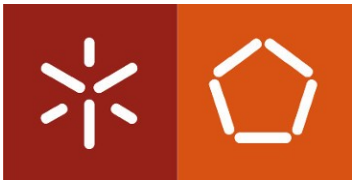


Universidade do Minho
Departamento de Informática
2020/2021

TP2: Protocolo IPv4

Redes de Computadores

25 de Novembro de 2020

Grupo 77

Rita Gomes, A87960

Mário Real, A72720

Mestrado Integrado em Engenharia Informática
Universidade do Minho

Parte I - Datagramas IP e Fragmentação

Exercício 1:

Prepare uma topologia CORE para verificar o comportamento do traceroute. Ligue um host (pc) Cliente1 a um router R2; o router R2 a um router R3, que por sua vez, se liga a um host (servidor) Servidor1. (Note que pode não existir conectividade IP imediata entre o Cliente1 e o Servidor1 até que o anúncio de rotas estabilize). Ajuste o nome dos equipamentos atribuídos por defeito para a topologia do enunciado.

Resposta:

Ao formular a topologia requerida na questão anterior, obteve-se o seguinte resultado:

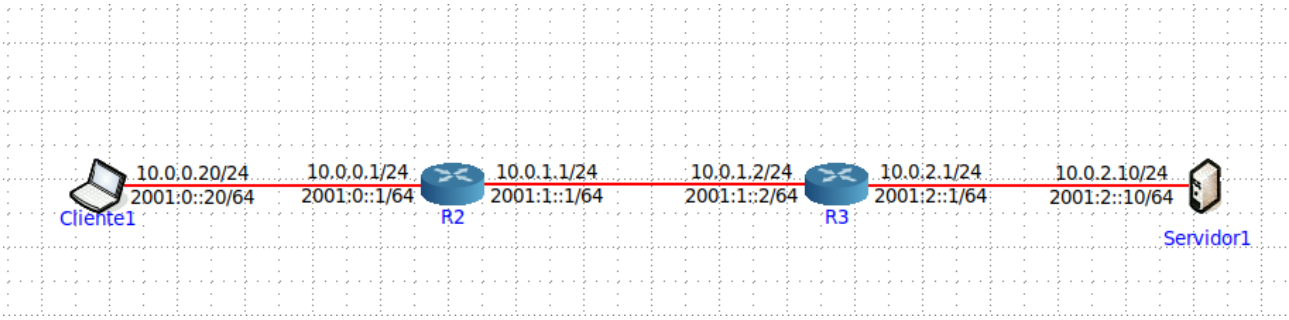


Figura 1: Topologia CORE requerida anteriormente.

a) *Active o wireshark ou o tcpdump no Cliente1. Numa shell do Cliente1, execute o comando traceroute -I para o endereço IP do Servidor1.*

Resposta:

O traceroute informa os vários valores de ida-e-volta para cada um dos três pacotes em cada percurso.

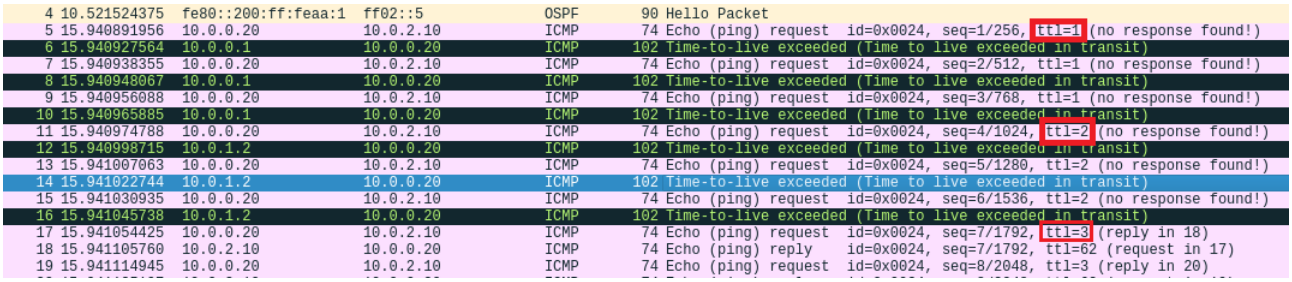
```
root@Cliente1:/tmp/pycore.40459/Cliente1.conf# traceroute -I 10.0.2.10
traceroute to 10.0.2.10 (10.0.2.10), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.177 ms  0.190 ms  0.047 ms
 2  10.0.1.2 (10.0.1.2)  0.096 ms  0.058 ms  0.055 ms
 3  10.0.2.10 (10.0.2.10)  0.095 ms  0.043 ms  0.045 ms
```

Figura 2: Comportamento do traceroute.

b) *Registe e analise o tráfego ICMP enviado pelo Cliente1 e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.*

Resposta:

Após efetuar o comando “traceroute -I 10.0.2.10” e ativar o wireshark no Cliente1, obtemos os seguintes resultados:



4	10.521524375	fe80::200:ff:feaa:1	ff02::5	OSPF	90 Hello Packet
5	15.940891956	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=1/256, ttl=1 (no response found!)
6	15.940927564	10.0.0.1	10.0.0.20	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
7	15.940938355	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=2/512, ttl=1 (no response found!)
8	15.940948067	10.0.0.1	10.0.0.20	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
9	15.940956088	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=3/768, ttl=1 (no response found!)
10	15.940965885	10.0.0.1	10.0.0.20	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
11	15.940974788	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=4/1024, ttl=2 (no response found!)
12	15.940998715	10.0.1.2	10.0.0.20	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
13	15.941007063	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=5/1280, ttl=2 (no response found!)
14	15.941022744	10.0.1.2	10.0.0.20	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
15	15.941030935	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=6/1536, ttl=2 (no response found!)
16	15.941045738	10.0.1.2	10.0.0.20	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
17	15.941054425	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=7/1792, ttl=3 (reply in 18)
18	15.941105760	10.0.2.10	10.0.0.20	ICMP	74 Echo (ping) reply id=0x0024, seq=7/1792, ttl=62 (request in 17)
19	15.941114945	10.0.0.20	10.0.2.10	ICMP	74 Echo (ping) request id=0x0024, seq=8/2048, ttl=3 (reply in 20)

Figura 3: Análise do tráfego ICMP enviado pelo Cliente1 e do tráfego ICMP recebido como resposta.

Como podemos observar na imagem anterior, são enviados três pacotes com TTL=1, TTL=2 e TTL=3, sequencialmente, para o Servidor1, cuja origem é o Cliente1 (10.0.0.20) e cujo destino é o Servidor1 (10.0.2.10). Este envio de pacotes é uma tentativa de comunicação por parte do Cliente1 com o Servidor1.

Inicialmente são enviados 3 pacotes com TTL=1 e outros 3 pacotes com TTL=2, contudo nenhum destes consegue chegar ao S1 e, por isso, o ICMP envia 6 mensagem de erro (102 Time-To-Live Exceeded) e estes são descartados no router r2 e r3, respetivamente.

Finalmente, os pacotes com TTL=3 conseguem chegar ao destino antes do TTL tomar o valor 0.

c) *Qual deve ser o valor inicial mínimo do campo TTL para alcançar o Servidor1? Verifique na prática que a sua resposta está correta.*

Resposta:

Como verificamos na alínea anterior, o TTL (time to live) mínimo para obtermos uma primeira resposta por parte do Servidor1 é 3. Podemos ainda confirmar este resultado observando a figura 1, na qual vemos que só na terceira linha é que atingimos o endereço 10.0.2.10, que corresponde ao endereço do Servidor1.

d) *Calcule o valor médio do tempo de ida-e-volta (Round-Trip Time) obtido?*

Resposta:

O *Round-Trip Time* é o tempo que um pacote demora desde que é enviado até ser recebido por parte do Servidor1, neste caso. Sendo assim, o tempo médio de ida-e-volta será a média dos três últimos valores, visto que só o campo 3 é que corresponde ao pacote de respostas. Desta forma, o valor obtido é (0.095+0.043+0.045)/3 = 0.061ms.

Exercício 2:

Pretende-se agora usar o traceroute na sua máquina nativa, e gerar de datagramas IP de diferentes tamanhos.

Procedimento a seguir: Usando o wireshark capture o tráfego gerado pelo traceroute para os seguintes tamanhos de pacote: (i) sem especificar, i.e., usando o 3 tamanho por defeito; e (ii) 32XX bytes, em que XX é o seu número de grupo. Utilize como máquina destino o host marco.uminho.pt. Pare a captura. Com base no tráfego capturado, identifique os pedidos ICMP Echo Request e o conjunto de mensagens devolvidas como resposta. Selecione a primeira mensagem ICMP capturada (referente a (i) tamanho por defeito) e centre a análise no nível protocolar IP (expanda o tab correspondente na janela de detalhe do wireshark).

Resposta:

Para a resolução deste exercício, a máquina nativa utilizada foi Linux.

