0.2.10	10.0.0.20	ICMP	74	Echo (ping) reply id=0x0047, seq=11/2816, ttl=62
27	11.619129	10.0.0.20	10.0.2.10	ICMP	74	Echo (ping) request id=0x0047, seq=12/3072, ttl=4

1.c) Qual deve ser o valor inicial do campo TTL para alcançar o destino s4? Verifique na prática que a sua resposta está correta.

O TTL mínimo para alcançar s4 deverá ser 3, pois quando o TTL toma valores inferiores a 3 o package não chega ao destino aparecendo a informação Time to live exceeded.

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

Atendendo ao primeiro screenshot e fazendo a média dos últimos 3 tempos obtidos, o valor médio do tempo de ida-e-volta é de ((0.013 + 0.009 + 0.006) / 3) = 0.00933333 segundos

2.

```
leonardo@Holmes:~$ sudo traceroute -I marco.uminho.pt
traceroute to marco.uminho.pt (193.136.9.240), 30 hops max, 60 byte packets
 1  172.26.254.254 (172.26.254.254)  161.121 ms  161.100 ms  161.102 ms
 2  172.16.2.1 (172.16.2.1)  2.156 ms  2.171 ms  2.171 ms
 3  172.16.115.252 (172.16.115.252)  3.192 ms  3.233 ms  3.255 ms
 4  marco.uminho.pt (193.136.9.240)  3.190 ms  3.194 ms  3.194 ms
leonardo@Holmes:~$
```

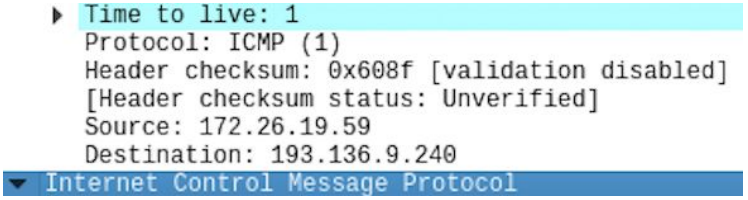
2.a) Qual é o endereço IP da interface ativa do seu computador?

O IP da interface ativa do computador é 172.26.19.59 .

```
leonardo@Holmes:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 28:d2:44:1f:cc:47
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:20 Memory:f2500000-f2520000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:47173 errors:0 dropped:0 overruns:0 frame:0
          TX packets:47173 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:4600184 (4.6 MB)  TX bytes:4600184 (4.6 MB)

wlan0     Link encap:Ethernet  HWaddr a4:4e:31:1b:3d:4c
          inet addr:172.26.19.59  Bcast:172.26.255.255  Mask:255.255.0.0
          inet6 addr: fe80::a64e:31ff:fe1b:3d4c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2295059 errors:0 dropped:0 overruns:0 frame:0
          TX packets:618944 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2818627635 (2.8 GB)  TX bytes:111369668 (111.3 MB)

leonardo@Holmes:~$
```

2.b) Qual é o valor do campo protocolo? O que identifica?

O campo protocolo vale 1 e identifica o protocolo ICMP.



2.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 tamanho do cabeçalho é de 20 bytes. Como o Total Length tem o valor de 52 bytes, os dados do datagrama tomam o valor de 32 bytes (52-20).

```
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x8e6c (36460)
  ▸ Flags: 0x4000, Don't fragment
  ▸ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x608f [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.26.19.59
    Destination: 193.136.9.240
Internet Control Message Protocol
```

2.d) O datagrama IP foi fragmentado? Justifique.

Não, pois o tamanho do Pacote não ultrapassa o MTU.

2.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.

Os Campos que variam entre os cabeçalhos IP enviados é o Time to Live e o identificador

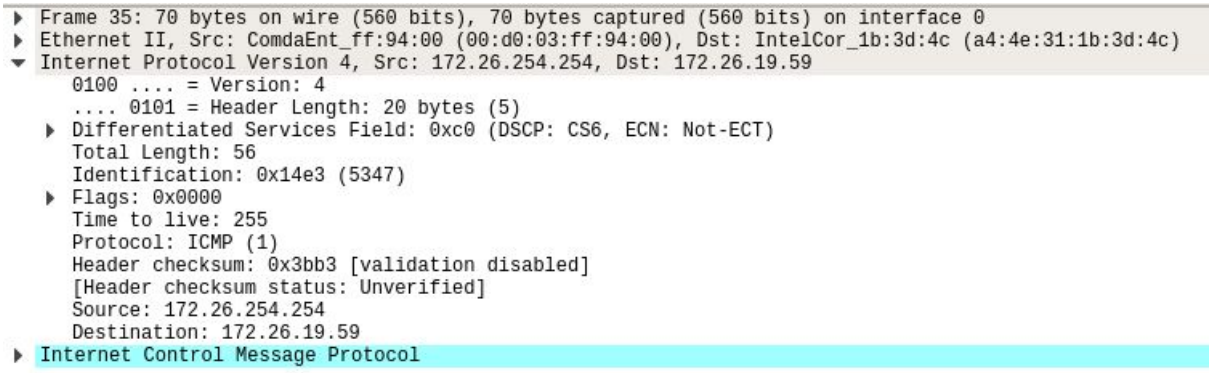
19	6.895188064	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=1/256, ttl=1 (no response found!)
20	6.895220661	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=2/512, ttl=1 (no response found!)
21	6.895235266	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=3/768, ttl=1 (no response found!)
22	6.895249773	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=4/1024, ttl=2 (no response found!)
23	6.895262956	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=5/1280, ttl=2 (no response found!)
24	6.895275132	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=6/1536, ttl=2 (no response found!)
25	6.895289197	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=7/1792, ttl=3 (no response found!)
26	6.895302347	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=8/2048, ttl=3 (no response found!)
27	6.895316712	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=9/2304, ttl=3 (no response found!)
28	6.895331212	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=10/2560, ttl=4 (reply in 42)
29	6.895343727	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=11/2816, ttl=4 (reply in 44)
30	6.895356136	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=12/3072, ttl=4 (reply in 43)
31	6.895371830	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=13/3328, ttl=5 (reply in 46)
32	6.895380108	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=14/3584, ttl=5 (reply in 47)
33	6.895387660	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=15/3840, ttl=5 (reply in 48)
34	6.895396674	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=16/4096, ttl=6 (reply in 50)
53	6.913972389	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=17/4352, ttl=6 (reply in 57)
54	6.914049652	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=18/4608, ttl=6 (reply in 59)
55	6.914093271	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=19/4864, ttl=7 (reply in 60)
73	6.920660452	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=20/5120, ttl=7 (reply in 79)
74	6.920719761	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=21/5376, ttl=7 (reply in 81)
75	6.920757790	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=22/5632, ttl=8 (reply in 82)
91	6.925747743	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=23/5888, ttl=8 (reply in 97)
92	6.925784409	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=24/6144, ttl=8 (reply in 98)
93	6.925806912	172.26.19.59	193.136.9.240	ICMP	74	Echo (ping) request id=0x6de3, seq=25/6400, ttl=9 (reply in 99)

