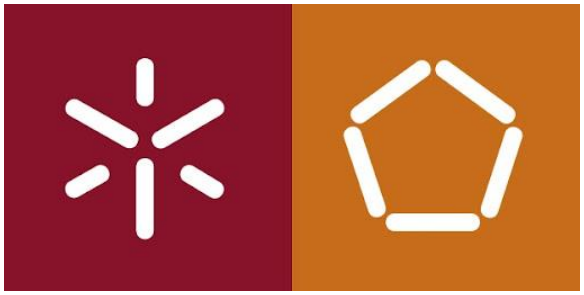


Universidade do Minho

Relatório do Trabalho Prático 2:

Protocolo IPv4 :: Datagramas IP e Fragmentação

Ano letivo 2021/2022

Abril 2022

Licenciatura em Engenharia Informática

Unidade Curricular de Redes de Computadores

Ana Gonçalves a93259

Luís Faria a93209

Jéssica Fernandes a93318

Índice

Índice de Figuras

Introdução

Neste relatório será abordado a resolução do trabalho prático da UC Redes de Computadores, cujo enunciado tinha como objetivo o aprofundamento do conhecimento sobre o protocolo IP, nomeadamente o formato de um pacote, a sua fragmentação, endereçamento e finalmente encaminhamento.

Deste modo, será apresentado neste documento, as resoluções da parte 1 e parte 2 , constituídas pelas perguntas e respostas justificadas, juntamente com uma conclusão do trabalho prático.

Parte 1

Questão 1 (Xubuntu)

a) Active o wireshark ou o tcpdump no host Bela. Numa shell de Bela execute o comando traceroute -I para o endereço IP do Monstro

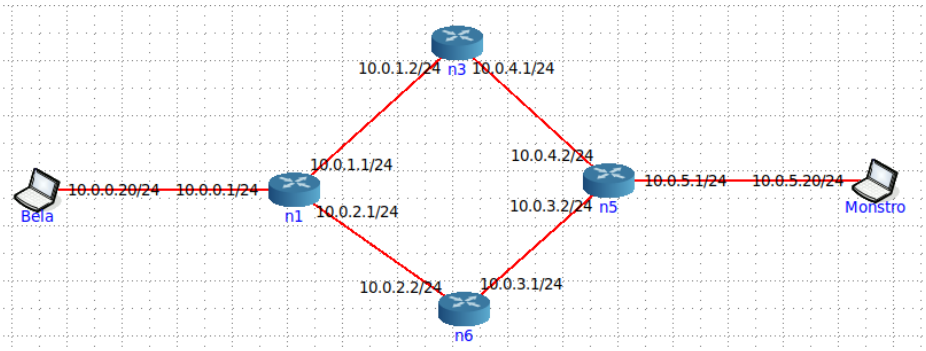


Figura 1: Topologia de Rede Core

```
root@Bela:/tmp/pycore.40631/Bela.conf# traceroute -I 10.0.5.20
traceroute to 10.0.5.20 (10.0.5.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.036 ms  0.007 ms  0.006 ms
 2  10.0.1.2 (10.0.1.2)  0.047 ms  0.008 ms  0.010 ms
 3  10.0.3.2 (10.0.3.2)  0.041 ms  0.012 ms  0.011 ms
 4  10.0.5.20 (10.0.5.20)  0.031 ms  0.013 ms  0.013 ms
root@Bela:/tmp/pycore.40631/Bela.conf#
```

Figura 2: Resultado de Traceroute no host Bela

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

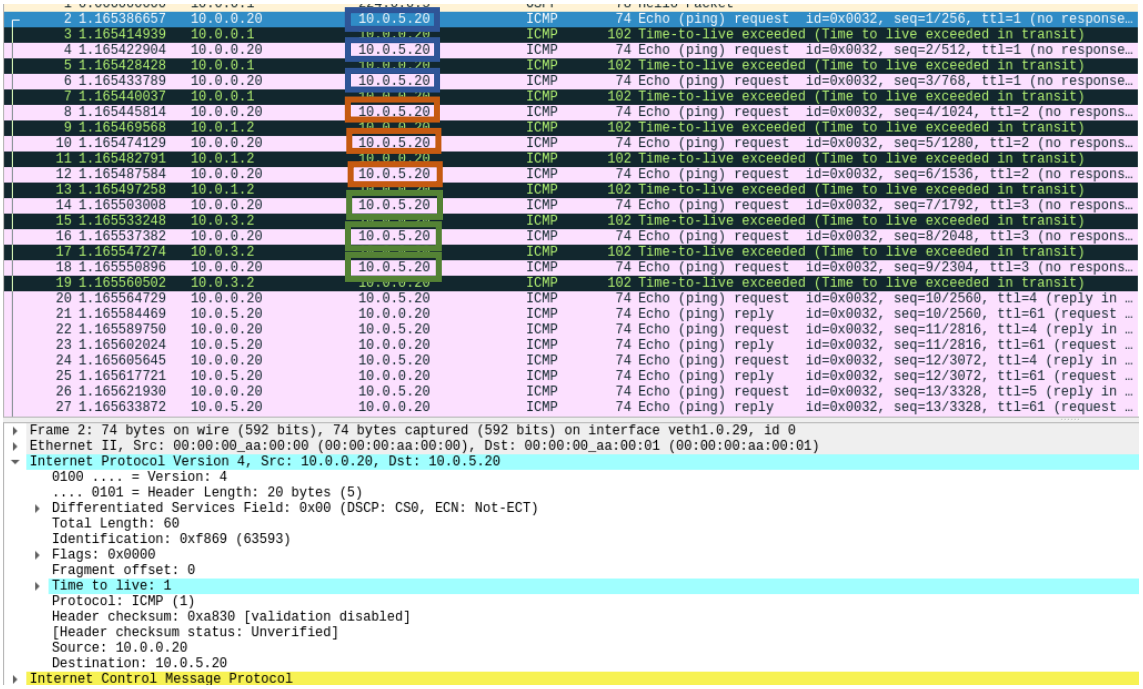


Figura 3: Tráfego capturado após Traceroute

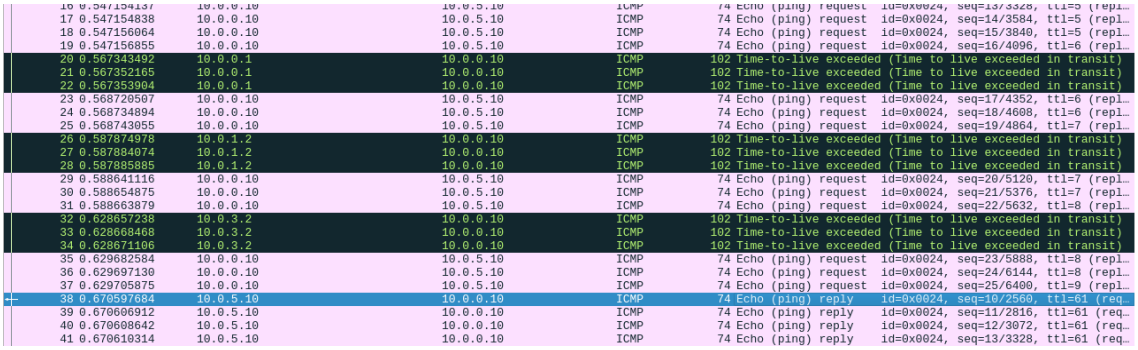
- Correspondem aos 3 pacotes enviados para o host Monstro com TTL de 1
- Correspondem aos 3 pacotes enviados para o host Monstro com TTL de 2
- Correspondem aos 3 pacotes enviados para o host Monstro com TTL de 3

E o ciclo continua até receber o primeiro pacote que confirma que chegou ao host Monstro.

O comportamento esperado do Traceroute, consistia no envio de X pacotes com TTL a crescer linearmente, de modo a chegar ao destino pretendido. E com a figura acima apresentada podemos confirmar que é esse o comportamento que efetivamente acontece. Neste caso específico são enviados 3 pacotes com TTL igual, inicializado a 1.

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

Como verificamos na questão anterior, o host Bela recebeu 3 sequencias de 3 pacotes que não alcançaram o destino. Sendo que cada sequência está associada a um pacote enviado com TTL de 1,2 e 3 correspondentemente, podemos confirmar que o valor mínimo de TTL para alcançar o servidor Monstro é de 4.