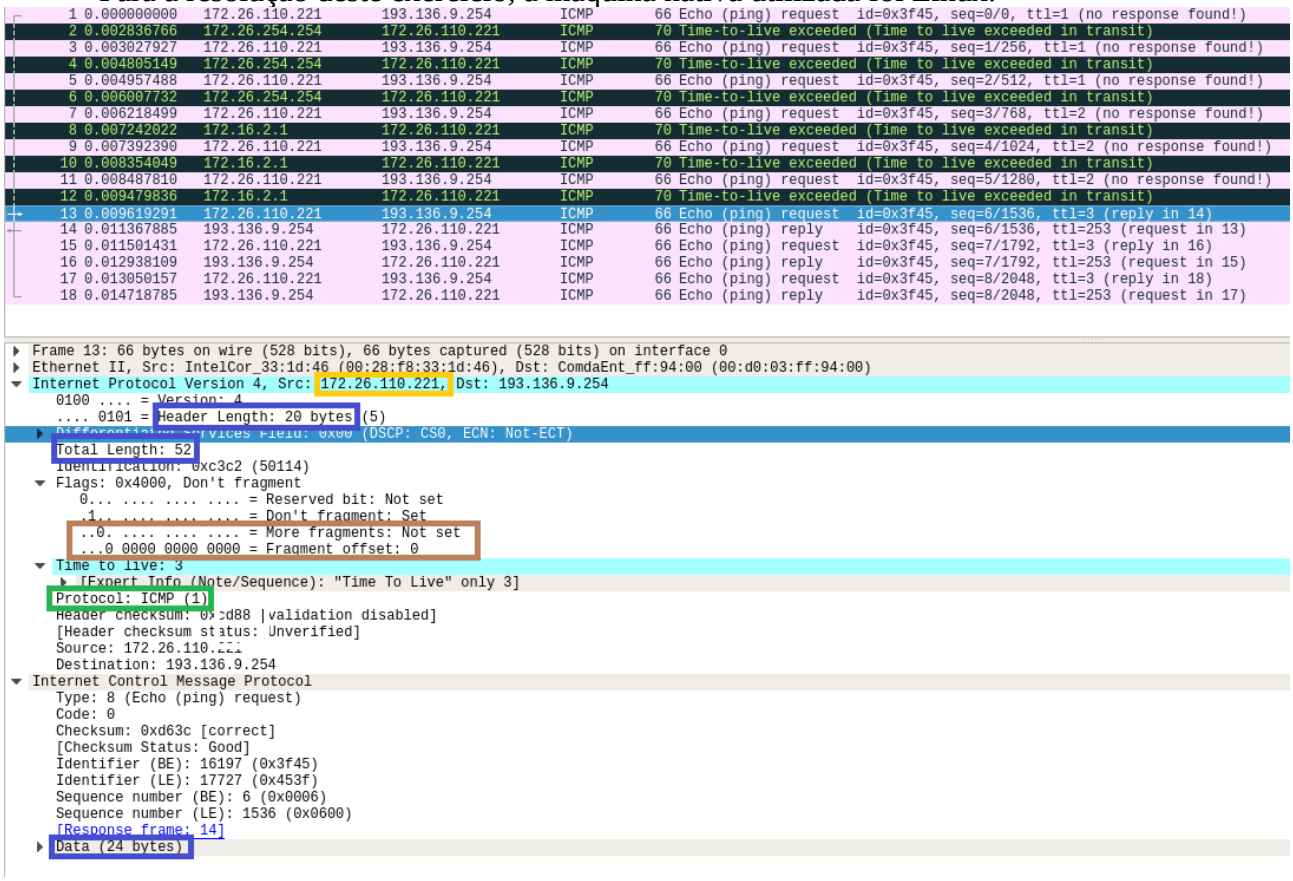


Figura 4: Tráfego gerado pelo traceroute usando o tamanho por defeito.

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

Resposta:

O endereço IP da máquina onde foi executado o traceroute é 172.26.110.221, tal como podemos observar na região a amarelo da imagem acima, no campo *Source*.

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

Resposta:

Como podemos observar na figura 3, na área selecionada a verde, o valor do campo protocolo é 1, correspondente ao protocolo ICMP ("Internet Control Message Protocol"), utilizado para fornecer relatórios de diagnóstico, de erro, e operacionais.

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?*

Resposta:

Para saber por quantos bytes o cabeçalho IP(v4) é constituído basta observar o campo *header length.* Daí concluímos que este valor corresponde a 20 bytes. O campo de dados (payload) do datagrama é 24 bytes, como é possível verificar no último campo a azul, *Data*. Sendo assim, o

payload é calculado através da seguinte expressão: 24 bytes = 52 (total length) – 20 (header length) – 8 (header do ICMP).

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

Resposta:

O datagrama IP não foi fragmentado. Como podemos observar na região a castanho da figura 3, a flag “Fragment offset” tem o valor 0, o que indica que está a apontar para o início do datagrama original, ou seja, não há mais nenhum fragmento sem ser o pacote original, que é o próprio. Vemos ainda que na flag “More fragments” está o valor “not set”, que denota a não existência de fragmentos para além do atual.

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.

Resposta:

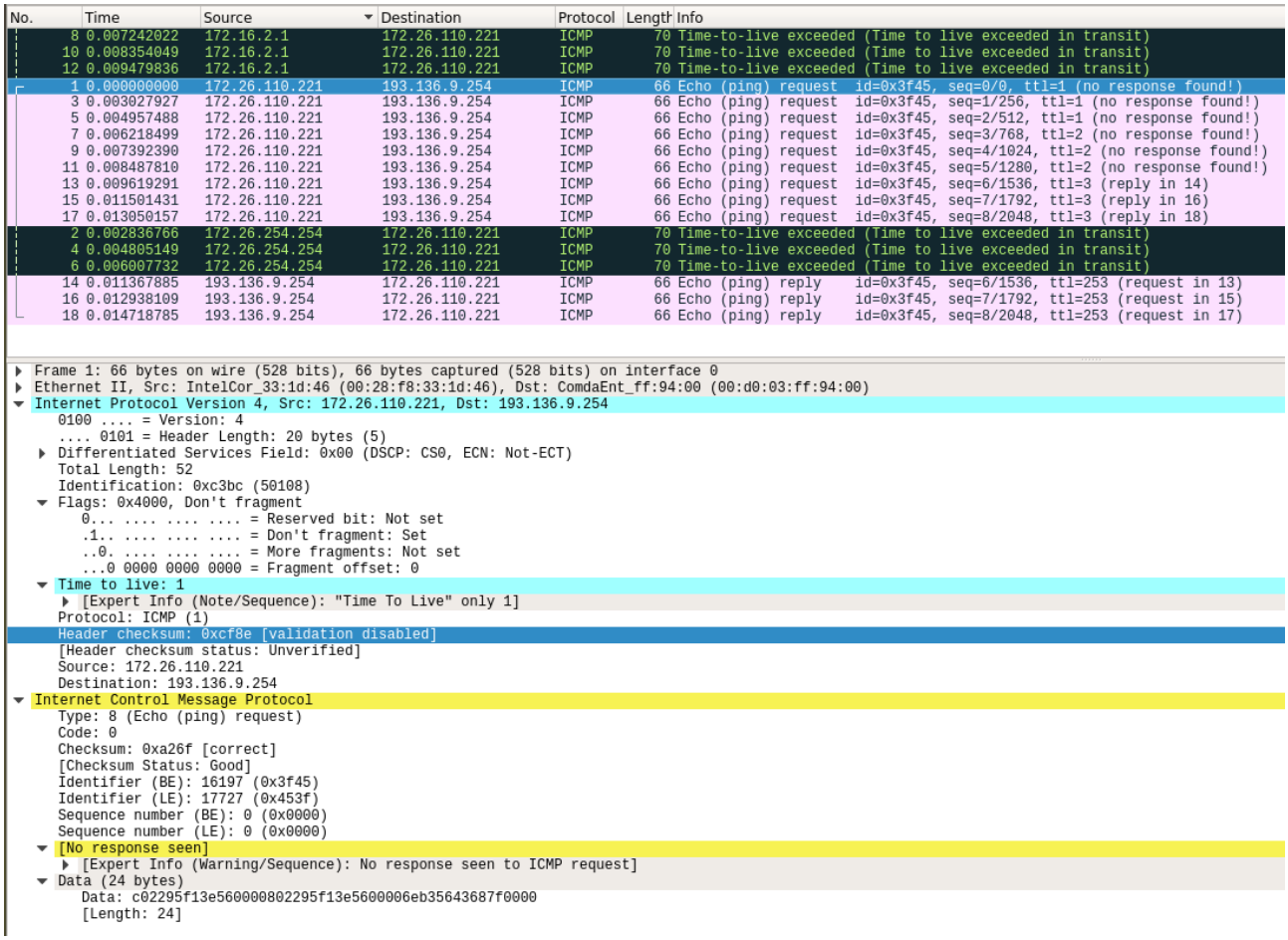


Figura 5: Tráfego organizado por endereço fonte.

Os campos do cabeçalho IP que variam de pacote para pacote são: campo de identificação, TTL e Header checksum. O campo de identificação muda porque cada pacote é diferente, logo, tem de ser identificado diferenciadamente. O TTL incrementa sempre a cada iteração um valor e o

Header checksum é um detetor de erros e por cada nó que o IPv4 passa tem de recalcular o checksum.

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

Resposta:

Os valores do campo de identificação aumentam sequencialmente um valor a cada pacote enviado/recebido. O TTL aumenta um valor a cada 3 pacotes, isto é, 3 pacotes com o TTL=1, 3 pacotes com o TTL=2 e 3 pacotes com o TTL=3.