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

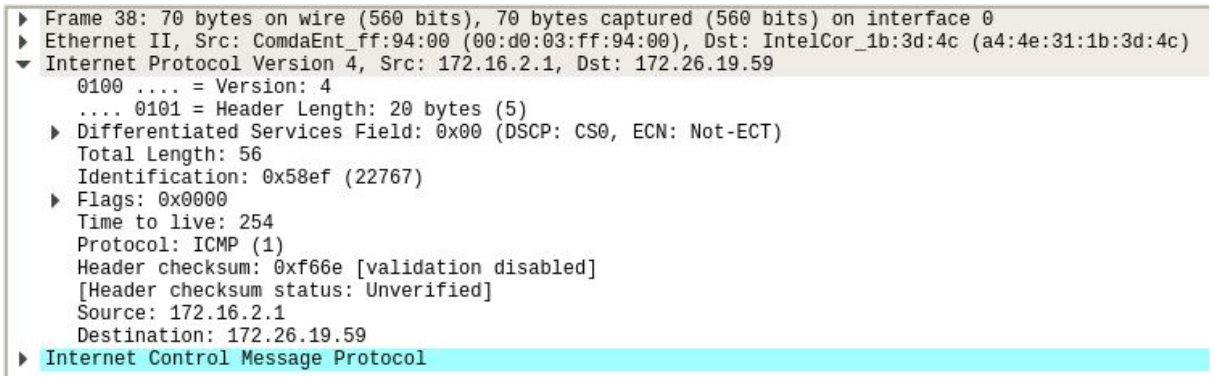
O identificador é sequencial e o *TTL* aumenta em uma unidade a cada três pacotes disparados pelo traceroute.

2.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 não permanece constante, começa em 255, que é o valor máximo, e é decrescido de uma unidade a cada nó que passa até chegar ao host de destino.



Mensagem de resposta do host a um nó de distância, TTL = 255

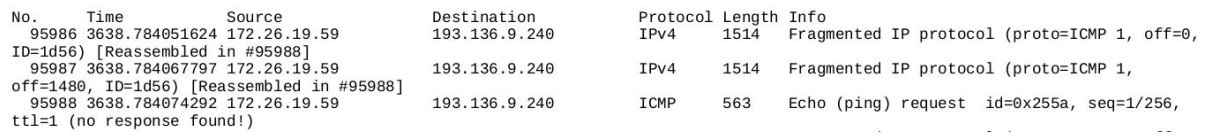


Mensagem de resposta do host a dois nó de distância, TTL = 254

3.

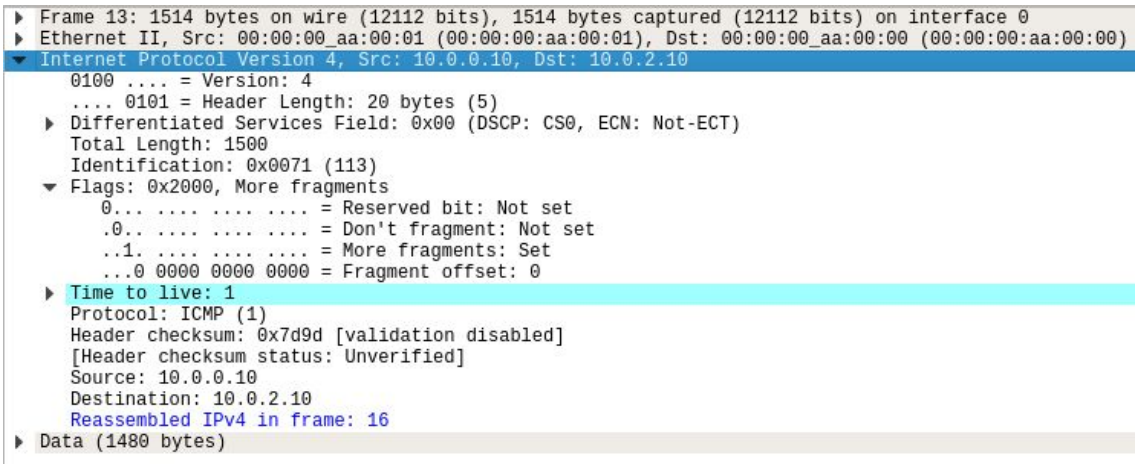
3.a) Localize a primeira mensagem ICMP. Porque é que houve necessidade de fragmentar o pacote inicial?

O pacote ultrapassou o *Maximum Transmission Unit* (MTU).



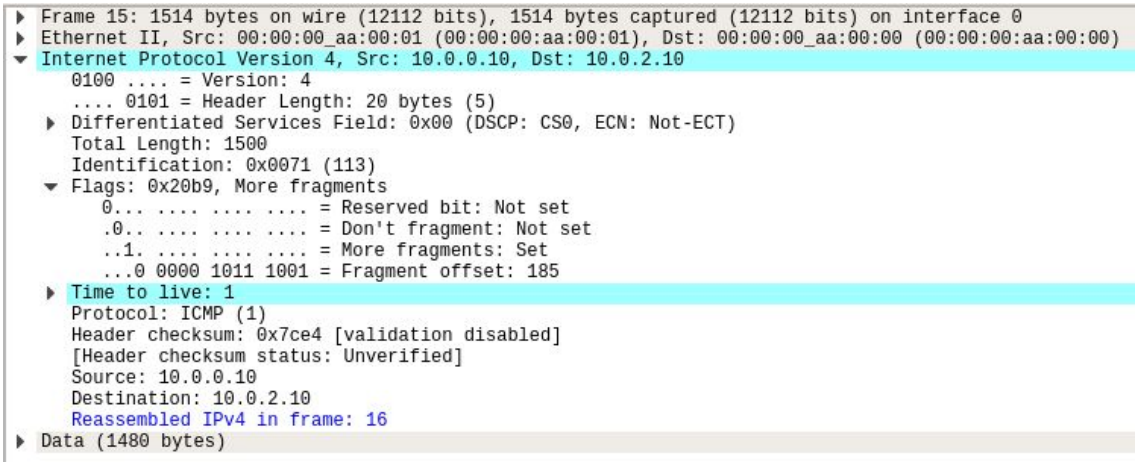
3.b) Imprima o primeiro fragmento do datagrama IP segmentado. Que informação no cabeçalho indica que o datagrama foi fragmentado? Que informação no cabeçalho IP indica que se trata do primeiro fragmento? Qual é o tamanho deste datagrama IP?