17 0.547154838	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=14/3584, ttl=5 (repl…
18 0.547156064	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=15/3840, ttl=5 (repl…
19 0.547156855	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=16/4096, ttl=6 (repl…
20 0.567343492	10.0.0.1	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
21 0.567352165	10.0.0.1	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
22 0.567353904	10.0.0.1	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
23 0.568720507	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=17/4352, ttl=6 (repl…
24 0.568734894	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=18/4608, ttl=6 (repl…
25 0.568743055	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=19/4864, ttl=7 (repl…
26 0.587874978	10.0.1.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
27 0.587884074	10.0.1.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
28 0.587885885	10.0.1.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
29 0.588641116	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=20/5120, ttl=7 (repl…
30 0.588654875	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=21/5376, ttl=7 (repl…
31 0.588663879	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=22/5632, ttl=8 (repl…
32 0.628657238	10.0.3.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
33 0.628668468	10.0.3.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
34 0.628671106	10.0.3.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)	
35 0.629682584	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=23/5888, ttl=8 (repl…
36 0.629697130	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=24/6144, ttl=8 (repl…
37 0.629705875	10.0.0.10	10.0.5.10	ICMP	74 Echo (ping) request	id=0x0024, seq=25/6400, ttl=9 (repl…
38 0.670597684	10.0.5.10	10.0.0.10	ICMP	74 Echo (ping) reply	id=0x0024, seq=10/2560, ttl=61 (req…
39 0.670606912	10.0.5.10	10.0.0.10	ICMP	74 Echo (ping) reply	id=0x0024, seq=11/2816, ttl=61 (req…
40 0.670608642	10.0.5.10	10.0.0.10	ICMP	74 Echo (ping) reply	id=0x0024, seq=12/3072, ttl=61 (req…
41 0.670610314	10.0.5.10	10.0.0.10	ICMP	74 Echo (ping) reply	id=0x0024, seq=13/3328, ttl=61 (req…

Figura 4: Pacote selecionado como o primeiro pacote enviado pelo host destino

Nesta figura, conseguimos identificar que o pacote selecionado foi enviado com sucesso do host Monstro, em resposta ao pacote com TTL de 4.

d) Calcule o valor médio do tempo de ida-e-volta (RTT - Round-Trip Time) obtido no acesso ao servidor. Para melhorar a média, poderá alterar o número pacotes de prova com a opção -q.

```
root@Bela:/tmp/pycore.40631/Bela.conf# traceroute -q 5 10.0.5.20
traceroute to 10.0.5.20 (10.0.5.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.037 ms  0.006 ms  0.006 ms  0.005 ms  0.005 ms
 2  10.0.1.2 (10.0.1.2)  0.019 ms  0.010 ms  0.009 ms  0.009 ms  0.009 ms
 3  10.0.3.2 (10.0.3.2)  0.030 ms  0.013 ms  0.015 ms  0.011 ms  0.012 ms
 4  10.0.5.20 (10.0.5.20)  0.045 ms  0.025 ms  0.015 ms  0.027 ms  0.014 ms
```

Figura 5: Comando traceroute com maior número de repetições, no host Bela para o host Monstro

Tratando-se a última linha dos vários RTT calculados, a média é igualada a 0.0198 ms.

e) O valor médio do atraso num sentido (One-Way Delay) poderia ser calculado com precisão dividindo o RTT por dois? O que torna difícil o cálculo desta métrica?

Não, o facto de que o caminho percorrido na transmissão Bela->Mostro poder ser diferente do caminho percorrido na transmissão Monstro->Bela implica que a divisão por 2 do RTT não vai ser uma versão fiável do One-Way Delay. Isto porque, se o caminho de ida for muito mais demorado, ao contrário do caminho de volta, o One-Way Delay num sentido vai ser muito mais elevado ao real, e o contrário acontece para o sentido oposto.

Questão 2 (Máquina Nativa: Ubuntu)

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

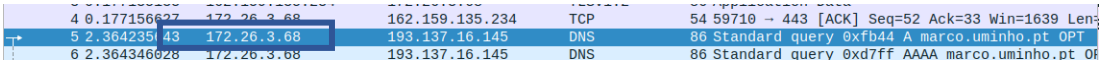


Figura 6: Pacote de Standard Query feita pela máquina nativa

Como podemos verificar o IP da interface do computador pessoal é 172.26.3.68

b) Qual é o valor do campo protocolo? O que permite identificar?

Na seguinte figura verificamos que o valor do campo do protocolo é ICMP, e é identificado pelo valor entre parentesis, que no caso de ICMP é 1.

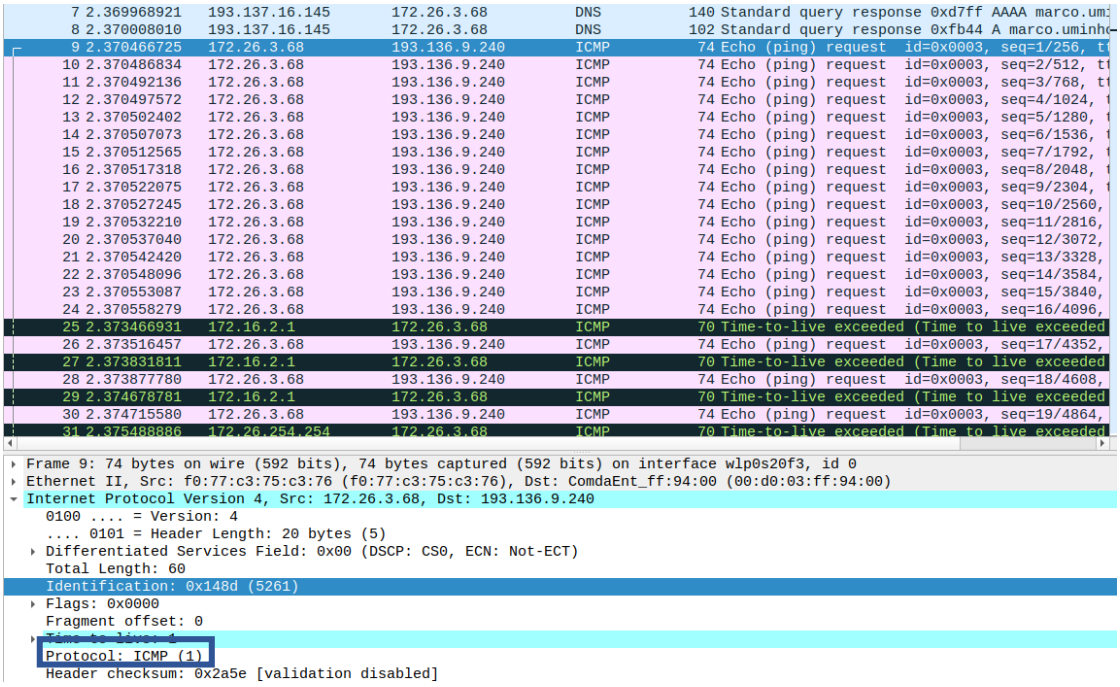


Figura 7: Primeiro Pacote enviado pela máquina nativa

c) Quantos bytes tem o cabeçalho IPv4? Quantos bytes tem o campo de dados (payload) do datagrama? Como se calcula o tamanho do payload?

Para poder verificar estas informações é necessário selecionar o pacote ICMP correto, e recolher o tamanho do pacote IP 60 bytes. Sendo o tamanho do overhead de 20 bytes, fica confirmado que o tamanho do campo de dados é de:

$$Tamanho\ IP\ -\ Overhead\ =\ 60 - 20 = 40\ bytes$$

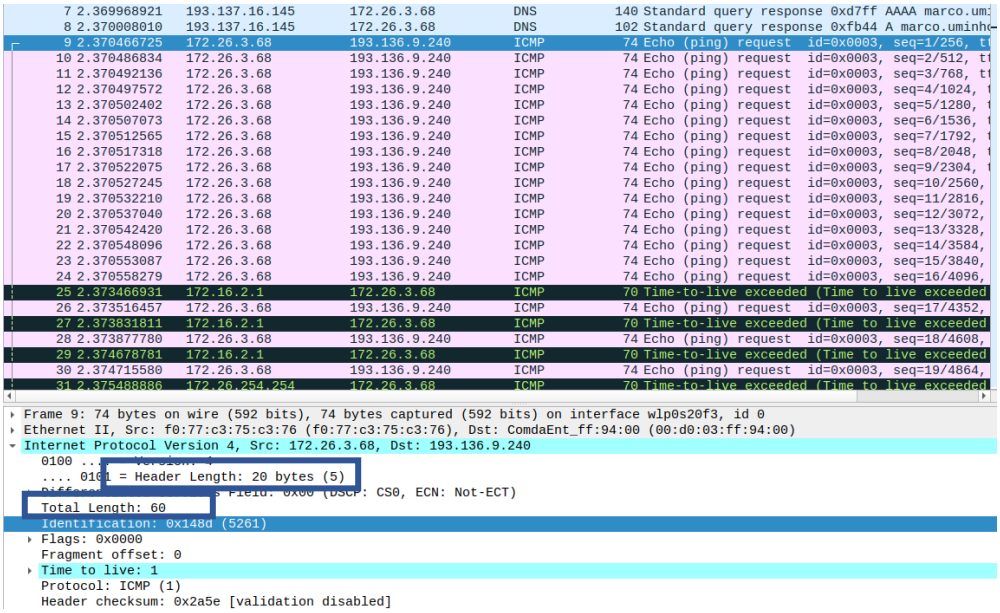


Figura 8: Primeiro Pacote enviado pela máquina nativa

d) O datagrama IP foi fragmentado? Justifique

Não, podemos confirmar através da leitura do wireshark que não existe nenhuma notificação de fragmentação de pacote: através da flag a indicar mais fragmentos nem de offsets diferentes de 0.