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?*

Resposta:

O valor do campo TTL na série de respostas ICMP com o TTL exceeded varia de 255 para 254, como podemos observar nas imagens seguintes:

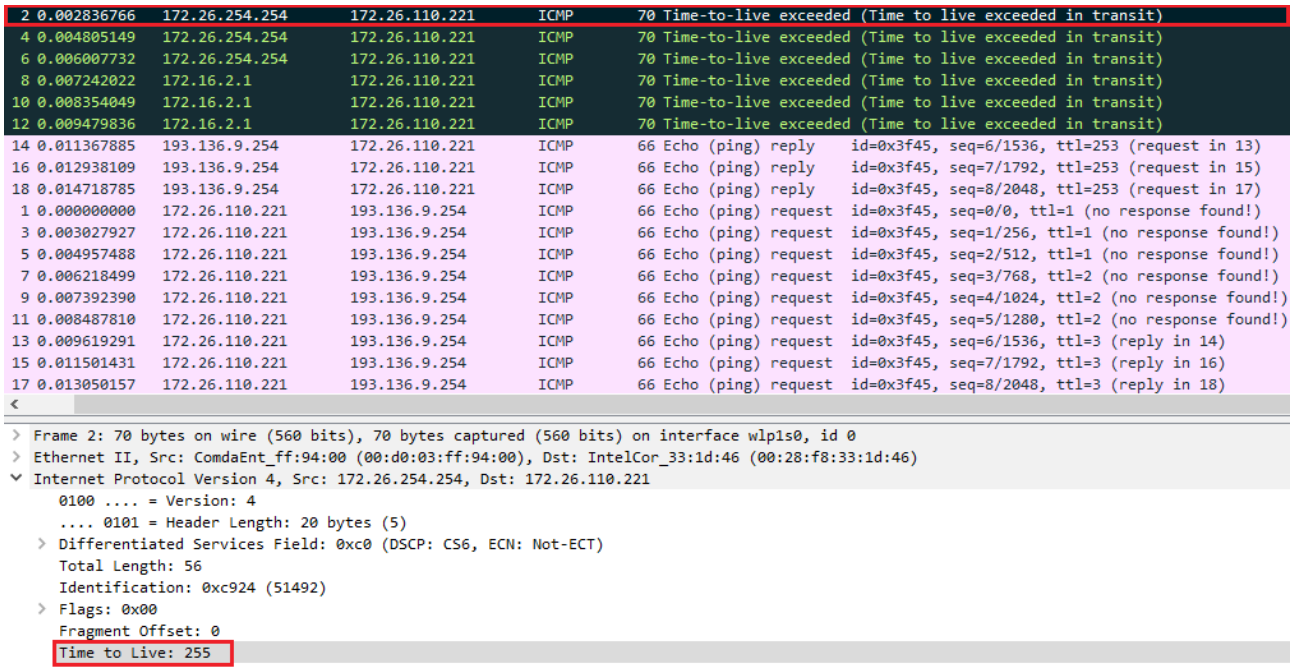


Figura 6: Tráfego organizado por endereço destino (TTL=255).

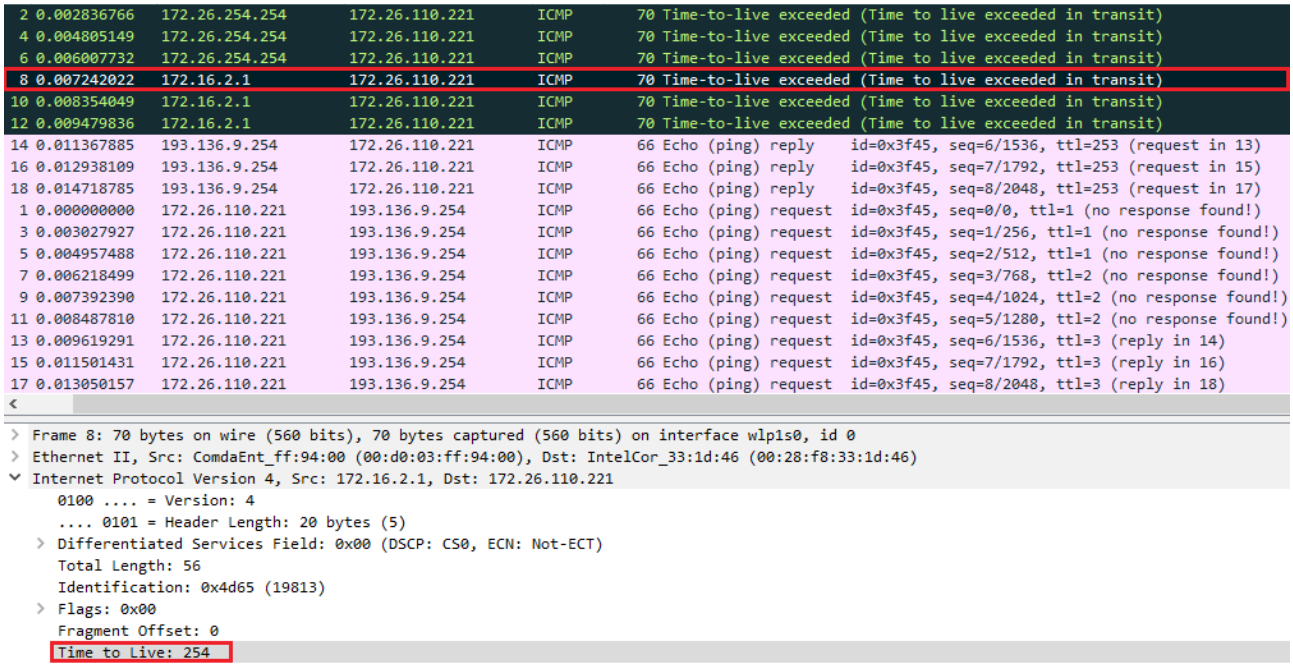


Figura 7: Tráfego organizado por endereço destino (TTL=254).

Conseguimos observar ainda que o TTL permanece constante dentro da mesma source, ou seja, na source 172.26.110.221 o TTL está a 255 e na source 172.16.2.1 está a 254. O valor do TTL passa de 255 para 254 porque passou por um nó intermédio até chegar ao nosso computador.

Exercício 3:

Pretende-se agora analisar a fragmentação de pacotes IP. Reponha a ordem do tráfego capturado usando a coluna do tempo de captura. Observe o tráfego depois do tamanho de pacote ter sido definido para 32XX bytes.

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

Resposta:

O tamanho máximo que se pode enviar num só pacote são 1500 bytes, visto que é a capacidade máxima de um datagrama. Como se pretende enviar 3277 bytes de informação não é possível enviar tudo de uma vez, logo é necessário fragmentar o pacote, neste caso em 3 fragmentos.

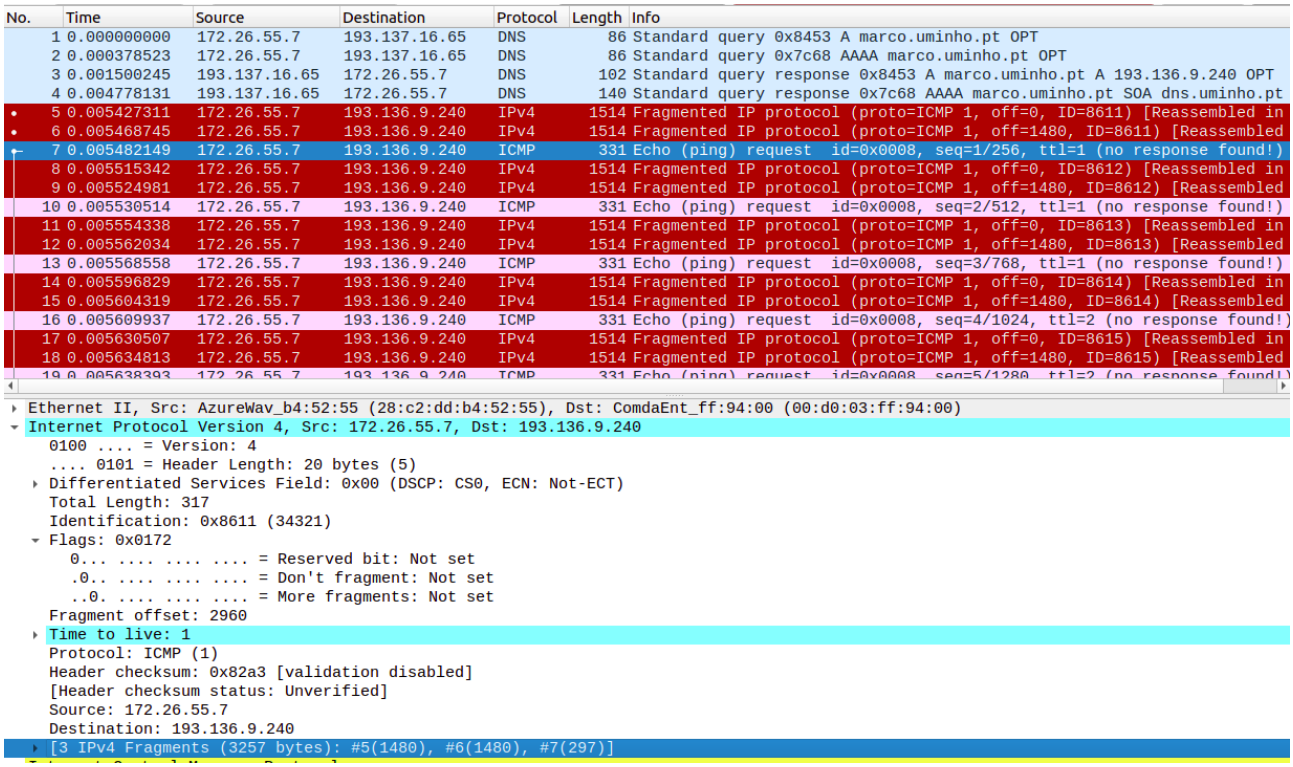


Figura 8: Fragmentação dos pacotes IP.