A flag *More Fragments* é igual a 1.
Fragment offset é 0.
O tamanho do datagrama IP é de 1514 bytes.



3.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?

Fragment offset agora é 185.
Há mais fragmentos pois a flag *more fragments* é igual a 1.



3.d) Quantos fragmentos foram criados a partir do datagrama original? Como se detecta o último fragmento correspondente ao datagrama original?

Foram criados 3 fragmentos, podemos detectar o último fragmento quando a flag *more fragments* é 0.

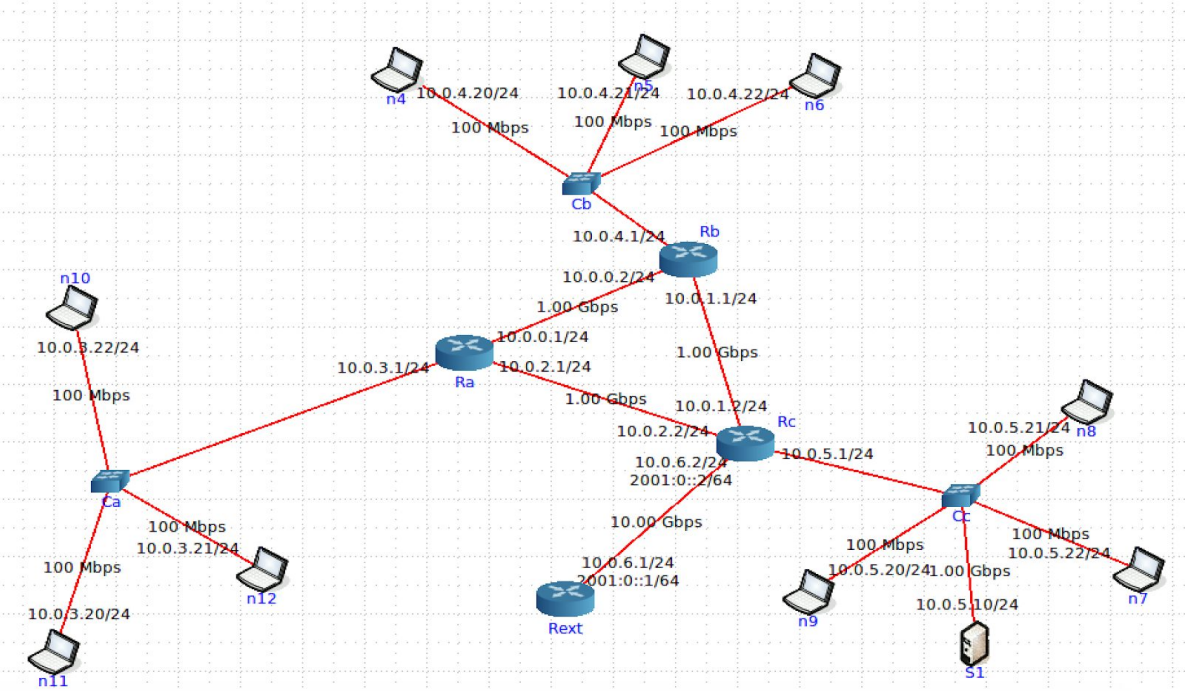
3.e) Indique, resumindo, os campos que mudam no cabeçalho IP entre os diferentes fragmentos, e explique a forma como essa informação permite reconstruir o datagrama original.

A flag *Fragment offset* muda entre os fragmentos, e de acordo com o seu valor é possível reconstruir o pacote pela ordem correta. Nas imagens usadas para demonstrar o pacote IP fragmentado, no primeiro fragmento a flag fragment offset é 0, no segundo é 185 e no terceiro e último é 370.

PARTE 2

1.

1.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.



1.b) Tratam-se de endereços públicos ou privados? Porquê?

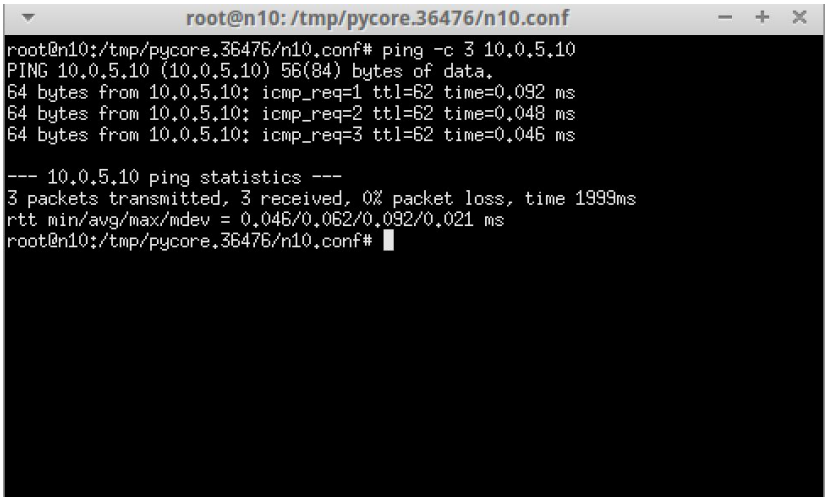
Tratam-se de endereços privados visto serem todos de classe A sendo que os seus endereços começam todos por 10.0._… , o que seria esperado visto estes fazerem parte de uma rede local.

1.c) Porque razão não é atribuído um endereço IP ao *switches*?