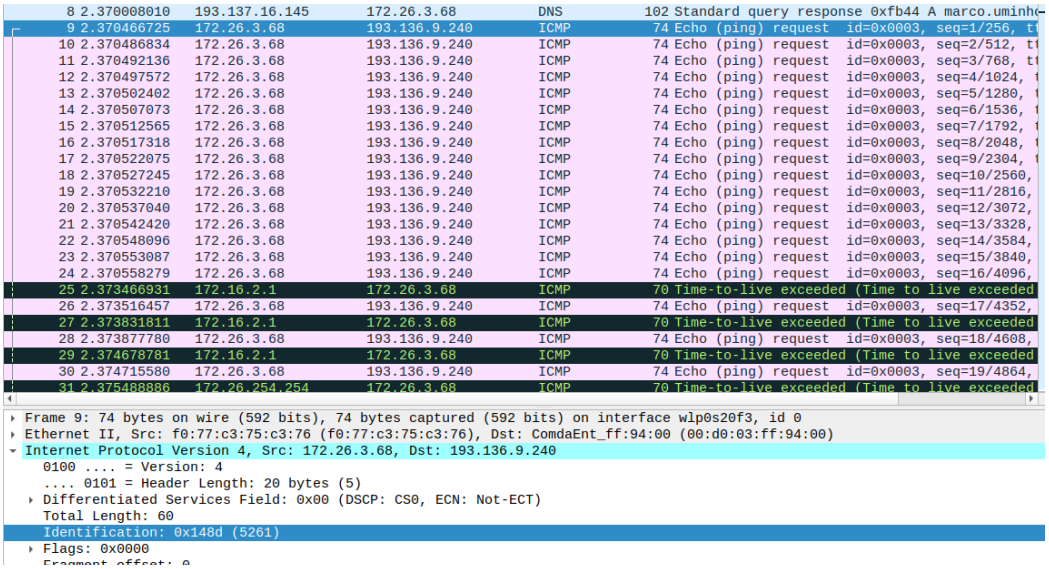


Figura 9: Primeiro pacote enviado pela máquina nativa

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.

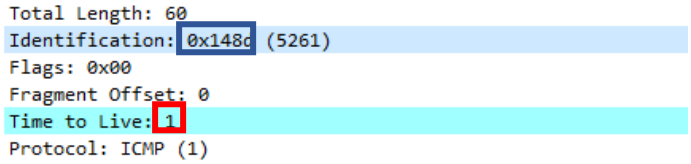


Figura 10: Primeiro pacote enviado

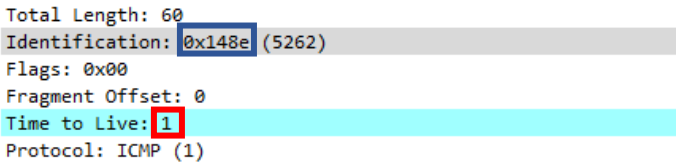


Figura 11: Segundo pacote enviado

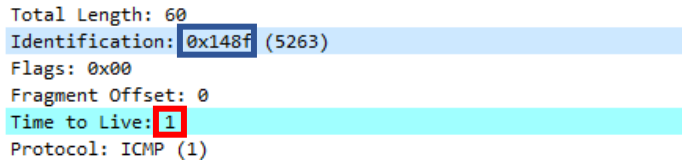


Figura 12: Terceiro pacote enviado

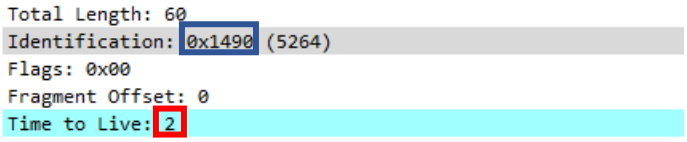


Figura 13: Quarto pacote enviado

As figuras anteriores apresentam os campos de IP do primeiro, segundo, terceiro e quarto pacote, respetivamente. Como são enviados 3 pacotes por 1 TTL, dentro desses grupos de 3 pacotes (Figura 11, 12 e 13 correspondem ao primeiro grupo enviado) , varia apenas o identificador. De grupo em grupo, o TTL varia adicionalmente.

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

Sim, verificamos que o identificador aumenta linearmente a cada pacote, e o TTL aumenta linearmente de 3 em 3 pacotes, iniciando o ciclo em TTL igual a 1.

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

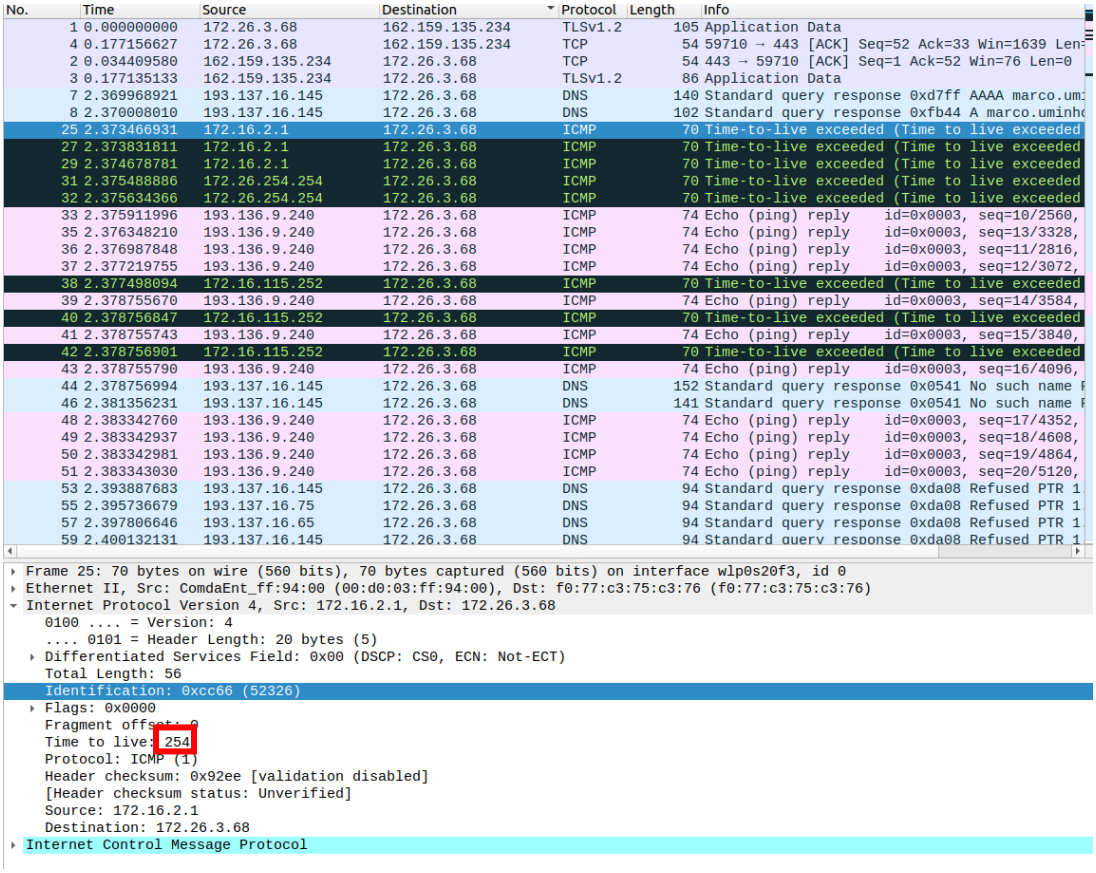


Figura 14: Primeiro pacote recebido por TTL excedido

Nos vários pacotes ICMP com Time-to-live excedido, foi verificado que os TTL recebidos são todos elevados e maioritariamente constantes (254 / 255). Isto deve-se ao facto de que os routers e o host não terem informação da quantidade de saltos que os pacotes vão ter de percorrer para chegar ao destino. E por isso, é necessário um valor alto de TTL para confirmar a sua chegada ao destino correto.