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?*

Resposta:

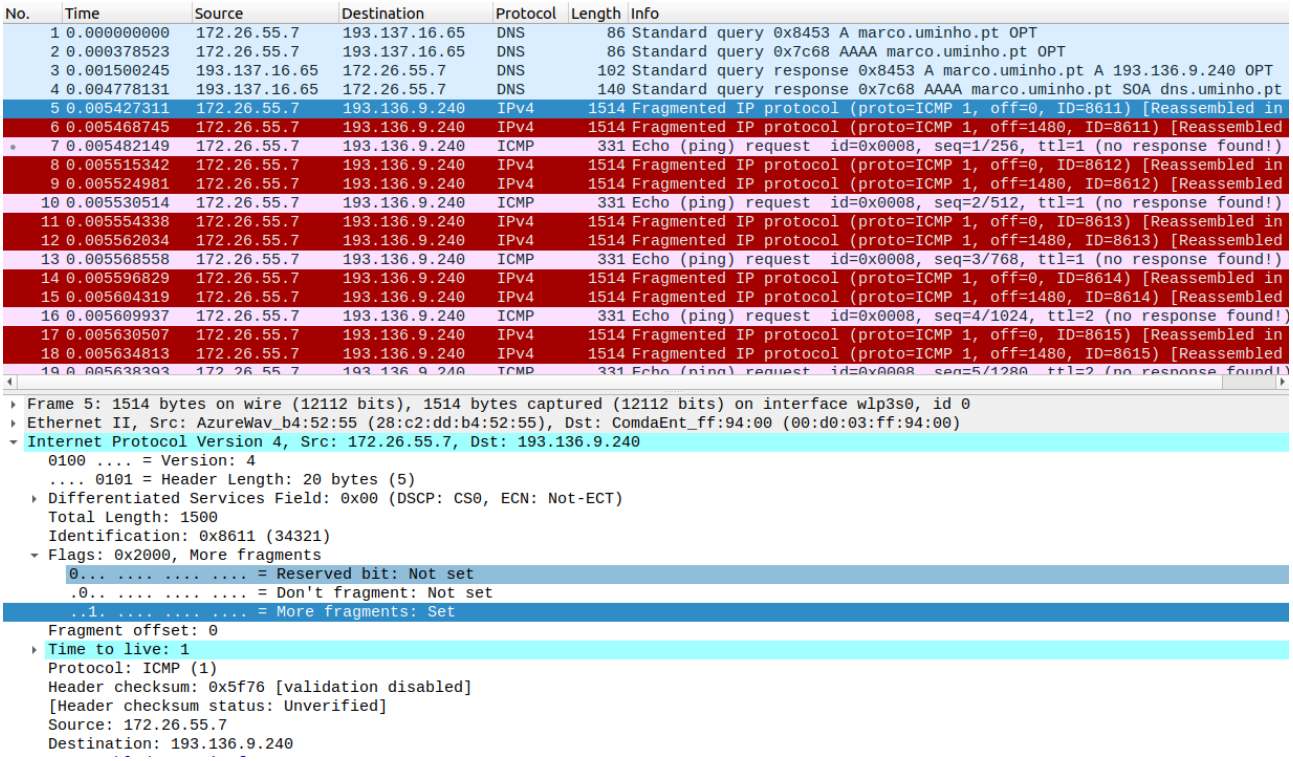


Figura 9: Primeiro fragmento do datagrama IP segmentado.

A flag “More Fragments” indica-nos se existem ou não mais fragmentos. Visto que esta flag está a 1, concluímos que existem mais fragmentos. A flag “Fragment offset” especifica a que parte do pacote corresponde este fragmento. Sendo que esta está a 0, então trata-se do primeiro fragmento. O campo “Total length” mostra-nos o tamanho total do pacote que é 1500 bytes.

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?*

Resposta:

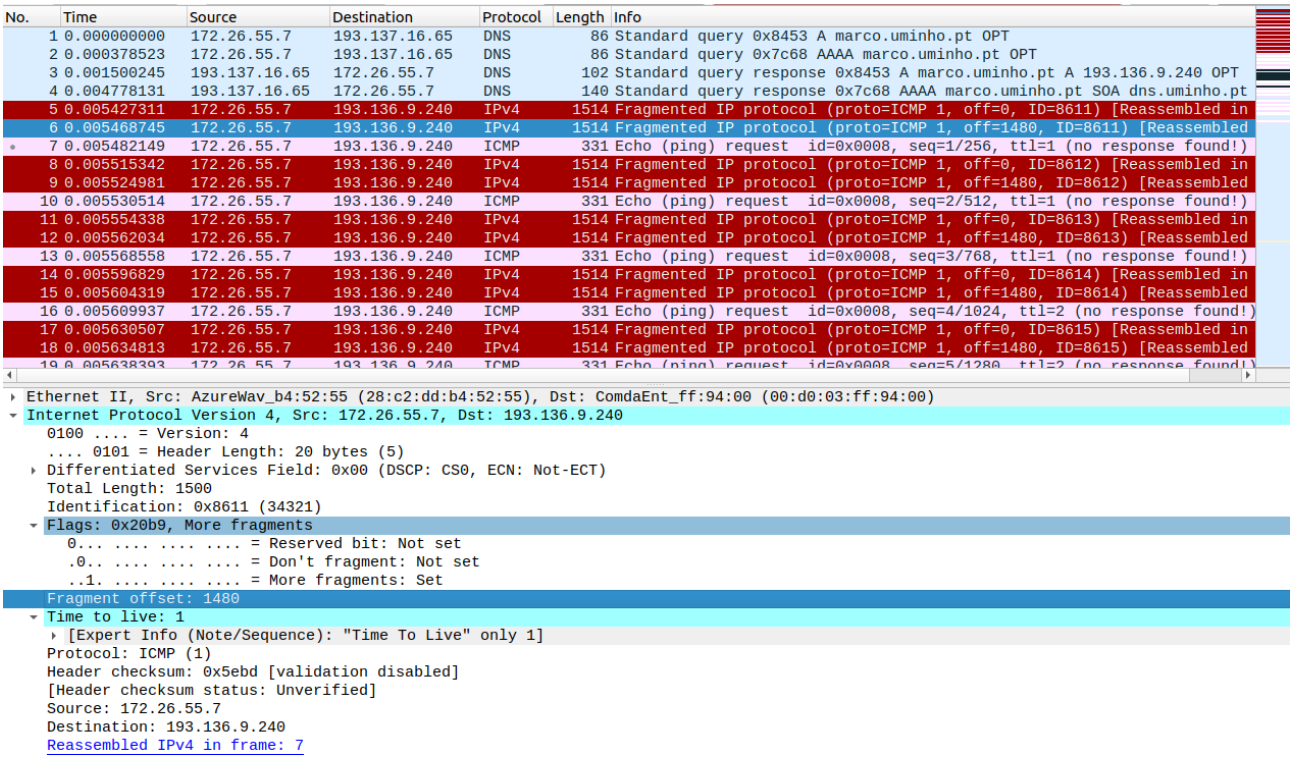


Figura 10: Segundo fragmento do datagrama IP segmentado.

Neste caso, a flag “Fragment offset” está a 1480, o que significa que este fragmento não é o primeiro, pois caso o fosse esta flag estaria a 0 como vimos na alínea anterior. Visto que a flag “More fragments” está a 1, concluímos que este não é o último fragmento.

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

Resposta:

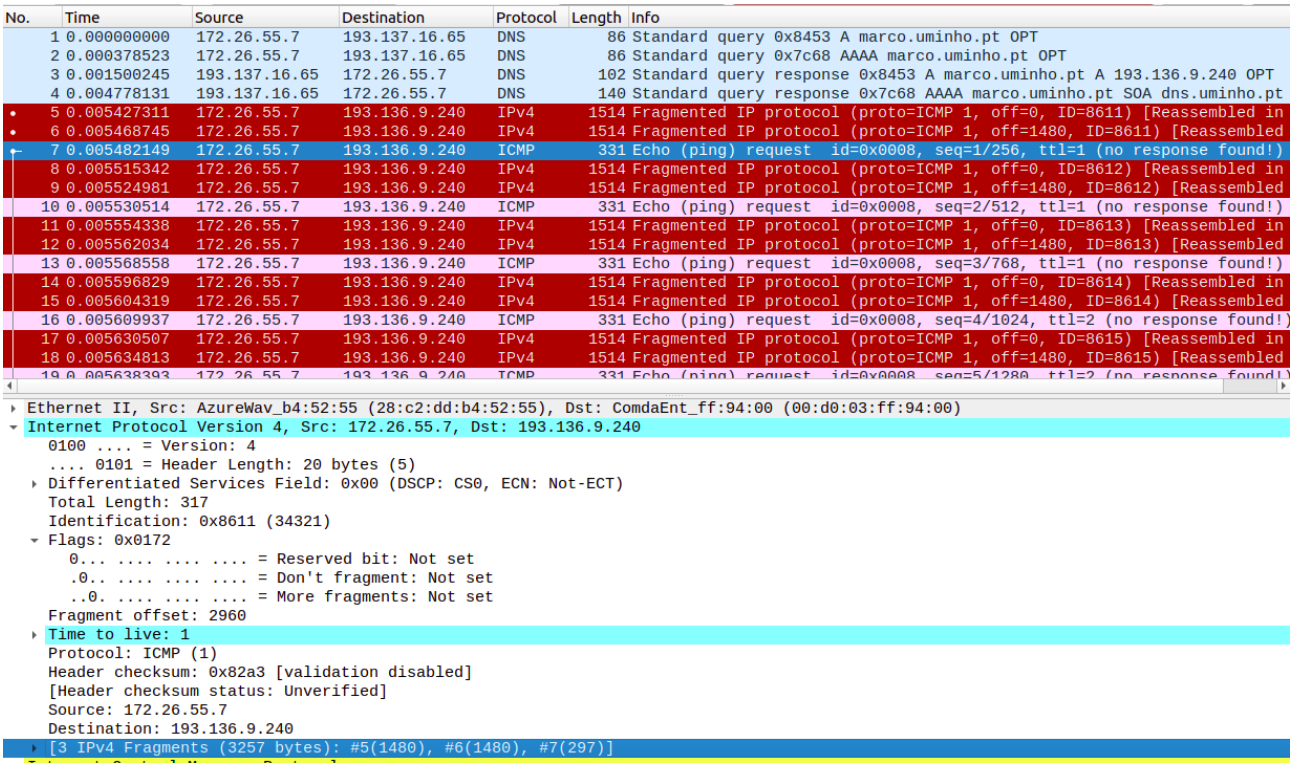


Figura 11: Fragmentos criados a partir do datagrama original.

Como podemos observar no final da imagem acima, foram criados 3 fragmentos a partir do datagrama original. Através do campo “Identification” conseguimos observar que todos estes fragmentos têm um número de identificação igual a 0x8611, daí concluímos que pertencem todos ao mesmo datagrama. Visto que a flag “More fragments” está a 0, então concluímos que este é o último fragmento.

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.*

Resposta:

As flags “Fragment offset” e “More fragments” são os únicos campos que se alteram no cabeçalho IP entre os diferentes fragmentos. A “Fragment offset” indica a posição do fragmento no datagrama original, sendo que o seu valor irá ser sucessivamente mais elevado, consoante a sua posição como fragmento no datagrama e baseado nisso o pacote será reconstruído desde a sua posição inicial até ao fim do deslocamento. A “More fragments” indica-nos se há ou não mais fragmentos.

Parte II – Endereçamento e Encaminhamento IP

Exercício 1:

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

Resposta:

Na imagem seguinte podemos ver os endereços IP atribuídos aos vários equipamentos.

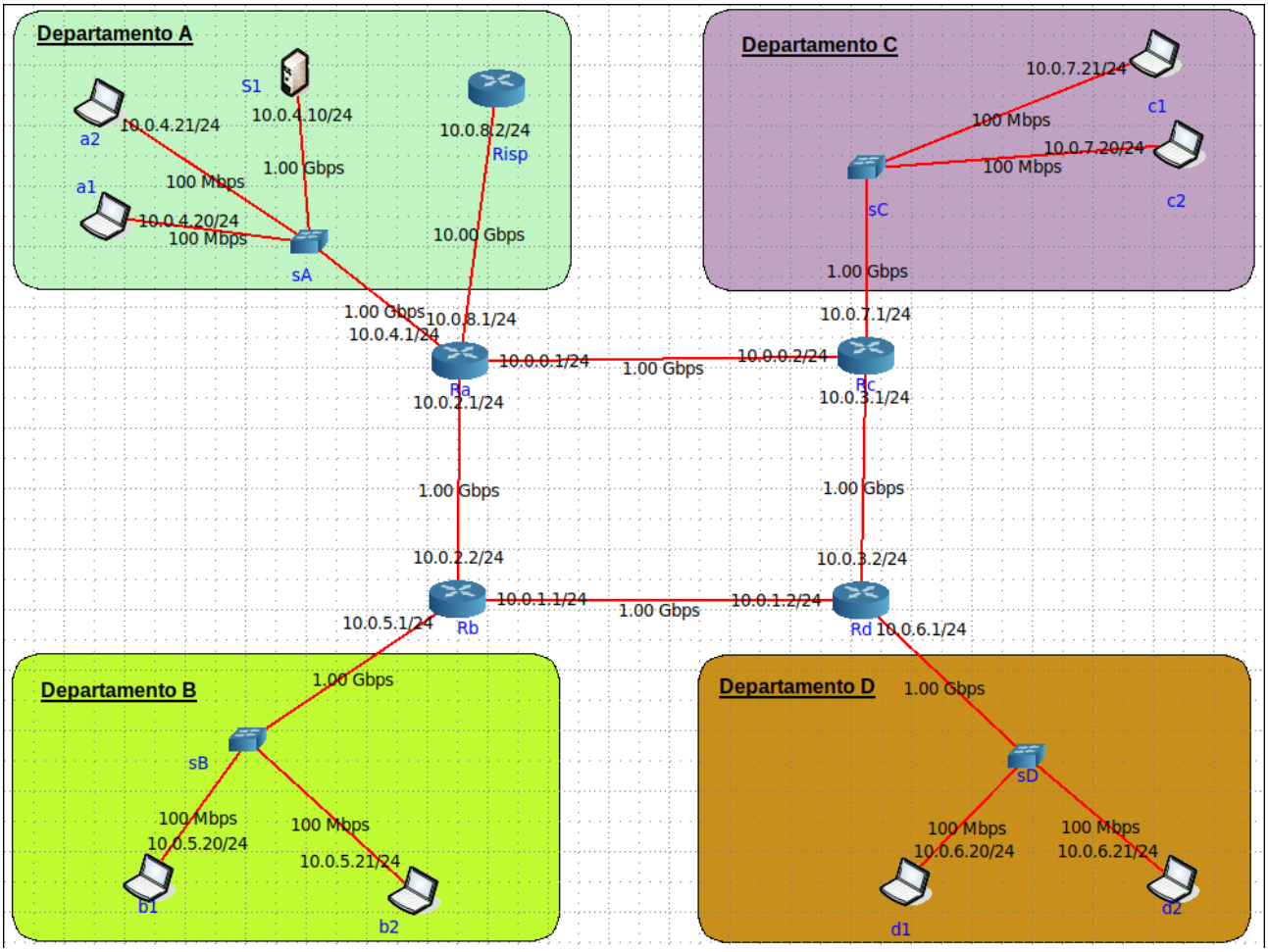


Figura 12: Topologia Core para verificar o comportamento do traceroute.

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.*

Resposta:

Como podemos ver na imagem acima, a máscara utilizada é /24, em notação CIDR, que corresponde a 255.255.255.0.

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

Resposta:

São endereços privados, visto que todos eles estão contidos no intervalo [10.0.0.0; 10.255.255.255], que é uma gama reservada a endereços privados.

c) *Porque razão não é atribuído um endereço IP aos switches?*

Resposta:

Não são atribuídos endereços IP's aos switches, visto que os switches atuam apenas no nível 2, camada de Ethernet, e não existem IP adresses ao nível de packets de Ethernet, operando apenas sobre o seu MAC address.

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).*

Resposta:

A existência de conectividade implica a existência de um caminho de ida e volta. Sendo assim, as figuras a seguir comprovam que o servidor S1 está ligado a todos os outros laptops, uma vez que os pacotes enviados conseguem chegar ao destino.

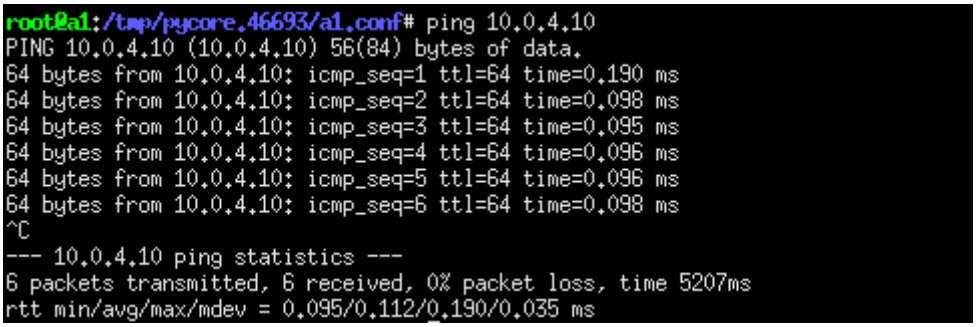


Figura 13: Conectividade entre S1 e um laptop do departamento A.

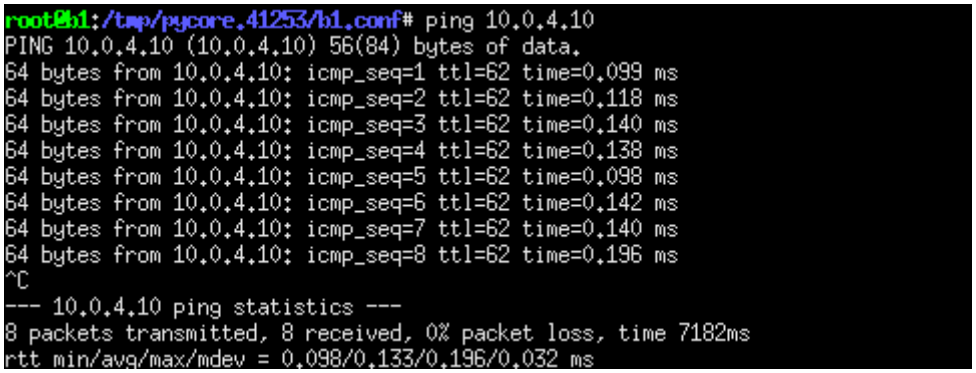


Figura 14: Conectividade entre S1 e um laptop do departamento B.