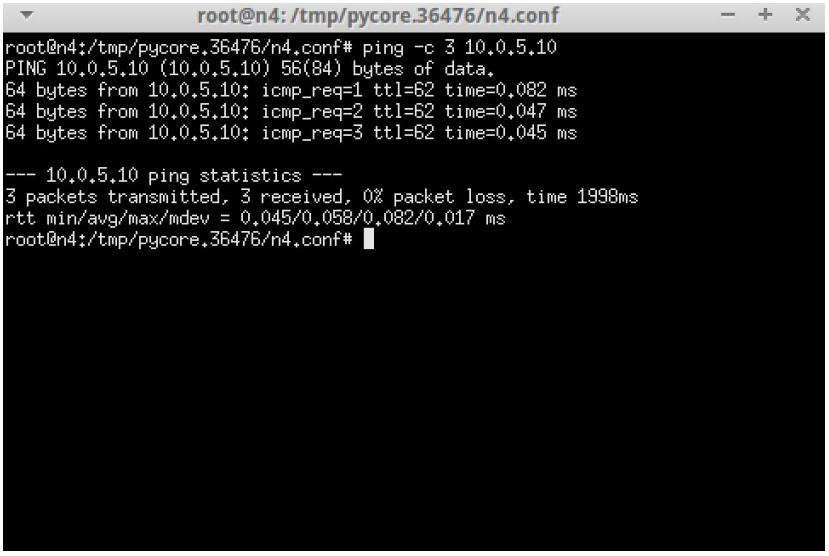
Pois os switches só implementam até a camada 2 (Data Link) do modelo OSI e os endereços IP são atribuídos na camada 3.

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

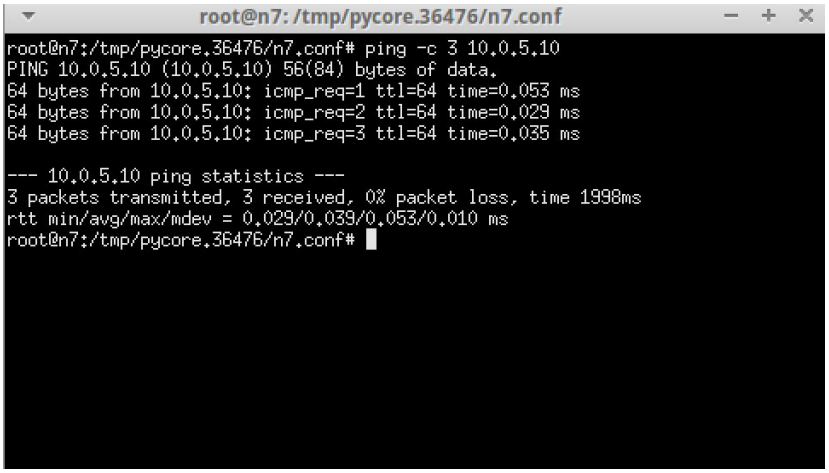
Departamento A (entre o laptop n10 e o servidor S1)



Departamento B (entre o laptop n4 e o servidor S1)

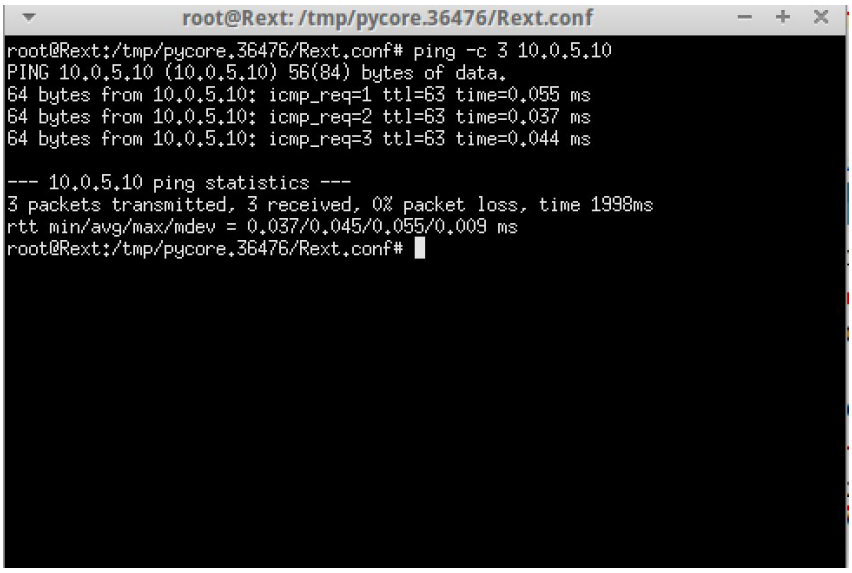


Departamento C (entre o laptop n7 e o servidor S1)



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

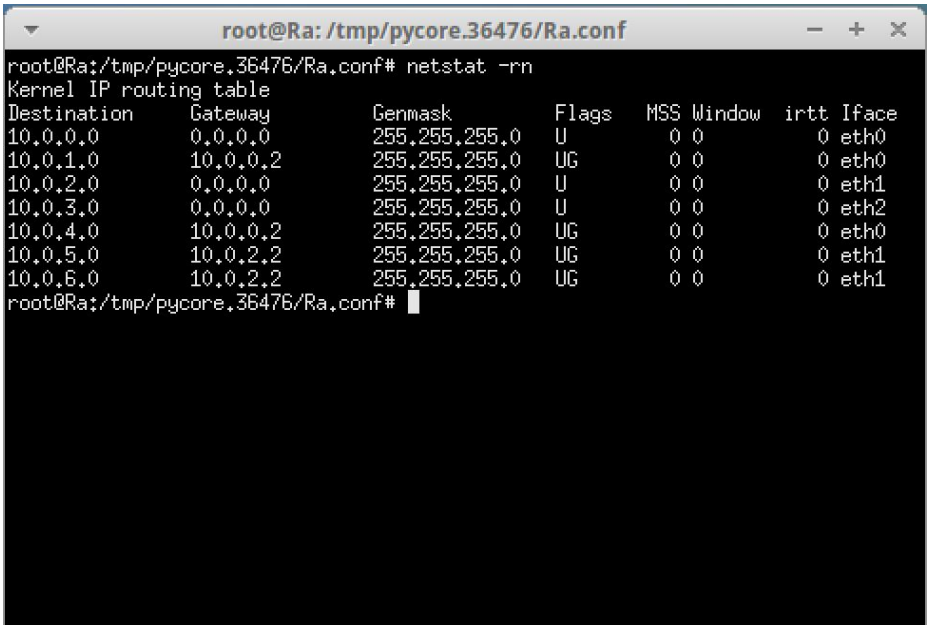
Ping entre o router Rext e os Servidores S1