Questão 3 (Máquina Nativa: Ubuntu com Tamanho de pacote = 4071)

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

A primeira mensagem ICMP enviada foi a seleciona na seguinte figura:

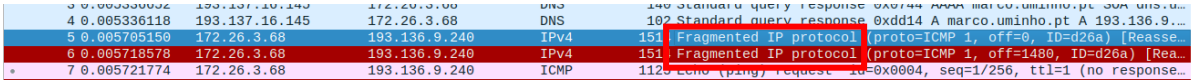


Figura 15: Primeiro fragmento enviado

Esta fragmentação ocorreu por uma única razão: o tamanho do pacote que foi forçado. O facto de o traceroute enviar um pacote de tamanho tão grande, obrigou a uma fragmentação deste, mais especificamente, em 3 pacotes.

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?

Selecionando o primeiro fragmento do enviado, verificamos que podemos confirmar que se trata de um pacote fragmento graças à flag levantada que anuncia a existência de mais pacotes fragmentos por receber.

Identificamos que se trata do primeiro fragmento pois o seu offset é igual a 0, o que implica que na reconstrução do pacote original, ele vai ser inserido no início do pacote construído.

E finalmente, como podemos verificar pela figura, o tamanho do primeiro datagrama fragmentado é de 1500 bytes.

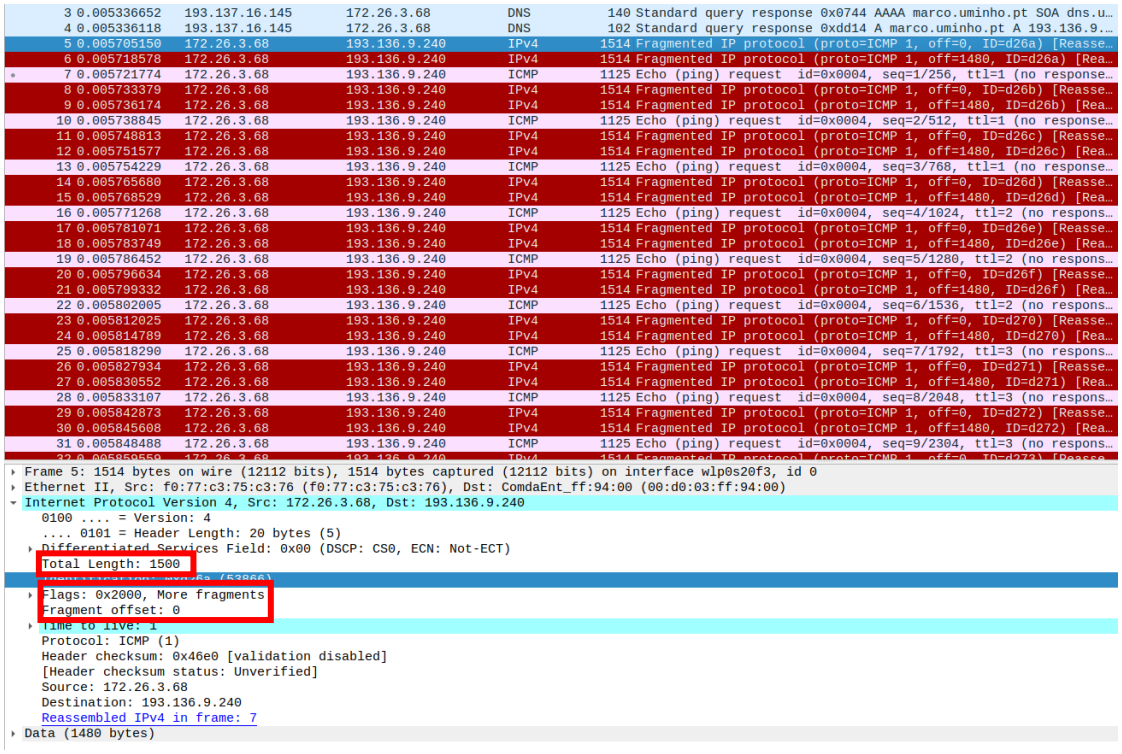


Figura 16: Primeiro fragmento enviado

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?

Na seguinte figura, verificamos que não se trata do primeiro fragmento visto que offset já não se encontra igual a 0. No entanto, ainda existem mais fragmentos por enviar pois a flag “more fragments” ainda se encontra levantada.

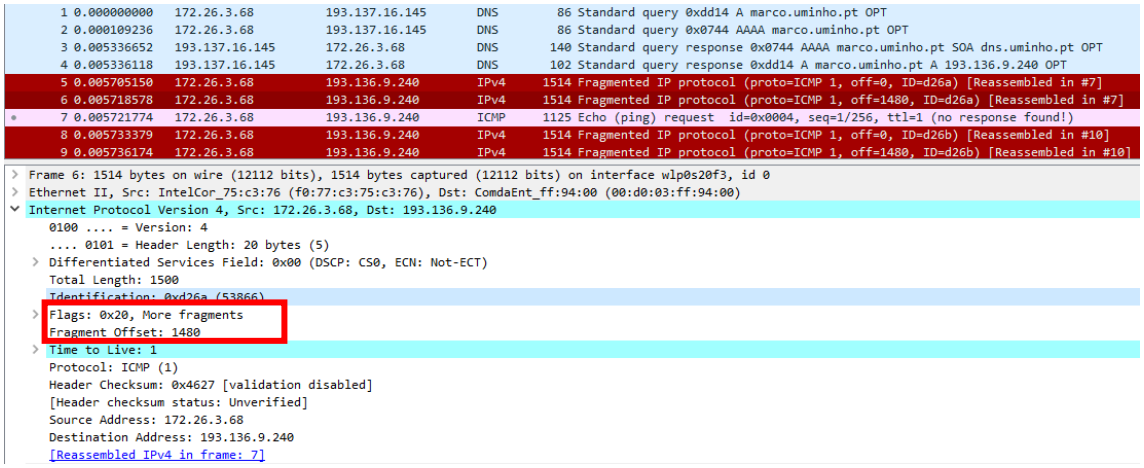


Figura 17: Segundo fragmento enviado

d) Quantos fragmentos foram criados a partir do datagrama original?

Como podemos verificar, foram criados 3 fragmentos apartir de um datagrama de tamanho 4071 bytes.

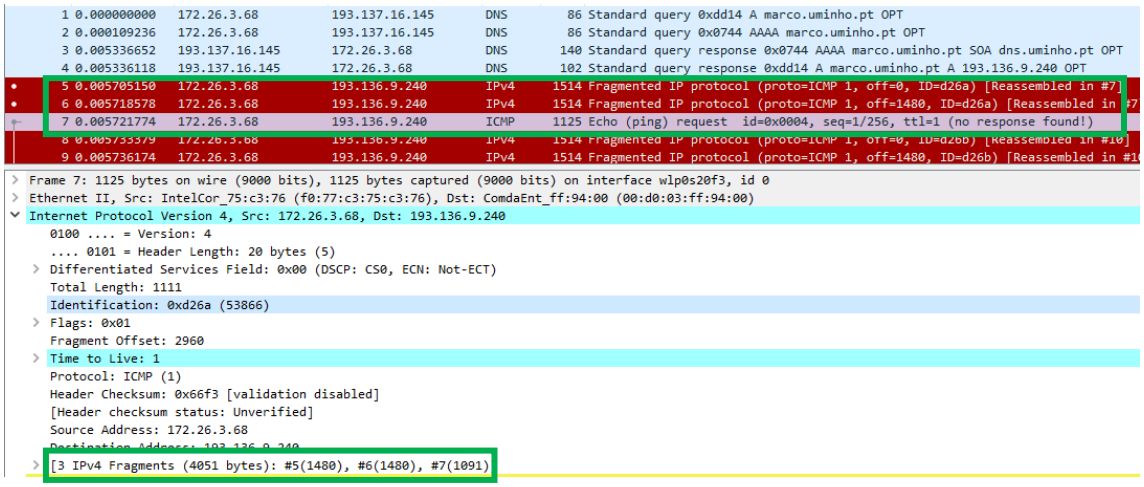


Figura 18: Terceiro e último fragmento do primeiro datagrama original

f) 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.