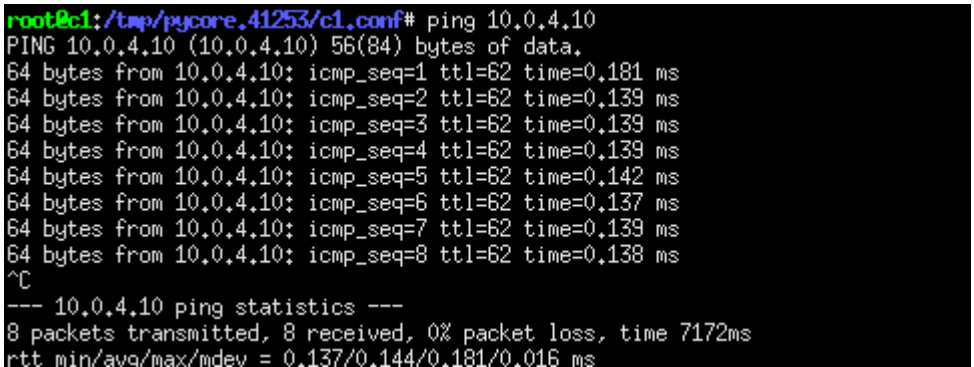


Figura 15: Conectividade entre S1 e um laptop do departamento C.

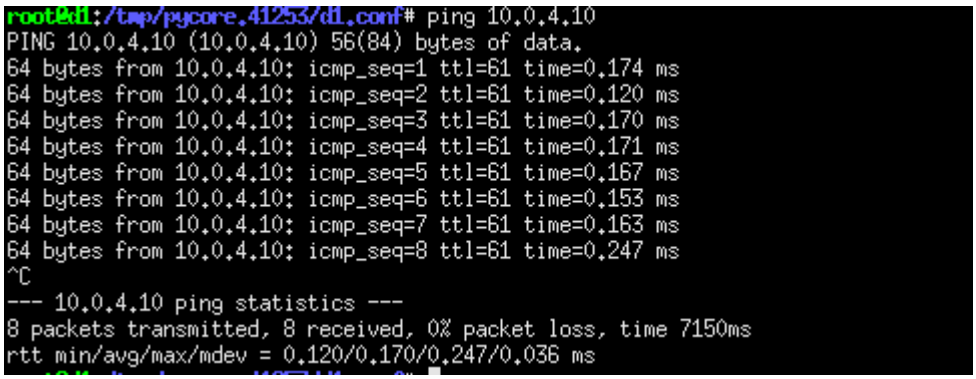


Figura 16: Conectividade entre S1 e um laptop do departamento D.

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

Resposta:

Como se verifica na imagem a seguir, existe conectividade entre o router externo, router RISP, e o servidor S1, sendo que todos os pacotes enviados foram recebidos.

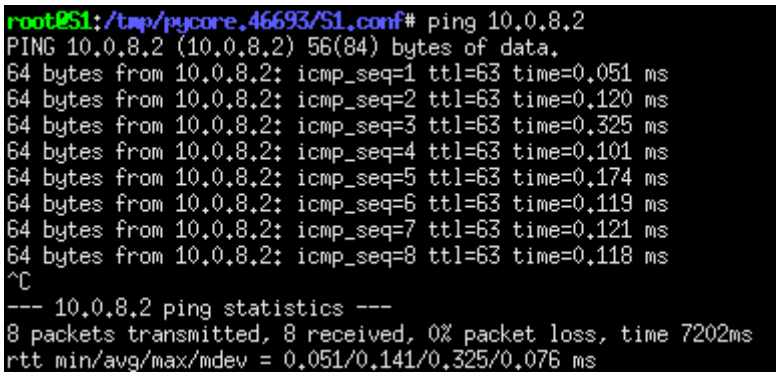


Figura 17: Conectividade entre Risp e o servidor S1.

Exercício 2:

Para o router e um laptop do departamento C:

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).*

Resposta:

Nas tabelas de encaminhamento geradas conseguimos obter informações tais como: "Destination" (sub-redes de destino), "Gateway" (indicação do próximo salto, onde podemos saber por qual equipamento adjacente ao distinto, o nosso pacote terá de passar), "Genmask" tipo de máscara usada, entre outras…

```
root@Rc:/tmp/pycore.41253/Rc.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        10.0.3.2        255.255.255.0   UG        0 0          0 eth1
10.0.2.0        10.0.0.1        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
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        10.0.3.2        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        10.0.0.1        255.255.255.0   UG        0 0          0 eth0
```

Figura 18: Tabela de encaminhamento do router C.

Como conseguimos observar na tabela referente ao router do departamento **C (‘Rc’),** nas linhas/entradas onde o *Gateway* devolve o valor 0.0.0.0. (correspondente ao próprio router), estão representadas **3** ***Destination***(s) (10.0.0.0; 10.0.3.0; 10.0.7.0) que são as redes que, respetivamente, interligam o Router **C** ao **Ra** (router A)**, Rd (**Router D) e **Departamento C** (Rc, sC, c1 e c2).

Já na segunda linha/entrada podemos observar que temos como destino a rede entre **Rb** e **Rd** (10.0.1.0), e como próximo salto a interface de endereço (10.0.3.2) (**Rd)** indicada no *Gateway*, constituindo assim o caminho mais próximo entre o destino pretendido e o Router **C**. Todas as outras entradas da tabela seguem o mesmo raciocino.

```
root@c1:/tmp/pycore.41253/c1.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
```

Figura 19: Tabela de encaminhamento para o laptop do departamento C.

Como podemos observar na tabela referente ao laptop do departamento **C** (‘**c1’)** , a primeira entrada de destino indica-nos a rota por defeito (0.0.0.0), sendo o salto seguinte o router C (‘**Rc’).** Já na entrada seguinte da tabela, tem como destino a rede do Departamento **C** (10.0.0.7.0), onde o próximo salto será o próprio host.

A nossa rota por defeito 0.0.0.0 é usada quando não é conhecido o destino de um pacote.

Já na segunda entrada, o destino 10.0.7.0 é usado quando a partir do laptop tenciona-se mandar um pacote para a própria rede. Isto justifica-se porque sendo o laptop um *end system*, tem apenas capacidade de encaminhamento para sistemas na sua rede local.

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

Resposta:

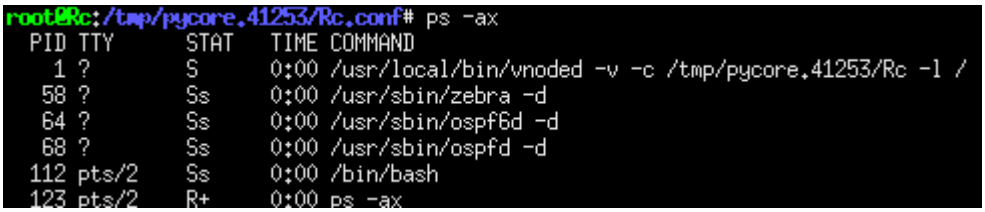


Figura 20: Processos do do router C.

Na imagem acima, após efetuar o comando ps -ax no router C concluímos que está a ser usado encaminhamento dinâmico entre os routers, visto que estes trocam informação de routing entre si e esta atualização dinâmica de rotas é obtida através do protocolo específico de encaminhamento OSPF.

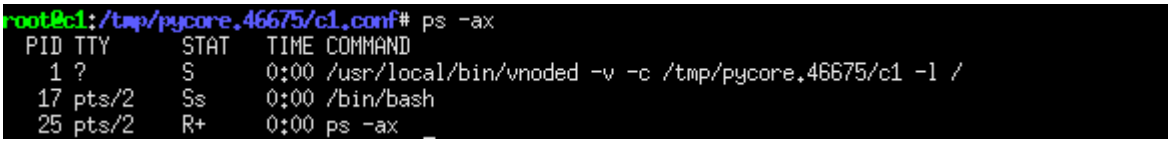


Figura 21: Processos do laptop do departamento C.

Nesta segunda imagem, efetuamos o comando ps -ax num dos laptops do departamento C e concluímos que estamos perante um encaminhamento estático, visto que não nos deparamos com o protocolo OSPF.

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 implicações tem esta medida para os utilizadores da organização MIEI-RC que acedem ao servidor. Justifique.*

Resposta:

Ao remover a rota por defeito no servidor S1 estamos a impossibilitá-lo de se ligar a equipamentos fora do seu departamento e, desta forma, o mesmo deixa de poder enviar pacotes para outras redes. Sendo assim, os clientes conseguem transmitir pacotes, mas não recebem resposta do servidor, porque são descartados todos os pacotes que não têm como destino a própria rede local (10.0.4.0/24).