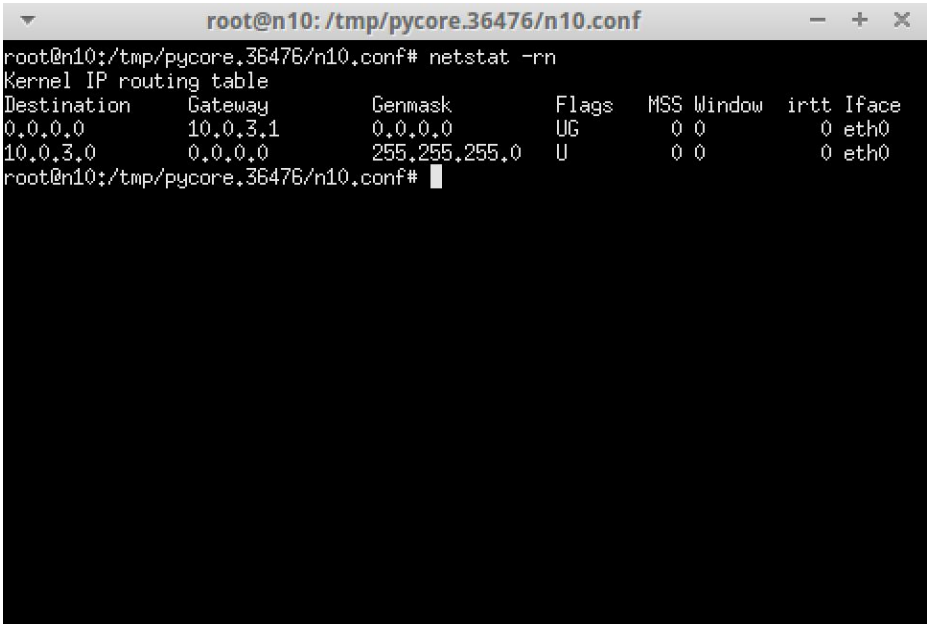


2.

2.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 respectivo (man netstat).

A coluna Destination diz respeito aos endereços que os datagramas têm como destino, em que 0.0.0.0 significa qualquer endereço que não faz parte da rede em que o host está ou que não há uma rota; a coluna Gateway diz respeito a endereços onde passam os datagramas; a coluna Genmask indica a máscara da rede respetiva; a coluna Flags indica a flag dos diferentes datagramas enviados; a coluna Iface indica a interface local por onde estes foram enviados.

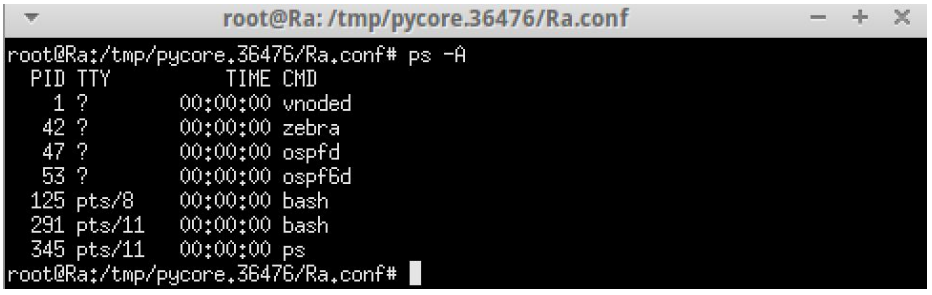


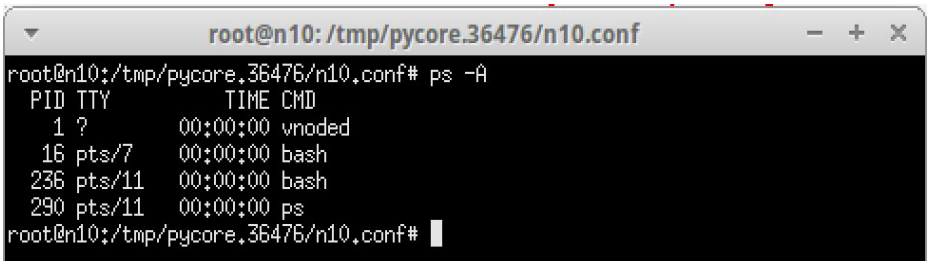


2.b) Diga, justificando, se está a ser usado encaminhamento estático ou dinâmico (sugestão: analise que processos estão a correr em cada sistema).

Recorrendo ao comando *ps -A*, constatamos que:

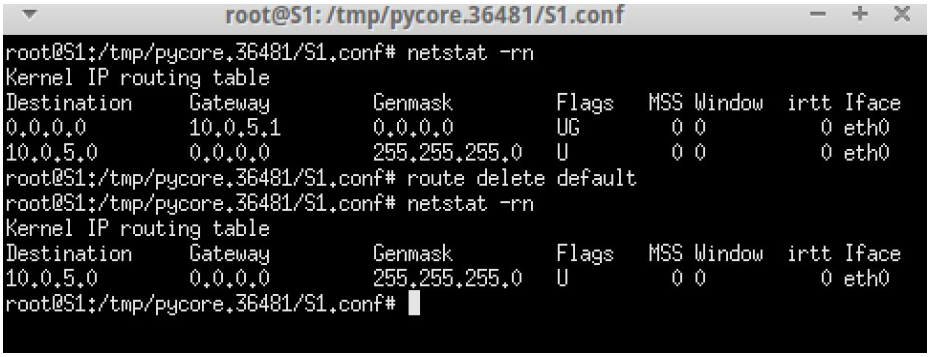
No router, os processos 47 e 53 mostram que está a ser usado uma forma de encaminhamento dinâmico. Constatamos isso na coluna CMD que nos diz que o router inclui o protocolo OSPF (*Open Shortest Path First*). No laptop está a ser usado encaminhamento estático, não sendo observáveis processos relativos a encaminhamento dinâmico.