Considerando um pacote original fragmentado, cada um dos fragmentos vai ter o mesmo identificadores e um offset diferente. Para construir o datagrama original, será necessário reunir todos os pacotes recebidos com o mesmo identificador até a flag de "more fragments" já não estiver levantada, e reuni-los de acordo com os seus offsets, começando pelo offset igual a 0.

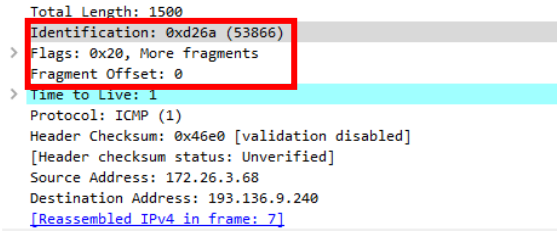


Figura 19: Primeiro fragmento

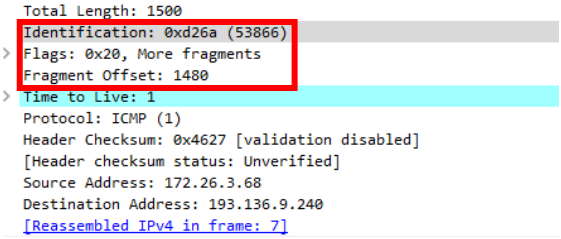


Figura 20: Segundo Fragmento

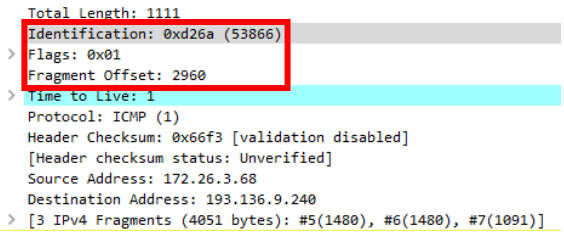


Figura 21: Terceiro e último fragmento

e) Verifique o processo de fragmentação através de um processo de cálculo.

Sendo o tamanho original do datagrama de 4071, vamos verificar como ocorreu a fragmentação capturada pelo wireshark.

Se não ocorrer fragmentação, seria enviado um pacote com tamanho de 4071 bytes, em que 20 deles corresponderiam ao overhead e 4051 de conteúdo. No entanto, com a fragmentação, isso não aconteceu.

O que efetivamente aconteceu foi que um dos pacotes foi preenchido com o máximo de conteúdo possível, 1480 bytes, juntamente com 20 bytes, correspondendo a um tamanho total de 1500 bytes, como se pode verificar na figura 20 e 21. Esta situação ocorre 2 vezes, pois o tamanho do pacote obriga a recorrer a 3 pacotes fragmentados. No entanto, no terceiro fragmento, a quantidade de dados restante vai ser igual a 4051 – 1480 – 1480 = 1091. Sendo esta a quantidade de dados, e 20 bytes de overhead, vai resultar num pacote de tamanho de 1111, confirmado pela figura 22.

h) Escreva uma expressão lógica que permita detetar o último fragmento correspondente ao datagrama original

Através de pseudocódigo conseguimos fazer esta identificação:

```
While (flag != “more fragments” && wantedID == currentID && offset != 0)
       Fragment = newFragment
LastFragment = Fragment
```

Parte 2

Questão 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.

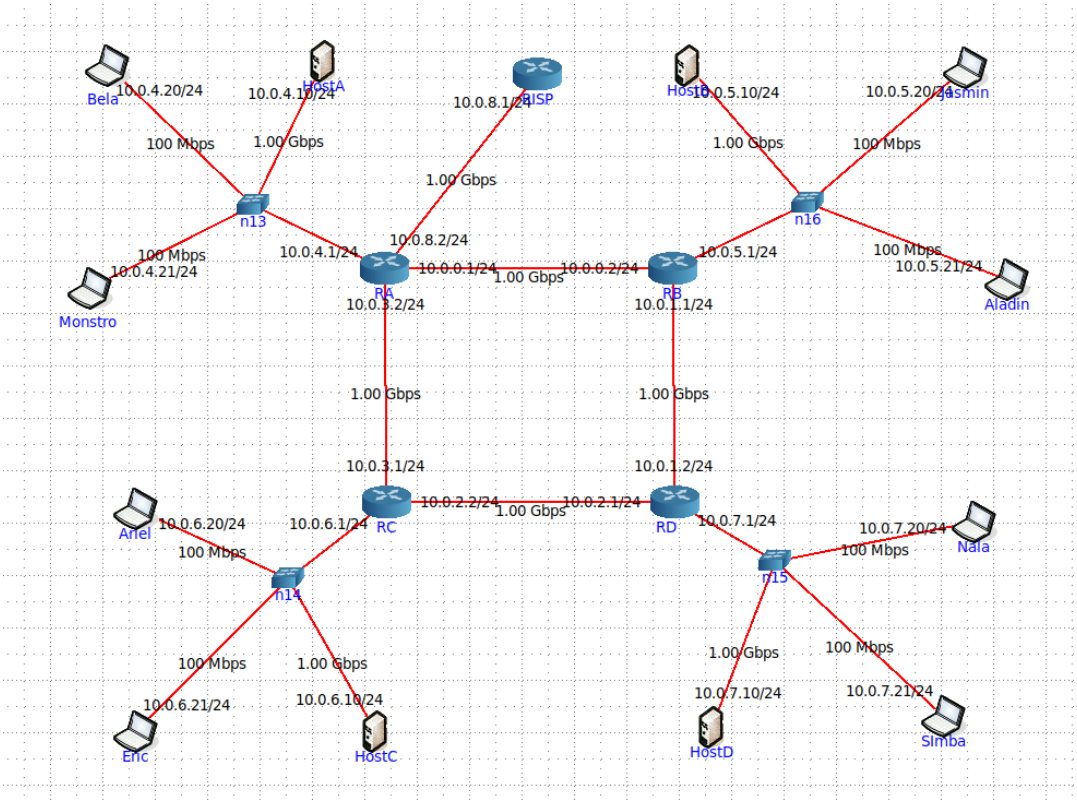


Figura 22: Topologia da Rede

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

Uma das faixas de endereços que está reservada é para os endereços privados, que varia entre os 10.0.0.0/24 e 10.255.255.255/24. E assim confirmamos pelos endereços da figura 23 que estão todos dentro dessa gama, e por isso são privados.

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

Os ethernet switches são instrumentos que operam apenas no nível 2 da pilha protocolar. Isto significando que eles não operam a nível da rede e por isso não precisam de adotar um endereço IP.

d) Usando o comando ping certifique-se que existe conectividade IP interna a cada departamento (e.g. entre um laptop e o servidor respetivo).