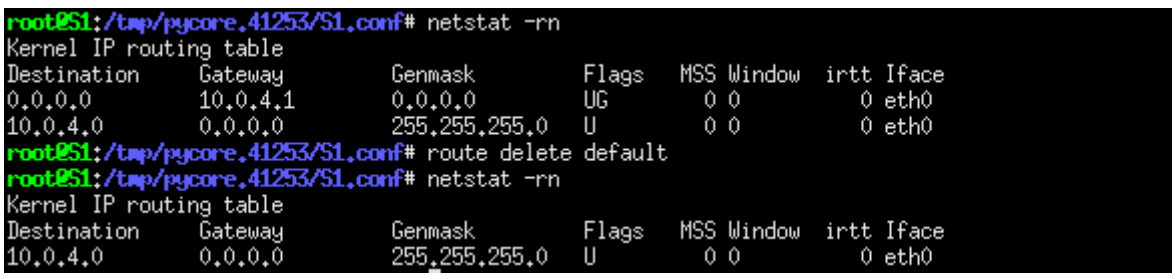


Figura 22: Resultado do comando netstat -rn depois de remover a rota por defeito.

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. 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.*

Resposta:

```
root@S1:/tmp/pycore.41253/S1.conf# route add -net 10.0.5.0 netmask 255.255.255.0
 gw 10.0.4.1
root@S1:/tmp/pycore.41253/S1.conf# route add -net 10.0.6.0 netmask 255.255.255.0
 gw 10.0.4.1
root@S1:/tmp/pycore.41253/S1.conf# route add -net 10.0.7.0 netmask 255.255.255.0
 gw 10.0.4.1
root@S1:/tmp/pycore.41253/S1.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.4.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.5.0        10.0.4.1        255.255.255.0   UG        0 0          0 eth0
10.0.6.0        10.0.4.1        255.255.255.0   UG        0 0          0 eth0
10.0.7.0        10.0.4.1        255.255.255.0   UG        0 0          0 eth0
```

Figura 23: Resultado do comando netstat -rn depois de adicionar rotas entre S1 e os restantes routers.

Para restaurar a conectividade para o servidor S1 adicionamos à tabela os endereços das sub-redes (10.0.5.0, 10.0.6.0, 10.0.7.0), para que seja possível haver partilha de informação entre eles.

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.

Resposta:

Efetuando o comando ping de um PC de cada uma das redes dos outros departamentos após adicionar as rotas para os diversos routers dos departamentos, vemos que o servidor está de novo acessível.

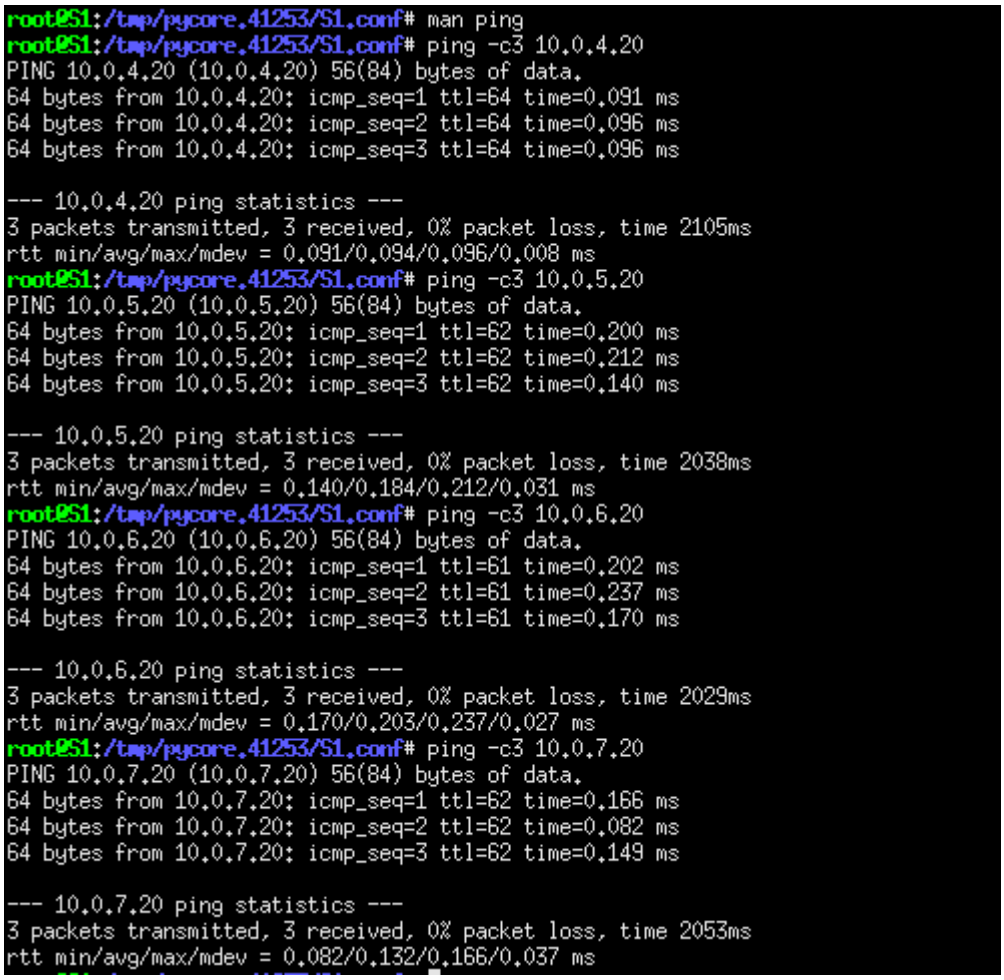


Figura 24: Envio de mensagens do servidor S1 para laptops do departamento A, B, C e D.

Exercício 3:

1) *Considere que dispõe apenas do endereço de rede IP 130.XX.96.0/19, 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. Assuma que todos os endereços de sub-redes são usáveis. Deve justificar as opções usadas.*

Resposta:

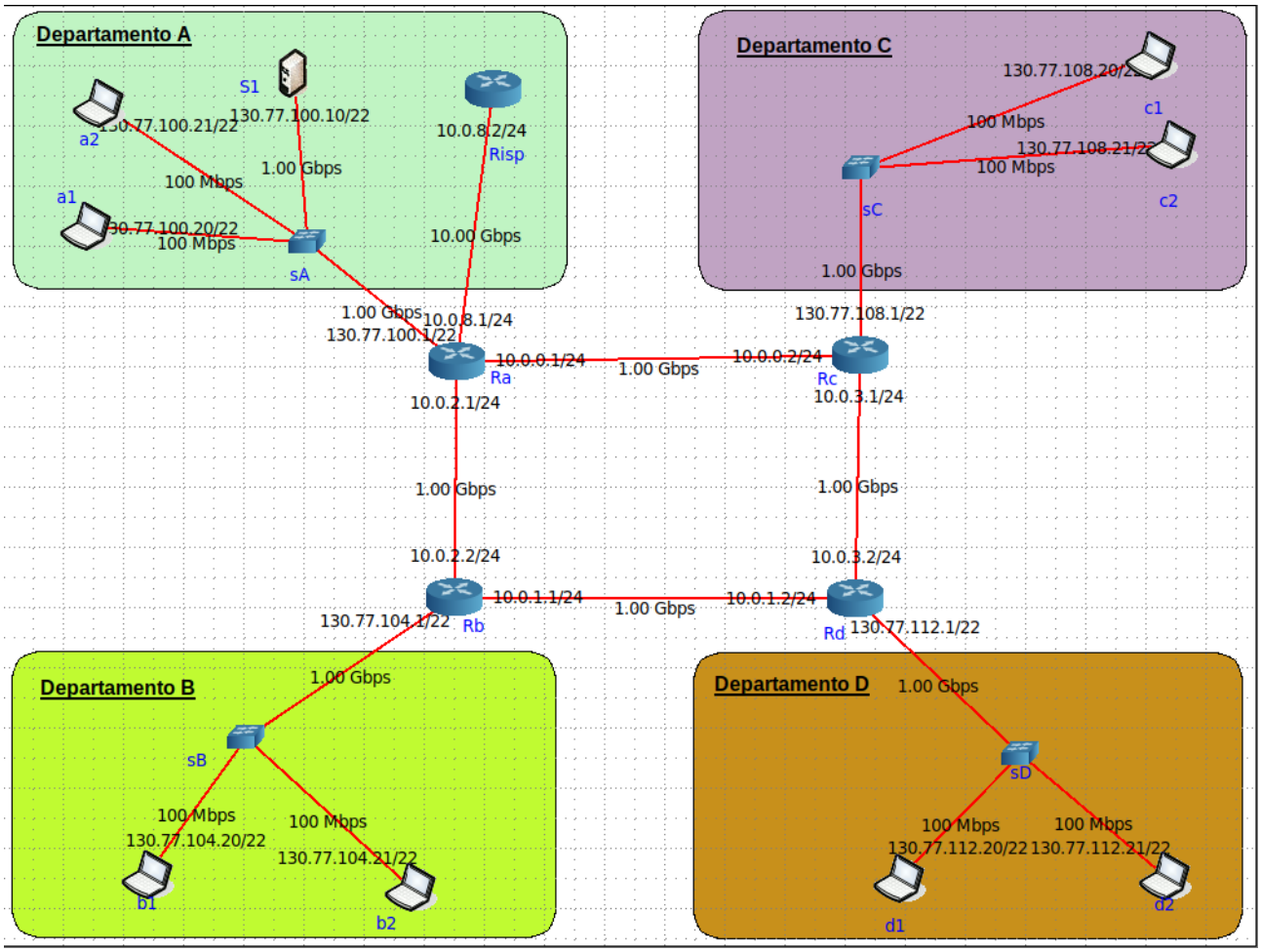


Figura 25: Novo esquema de endereçamento para as redes do departamento.