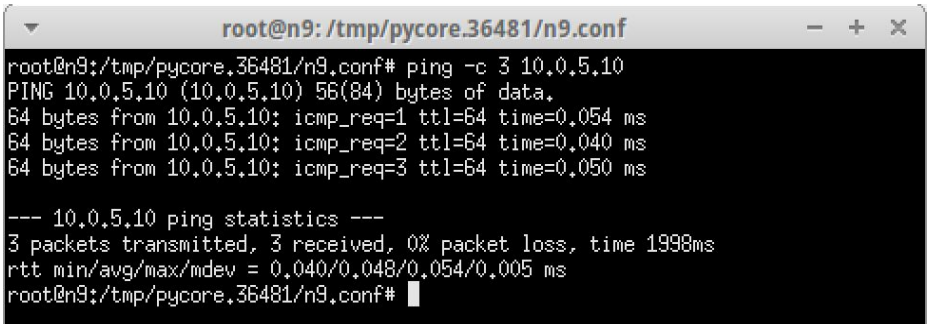


2.c) Admita que, por questões administrativas, a rota por defeito (0.0.0.0 ou *default*) deve ser retirada definitivamente da tabela de encaminhamento do servidor S1 localizado no departamento C. Use o comando route delete para o efeito. Que implicações tem esta medida para os utilizadores da empresa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

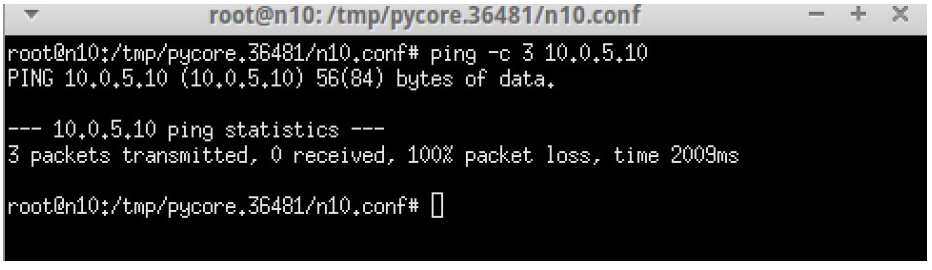
Usando o comando *route delete default,* obtemos a tabela de encaminhamento ficando apenas com uma entrada, a que garante a conectividade interna do departamento C.



Assim sendo, os utilizadores da empresa que acedem ao servidor a partir do departamento C não serão afetados (como podemos ver na figura abaixo).

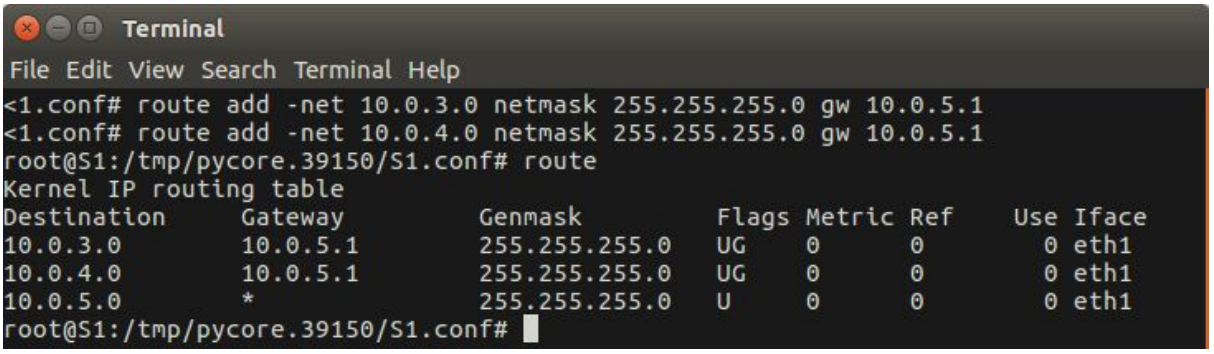


No entanto, fora do departamento C, estamos a cortar a possibilidade do servidor S1 se conectar com outros equipamentos fora da rede deste departamento.



2.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.

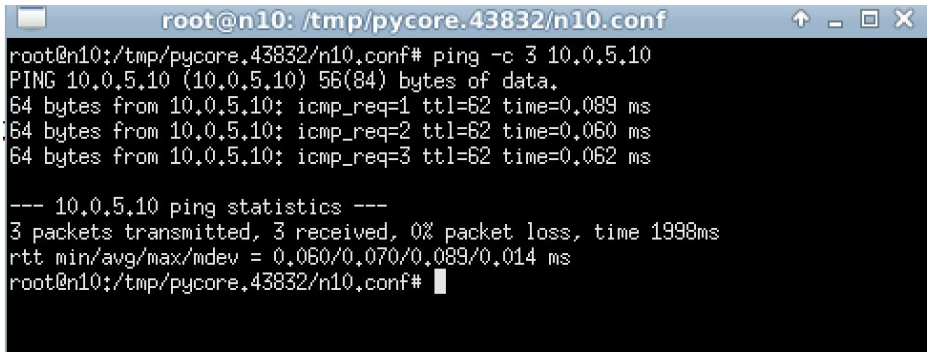
Para restaurar a conectividade entre S1 e os restantes departamentos é necessário incluir uma entrada na sua tabela de encaminhamento, por exemplo o primeiro comando serve para indicar que um pacote que tenha como destino a sub-rede 10.0.4.0 tem que ser encaminhado para 10.0.5.1. O mesmo se aplica à sub-rede 10.0.3.0.



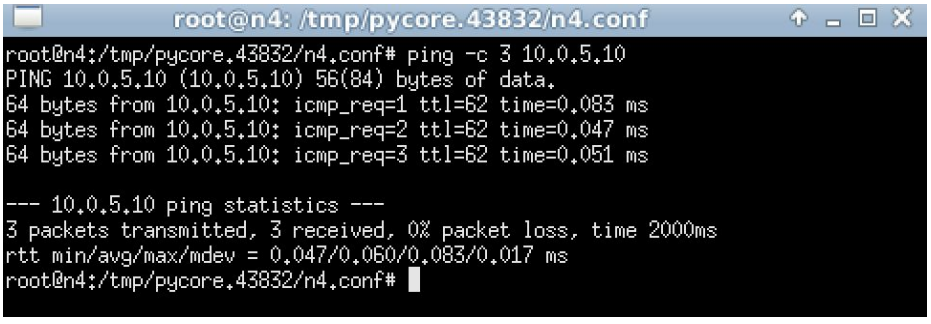
2.e) Teste a nova política de encaminhamento garantindo que o servidor está novamente acessível, utilizando para o efeito o comando *ping*. Registe a nova tabela de encaminhamento do servidor.