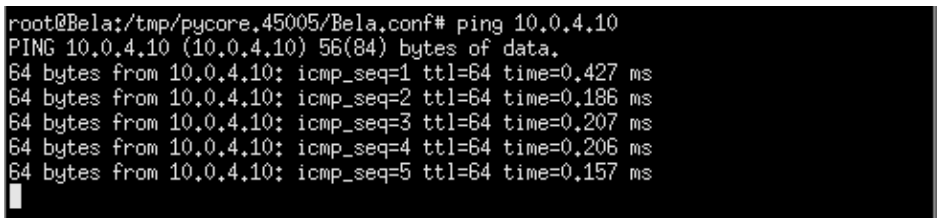


Figura 23: Comando Ping de Bela para SA

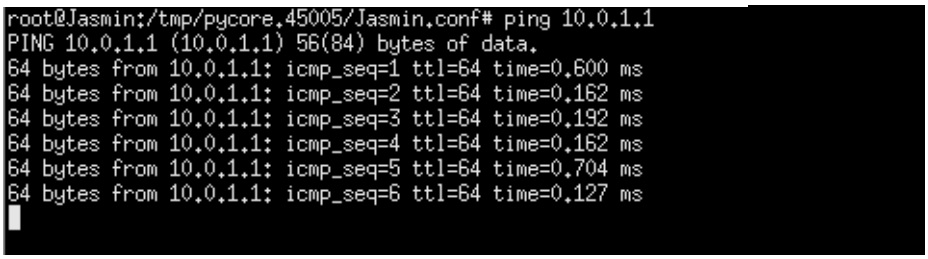


Figura 24: Comando Ping de Jasmin para SB

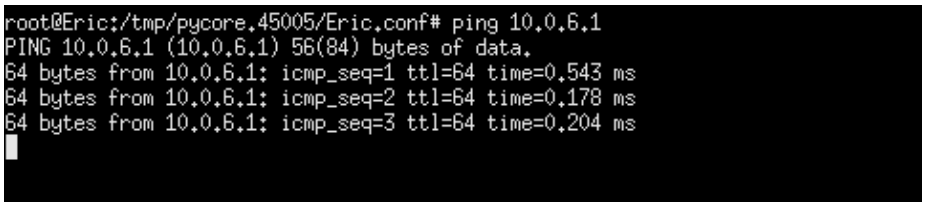


Figura 25: Comando Ping de Eric para SC

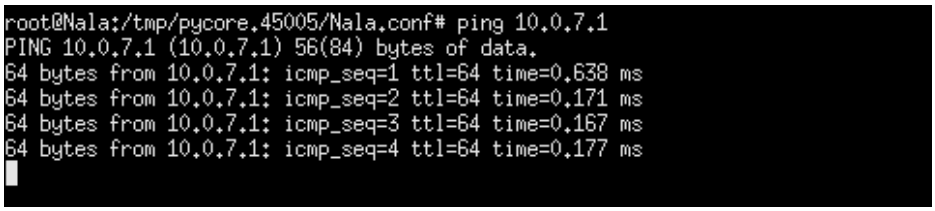


Figura 26: Comando ping de Nala para SD

Podemos verificar pelas figuras acima que em todos os departamentos é possível estabelecer conectividade interna.

e) Execute o número mínimo de comandos ping que lhe permite verificar a existência de conetividade IP entre departamentos.

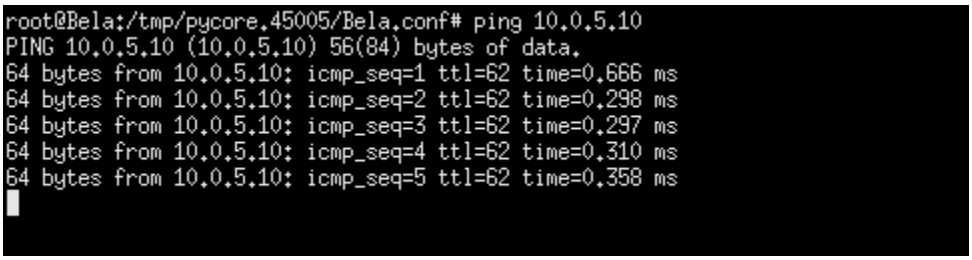


Figura 27: Comando ping de Bela para SB

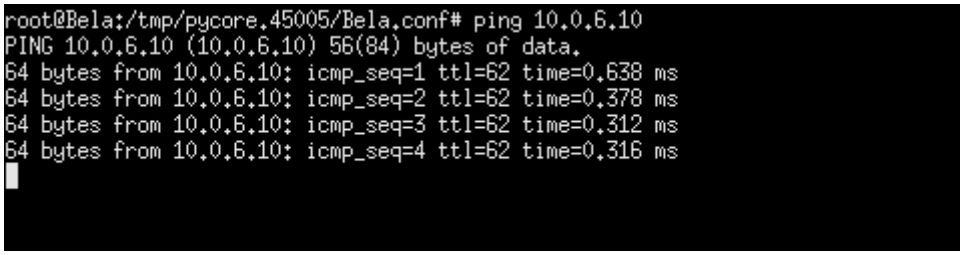


Figura 28: Comando Ping de Bela para SC

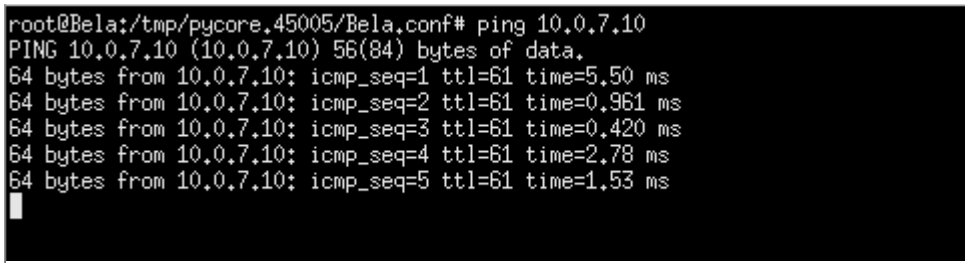


Figura 29: Comando Ping de Bela para SD

Com estes 3 comandos confirmamos o estabelecimento da ligação entre A com B,C e D. E o facto de se tratar de um estabelecimento de ligação podemos afirmar que as ligações inversas também estão operacionais. A seguir temos as ligações que faltavam averiguar relativamente ao departamento C

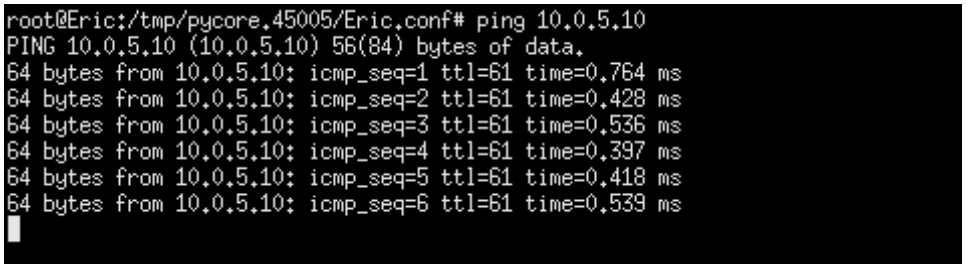


Figura 30: Comando Ping de Eric para SB

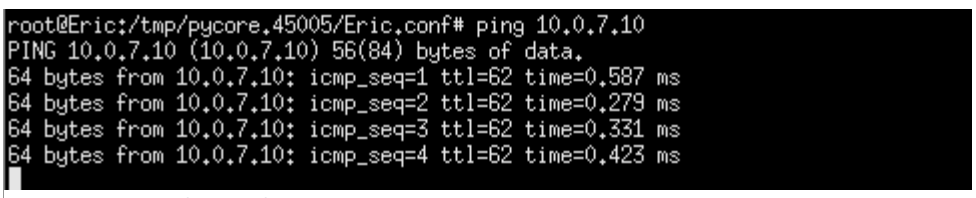


Figura 31: Comando Ping de Eric para SD

E finalmente, temos a confirmação da ligação restante, D e B:

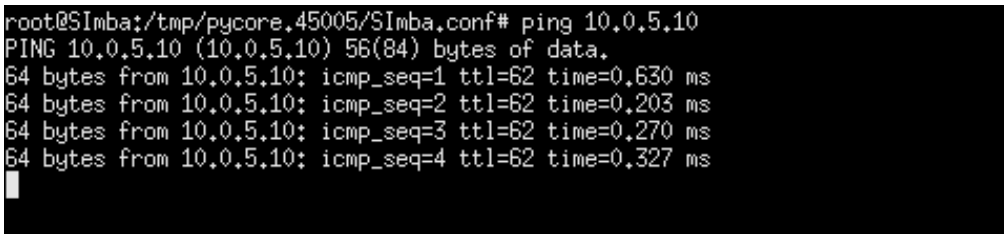


Figura 32: Comando Ping de Simba para SB

Concluindo, para realizar a testagem de ligação entre departamentos são necessários no mínimo 6 comandos pings nesta topologia.

f) Verifique se existe conectividade IP do portátil Bela para o router de acesso RISP

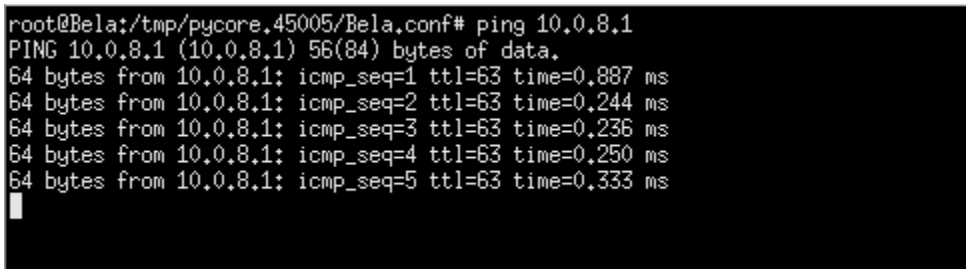


Figura 33: Comando Ping de Bela para RISP

Confirmamos com esta figura que efetivamente existe uma conectividade entre Bela e o Router ISP.