Como existem 4 departamentos, para representar as 4 redes correspondentes a estes departamentos, são necessários 3 bits, visto que $2^3 - 2 = 6 > 4$ (é subtraído 2, porque decidimos manter o endereço com os bits todos a 1's e 0's reservados, uma vez que estes correspondem ao endereço de broadcast e à comunicação com todos os dispositivos, respetivamente). Ficamos então com um total de 6 sub-redes possíveis. Como reservamos 3 bits para fazer sub-netting, a máscara de rede passa de /19 para /22.

130.77.011|XXX|00.0

000	Reservado	
001	Livre	Departamento A
010	Livre	Departamento B
011	Livre	Departamento C
100	Livre	Departamento D
101	Livre	
110	Livre	
111	Reservado	

Desta forma, associando os primeiros endereços IP's aos departamentos, tem-se que:

Departamento A: 130.77.100/22
Departamento B: 130.77.104/22
Departamento C: 130.77.108/22
Departamento D: 130.77.112/22

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

Resposta:
Como já referido na alínea anterior, reservamos 3 bits para fazer sub-netting, logo a máscara de rede passa de /19 para /22, correspondente a 255.255.252.0 em decimal, sobrando 32-19-3 = 10 bits para podermos alterar. O número de hosts IP é dado por $2^{10} - 2$ (um dos endereços reservado para broadcasting e outro para comunicar com todos os dispositivos), ficando com 1022 hosts IP .

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

Resposta:
Após os testes demonstrados em baixo, verificamos que a conetividade IP entre as várias redes locais na nossa organização MIEI-RC foi mantida. Para isso executamos o comando ping a partir do laptop (**a1**) no *Departamento A*, para a respetiva verificação da conectividade com:

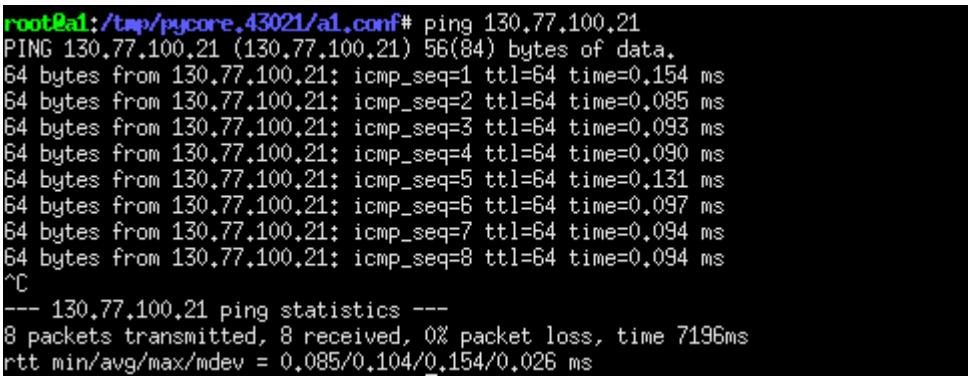


Figura 26 : Conectividade entre o *laptop* (**a1**) e o laptop (**a2**) no mesmo Departamento.

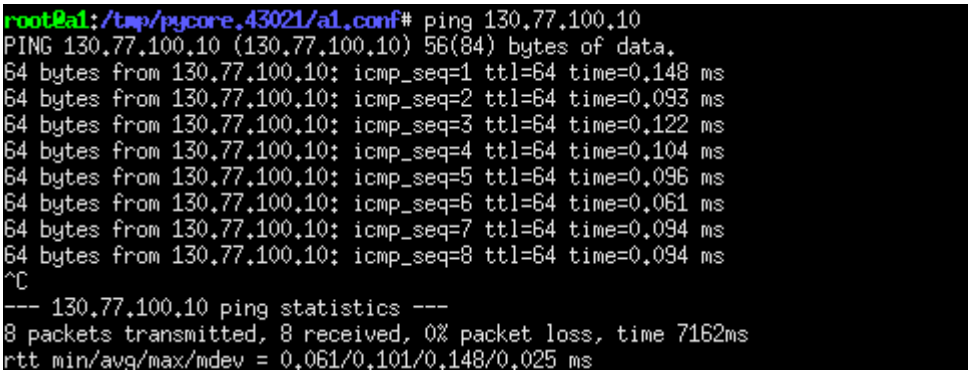


Figura 27 : Conectividade entre o *laptop* (**a1**) e o servidor (S**1**) no mesmo Departamento.

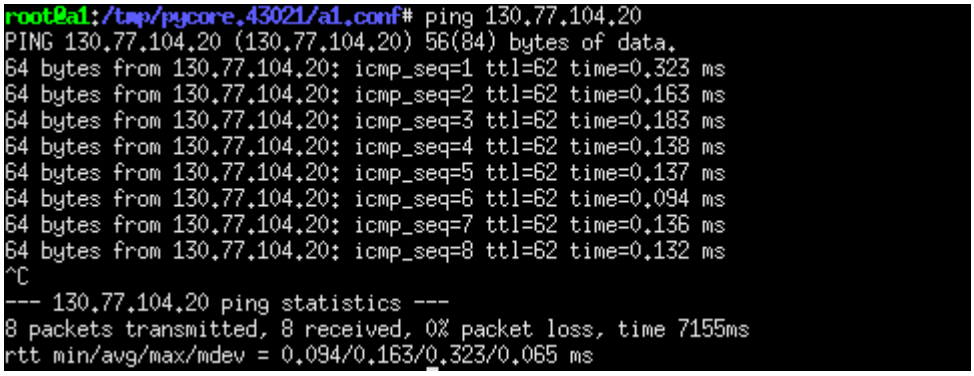


Figura 28: Conectividade entre o *laptop* (**a1**) e o laptop (**b1**) do Departamento **B.**

Figura 29: Conectividade entre o *laptop* (**a1**) e o laptop (**c1**) do Departamento **C.**

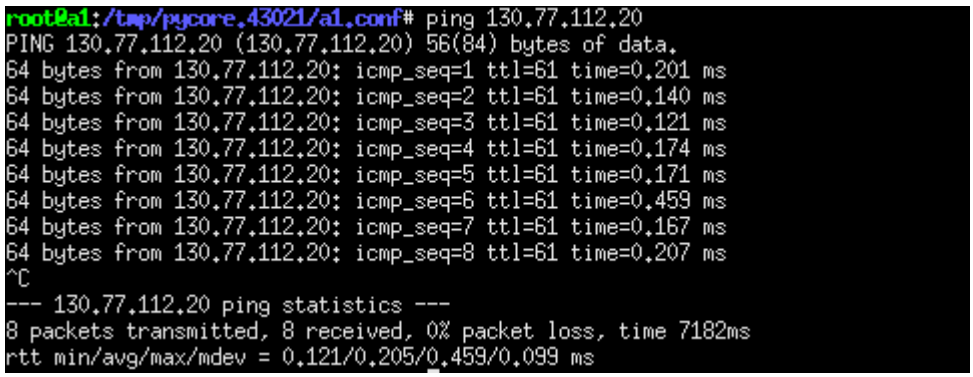


Figura 30: Conectividade entre o *laptop* (**a1**) e o laptop (**d1**) do Departamento **D.**

Realizamos um processo de teste semelhante aos anteriores, com os mesmos destinos, mas a partir do laptop (**d1**) no *Departamento D*, obtendo o mesmo sucesso nas conectividades entre as várias redes. De seguida mostramos o último comando ping executado do laptop (**d1**) para o Servidor (**S1**) do *Departamento A*.

Figura 31: Conectividade entre *laptop* (**d1**) e o servidor (**S1**) do Departamento **A.**

Conclusões

Este foi um estudo faseado do protocolo IPv4 onde foram abordados temas como fragmentação de pacotes , endereçamento e encaminhamento deste protocolo.

Na primeira parte do trabalho, foi feita uma análise ao protocolo de IPv4. De modo a concretizar isso, foi estabelecida uma topologia Core para estudar o comportamento e explorar o tráfego ICMP através da análise de datagramas. Também foi feita a análise de casos mais específicos, onde foi necessária uma fragmentação de pacotes IP devido à sua grande dimensão. Aqui foram analisadas as flags "fragment offset" e "more fragments", a posição de um fragmento relativamente a outros e se havia mais fragmentos para além do atual.

Na segunda parte, o estudo foi focado não na parte de transmissão de dados mas na estrutura do IPv4, mais concretamente no endereçamento, gestão de redes e criação de sub-redes. Após a construção da topologia com os vários departamentos e respetivos equipamentos, analisamos a conectividade entre os equipamentos de cada departamento. Em suma, vimos o funcionamento do encaminhamento entre redes diferentes de 4 departamentos, adquirindo conhecimentos essenciais em torno da divisão de sub-netting. Relativamente ao encaminhamento, concluiu-se que na rede local o uso de endereços privados é benéfico para que não haja uma maior carga de uso de endereços públicos, o que na atualidade é uma grande vantagem tendo em conta a dimensão da rede global.

Com a realização deste trabalho, aplicamos e aprofundamos os conhecimentos adquiridos nas aulas teóricas de Redes de Computadores sobre o Protocolo da Internet (IP) e como este funciona.