Verificando a conexão constatamos que o servidor está acessível a todos os departamentos e que a tabela de encaminhamento tem as sub-redes de destino associadas às gateways tal como estabelecemos com os comandos introduzidos na alínea anterior.

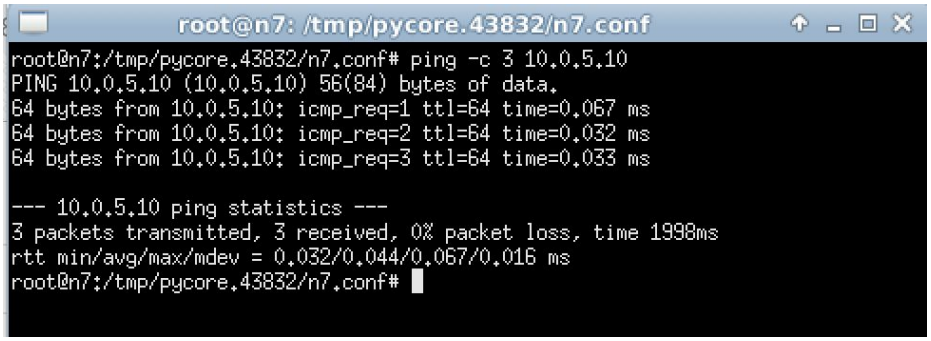
Departamento A (entre o laptop n10 e o servidor S1)



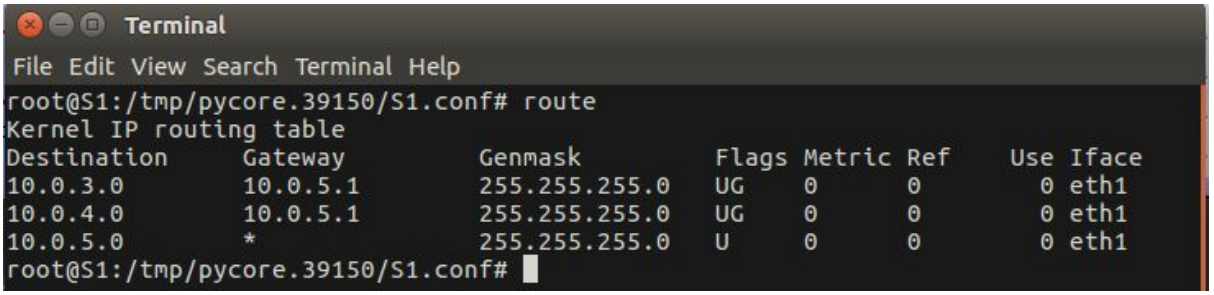
Departamento B (entre o laptop n4 e o servidor S1)



Departamento C (entre o laptop n7 e o servidor S1)

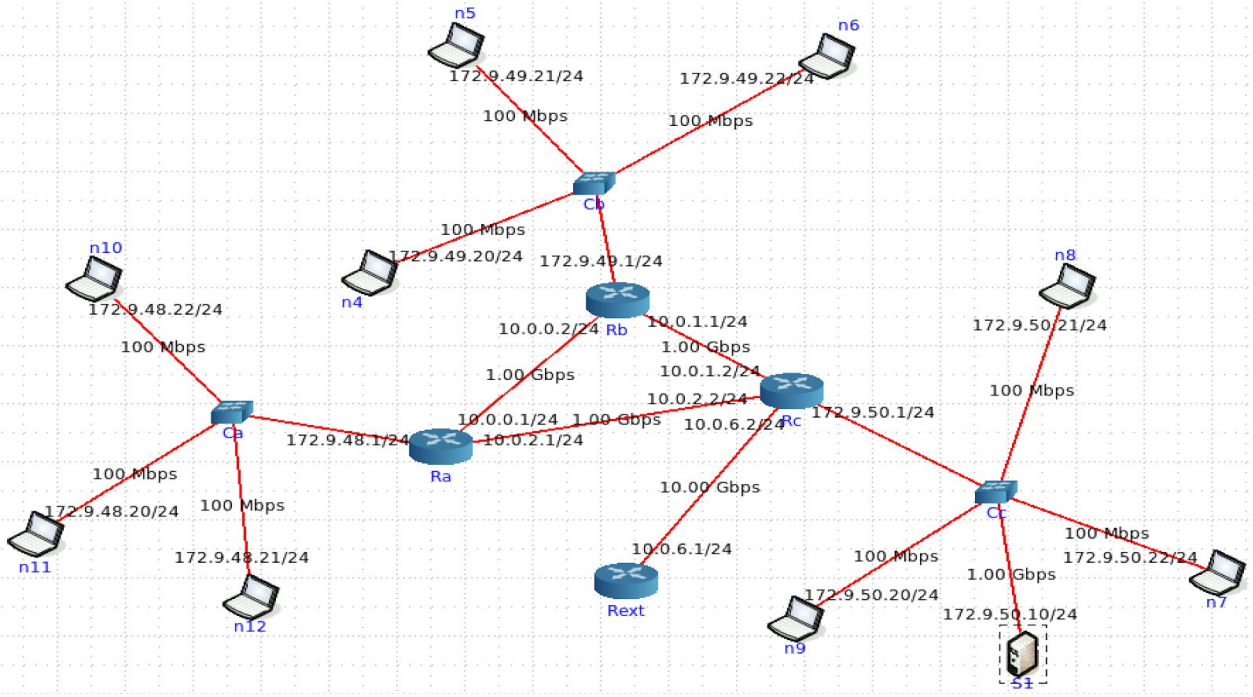


Nova tabela de encaminhamento



3.

3.1) Considere que dispõe apenas do endereço de rede IP 172.XX.48.0/20, em que XX é o decimal correspondendo ao seu número de grupo (PLXX). 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.