Questão 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 respetivo (man netstat).

```
root@Bela:/tmp/pycore.45005/Bela.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.4.1        0.0.0.0         UG        0 0          0 eth0
10.0.4.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
root@Bela:/tmp/pycore.45005/Bela.conf#
```

Figura 34: Tabela de Encaminhamento de Bela

Através do comando netstat -rn conseguimos obter a tabela de encaminhamento de Bela, e com isto podemos passar à sua análise sucinta. A tabela tem apenas 2 entradas, a segunda consiste na situação de quando um pacote com destino à sub-rede 10.0.4.0/24 alcançar Bela, ele próprio vai redirecionar esses pacotes a toda a sua sub-rede, e por isso é que o seu próximo salto vai ser ele mesmo. Logo o 0.0.0.0 da segunda entrada vai corresponder a 10.0.4.20. A primeira entrada corresponde a todos os outros destinos que não sejam a sua sub-rede e vai encaminhá-los para o seu router, 10.0.4.1.

```
root@RA:/tmp/pycore.45005/RA.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        10.0.0.2        255.255.255.0   UG        0 0          0 eth0
10.0.2.0        10.0.3.1        255.255.255.0   UG        0 0          0 eth1
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.4.0        0.0.0.0         255.255.255.0   U         0 0          0 eth2
10.0.5.0        10.0.0.2        255.255.255.0   UG        0 0          0 eth0
10.0.6.0        10.0.3.1        255.255.255.0   UG        0 0          0 eth1
10.0.7.0        10.0.0.2        255.255.255.0   UG        0 0          0 eth0
10.0.8.0        0.0.0.0         255.255.255.0   U         0 0          0 eth3
root@RA:/tmp/pycore.45005/RA.conf#
```

Figura 35: Tabela de Encaminhamento de RA

Relativamente à tabela de encaminhamento do router A, esta é constituída por 9 entradas. Para facilitar a interpretação foi criada a seguinte tabela que permite melhor visualizar e interpretar a tabela acima apresentada.

Destino	Pacote já se encontra na rede destino?	Próximo Salto	Observações
10.0.0.0	Sim	10.0.0.1	O pacote já se encontra na ligação ponto a ponto destino, por isso, ele próprio vai encaminhar o pacote.
10.0.1.0	Não	10.0.0.2	O pacote tem de chegar a ligação ponto a ponto entre B e D, e sendo B o mais próximo, o pacote vai ser encaminhado para a interface do router de B.
10.0.2.0	Não	10.0.3.1	O pacote tem de chegar a ligação ponto a ponto entre C e D, e sendo C o mais próximo, o pacote vai ser encaminhado para a interface do router de C.
10.0.3.0	Sim	10.0.0.1	O pacote já se encontra na ligação ponto a ponto destino, por isso, ele próprio vai encaminhar o pacote.
10.0.4.0	Sim	10.0.0.1	O pacote já se encontra na ligação ponto a ponto destino, por isso, ele próprio vai encaminhar o pacote.
10.0.5.0	Não	10.0.0.2	O pacote tem de chegar à sub-rede do departamento B, então o pacote vai ser encaminhado para a interface do router B.
10.0.6.0	Não	10.0.3.1	O pacote tem de chegar à sub-rede do departamento C, então o pacote vai ser encaminhado para a interface do router C.
10.0.7.0	Não	10.0.0.2	O pacote tem de chegar à sub-rede do departamento D, então o pacote vai ser encaminhado para a interface do router B, para ser encaminhado por seguida para a interface do router D.
10.0.8.0	Sim	10.0.0.1	O pacote já se encontra na ligação ponto a ponto destino, por isso, ele próprio vai encaminhar o pacote.

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 ou equivalente).

Como podemos verificar pela seguinte figura, no RouterISP, confirmamos que existe um processo a decorrer, ospfd (open shortest path first daemon). Correspondendo a um processo que procura constantemente o melhor caminho para fazer o encaminhamento, por isso, trata-se de um encaminhamento dinâmico.

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 SA. Use o comando route delete para o efeito. Que implicações tem esta medida para os utilizadores da LEI-RC que acedem ao servidor. Justifique.

```
root@HostA:/tmp/pycore.43025/HostA.conf# route delete default
root@HostA:/tmp/pycore.43025/HostA.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
root@HostA:/tmp/pycore.43025/HostA.conf#
```

Figura 36: Eliminação da rota default

Com a eliminação da rota por defeito, torna-se impossível alguma comunicação feita para fora da rede 10.0.4.0/24 que tenha sido direcionada para o servidor A. Isto acontece, pois, quando algum pacote chega ao servidor A, este não vai saber qual será o seu redireccionamento quando não for da sua própria rede (pois a entrada correspondente à sua rede ainda existe, e esses pacotes, o servidor consegue entregá-los ele próprio).

d) Não volte a repor a rota por defeito. Adicione todas as rotas estáticas necessárias para restaurar a conectividade para o servidor SA, por forma a contornar a restrição imposta na alínea c). Utilize para o efeito o comando route add e registe os comandos que usou.

```
root@HostA:/tmp/pycore.43025/HostA.conf# route delete default
root@HostA:/tmp/pycore.43025/HostA.conf# route add -net 10.0.5.0 gw 10.0.4.1 netmask 255.255.255.0
root@HostA:/tmp/pycore.43025/HostA.conf# route add -net 10.0.6.0 gw 10.0.4.1 netmask 255.255.255.0
root@HostA:/tmp/pycore.43025/HostA.conf# route add -net 10.0.7.0 gw 10.0.4.1 netmask 255.255.255.0
root@HostA:/tmp/pycore.43025/HostA.conf#
```

Figura 37: Comandos utilizados para repor rotas necessárias para desfazer efeito de c)

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.

```
root@HostA:/tmp/pycore.43025/HostA.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
root@HostA:/tmp/pycore.43025/HostA.conf#
```

Figura 38: Nova tabela de encaminhamento do servidor A

Com esta nova tabela de endereçamento, verificamos que está novamente funcional através das seguintes figuras a provar que existe conexão de novo para cada um dos departamentos.

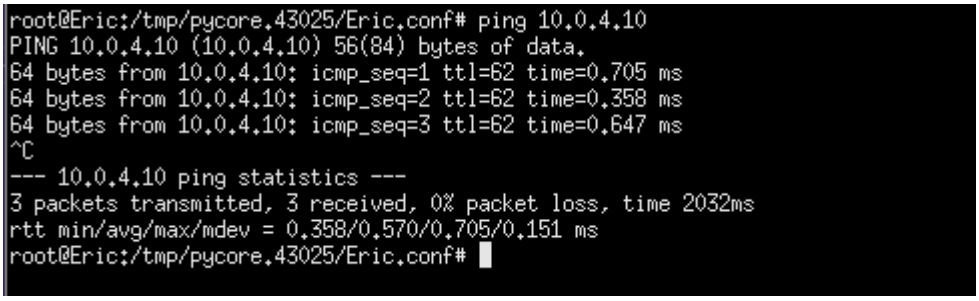


Figura 39: Comando ping do Dep. C para o servidor A

Figura 40: Comando ping do Dep.D para o servidor A

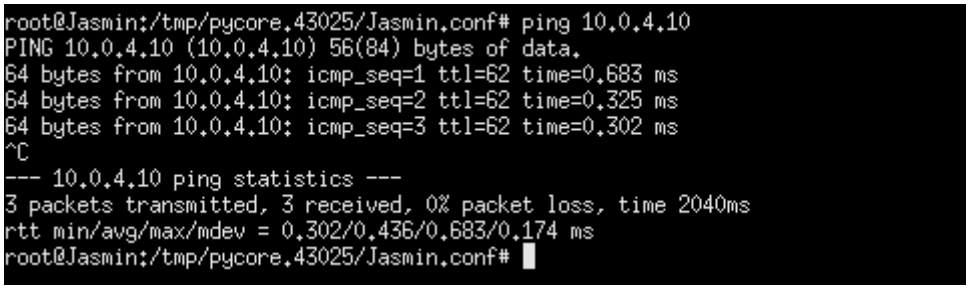


Figura 41: Comando ping do Dep. B para o servidor A

Questão 3

a) Considere que dispõe apenas do endereço de rede IP 192.168.XXX.128/25, em que XXX é o decimal correspondendo ao seu número de grupo (PLXX). Defina um novo esquema de endereçamento para as redes dos departamentos (mantendo as redes de acesso externo e backbone inalteradas), sabendo que o número de departamentos pode vir a aumentar no curto prazo. Atribua endereços às interfaces dos vários sistemas envolvidos. Assuma que todos os endereços de sub_redes são usáveis. Justifique as opções tomadas no planeamento.