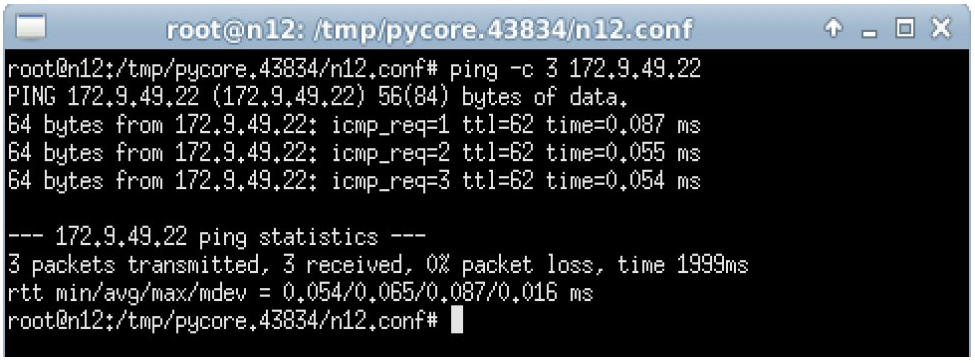
Através do endereço de rede IP 172.9.48.0/20, dispomos de um intervalo de endereços desde 172.9.48.1 até 172.9.63.254.

3.2) Qual a máscara de rede que usou (em formato decimal)? Quantos *hosts* IP pode interligar em cada departamento? Justifique.

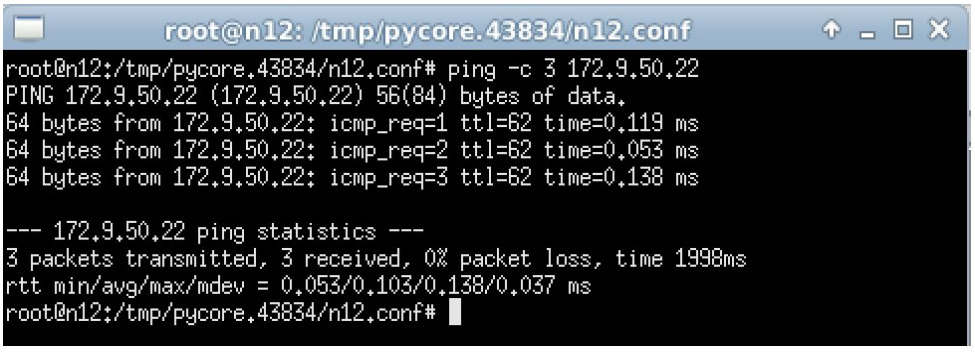
Usamos a máscara de rede 255.255.255.0. Esta máscara permite-nos interligar 254 hosts.

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

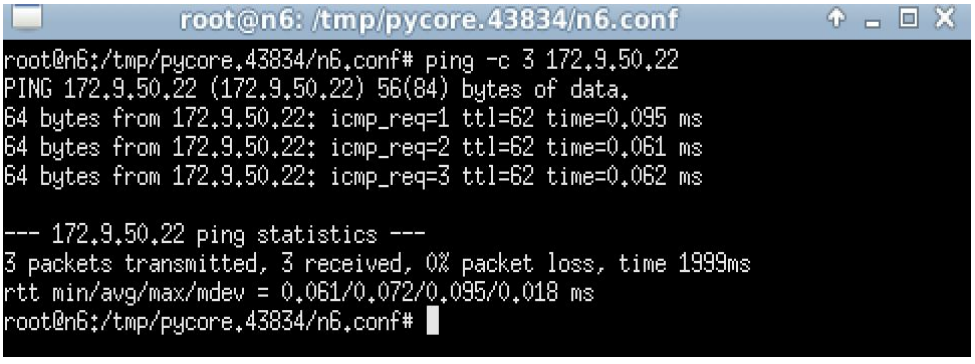
Departamento A e B (entre o laptop n12 e o laptop n6)



Departamento A e C (entre o laptop n12 e o laptop n7)



Departamento B e C (entre o laptop n6 e o laptop n7)



Conclusões

No decorrer da realização deste trabalho foram-nos apresentados vários conceitos novos como o conceito de fragmentação, de endereçamento, de *classless routing*, de sub-rede, entre outros.

Foram-nos apresentado também os programas *Wireshark* e o *Core* que nos permitem a criação e controlo de uma rede "virtual", durante a utilização dos mesmo surgiu a necessidade de comandos como o traceroute, comando que nos permite verificar o tráfego de dados entre dois hosts/routers, o ping, que nos permite verificar a conectividade entre dois host/routers, o netstat, que nos permite consultar a tabela de encaminhamento unicast, o route, que nos permite gerir as diferentes "rotas" da nossa rede, entre outros.

Através da utilização destes programas conseguimos obter uma solução para todos os problemas apresentados, obtendo assim um conhecimento mais aprofundado sobre os diferentes temas apresentados.

No caso do estudo dos conceitos de ICMP e IPv4, na primeira parte do trabalho, aprendemos, através da utilização do comando traceroute, a interpretar datagramas e, através do comando ifconfig, a obter informação sobre as diferentes interfaces de uma rede. Ainda nesta parte foram-nos apresentados problemas relativos a fragmentação de pacotes, como verificar que quando o pacote ultrapassa o *Maximum Transmission Unit* (MTU) existe a necessidade da mesma, basicamente dividindo um pacote em vários, e de uma forma mais generalizada obter informações sobre os diferentes pacotes através das funcionalidades do wireshark.

Passando à segunda parte do trabalho foi-nos requisitada a criação de uma rede mais complexa simulando o funcionamento entre diferentes redes como uma só. Através da verificação de conectividade entre máquinas dos diferentes departamentos, utilizando o comando ping, e de algum conhecimento teórico conseguimos responder ao primeiro tópico desta parte relativo ao endereçamento.

Relativamente ao segundo tópico desta parte utilizámos comandos como o netstat e o route add/delete para controlo e manipulação das rotas entre os diferentes endereços da rede. Durante este processo obtivemos um melhor entendimento relativamente não só ao conceito de rotas de dados mas também a conceitos como o de máscaras de redes e de tabelas de encaminhamento.

No terceiro e último tópico desta parte foi-nos pedido que manipulássemos os endereços das diferentes máquinas, de forma a obter uma melhor organização das redes dos diferentes departamentos, como objetivo de melhor cimentar os conhecimentos adquiridos anteriormente.

Concluindo este trabalho deu-nos a possibilidade de não só testar os nossos conhecimentos relativos a redes de computadores assim como obter uma melhor

compreensão dos mesmos e de novos termos/conceitos com os quais ainda não nos tinhamos deparado.