Considerando que o endereço IP disponível para realizar a sub-netting em todos os departamentos tem como mascara de 25, sobram 7 bits. Optamos por utilizar 4 bits para sub-redes e 3 bits para hosts.

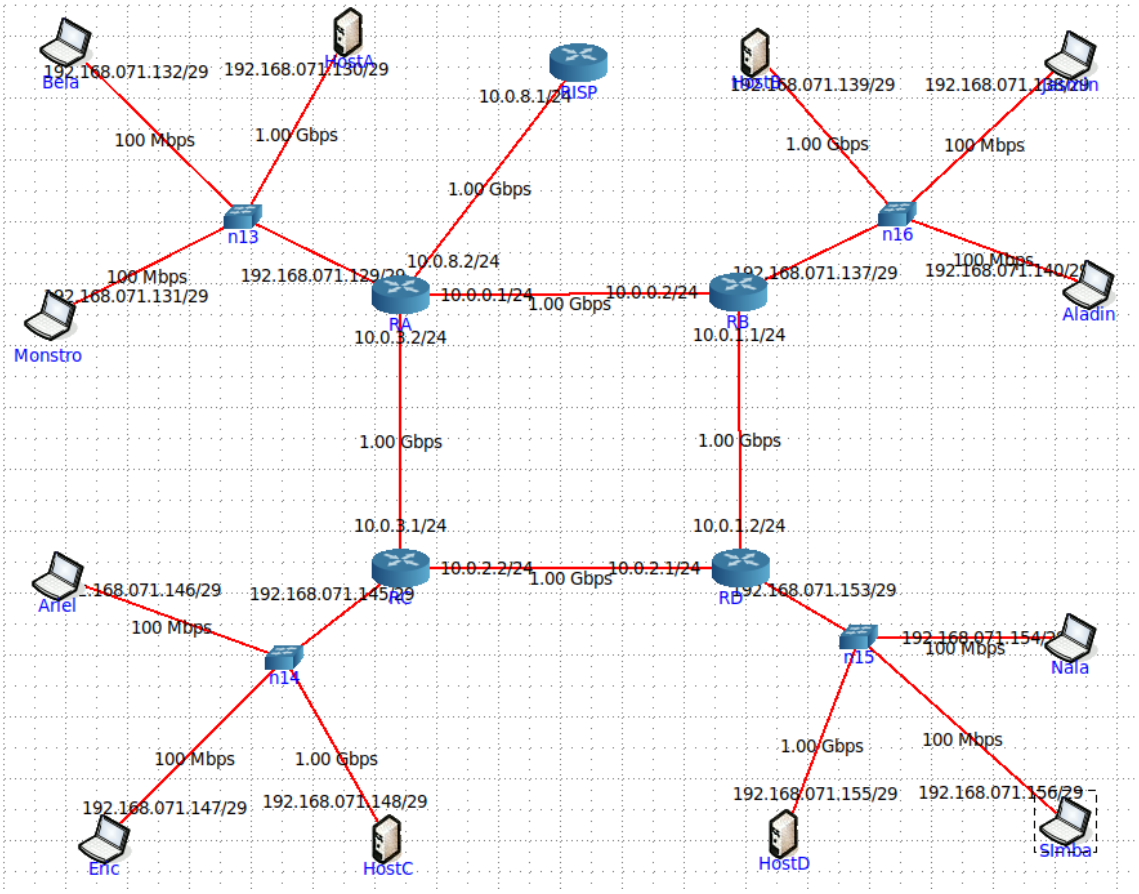


Figura 42: Topologia com sub-netting

b) Qual a máscara de rede que usou (em formato decimal)? Quantos hosts IP pode interligar em cada departamento? Quantos prefixos de sub-rede ficam disponíveis para uso futuro? Justifique

Optamos por reservar 3 bits para o host (de cada departamento) possibilitando a existência de $2^{3\ (nr\ de\ bits)} - 2\ (reservados) = 6$ hosts conectados a cada departamento. Sobram assim 4 bits para sub-redes, que se constatou que corresponde ao maior número de departamentos possíveis de ser endereçados, $2^{4\ (nr\ de\ bits)} = 16$ departamentos, 12 departamentos para endereçamento futuro. Em relação a máscara, esta aumentou para 29 bits (255.255.255.248).

c) Verifique e garanta que a conectividade IP interna na rede local LEI-RC é mantida. No caso de não existência de conetividade, reveja a atribuição de endereços efetuada e eventuais erros de encaminhamento por forma a realizar as correções necessárias. Explique como procedeu.

O grupo verificou que algumas alterações tiveram de ser feitas, mais especificamente a troca de endereçamento dos 4 routers pelo endereço menor da sua sub-rede de modo a criar tabelas de endereçamento cuja entrada default tenha como gateway o router de cada um dos departamentos. Só com esta alteração foi possível a comunicação entre departamentos, pois o endereçamento para a sua própria rede é possível de ambas as formas.

Finalmente, verificamos a existência de conexão com um ping entre departamentos:

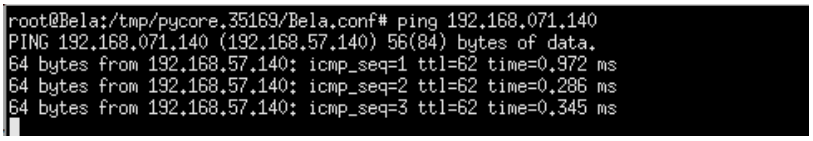


Figura 43: Comando ping entre Departamento A e B

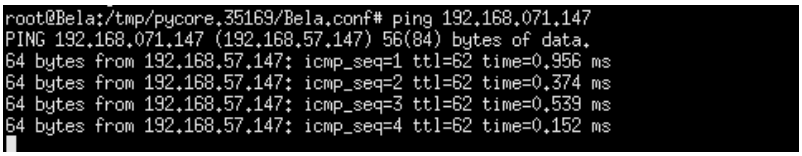


Figura 44: Comando ping entre departamento A e C

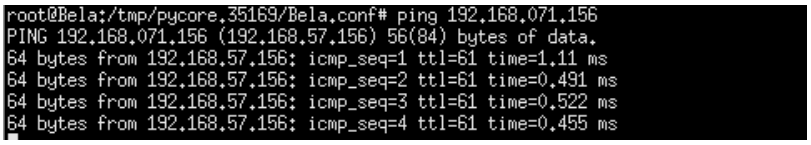


Figura 45: Comando ping entre departamento A e D

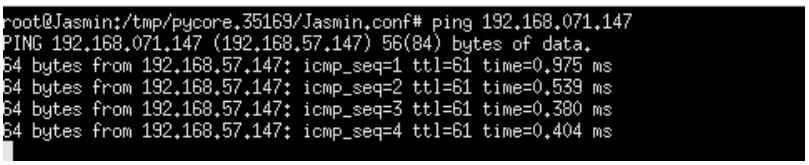


Figura 46: Comando ping entre departamento B e C

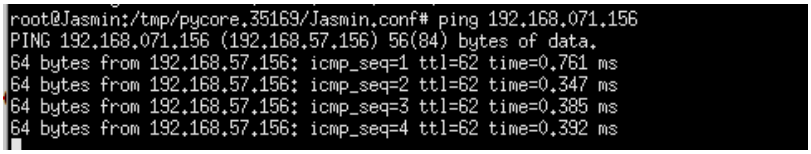


Figura 47: Comando ping entre departamento B e D

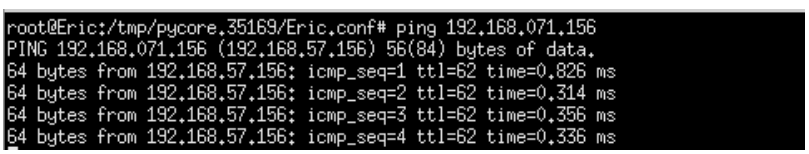


Figura 48: Comando ping entre departamento C e D

Conclusão

Terminado o trabalho prático 2, foram atingidos todos os objetivos propostos pelos docentes. Os resultados obtidos estão maioritariamente em concordância com os resultados esperados das experiências realizadas. Além disso, encontramo-nos bastante satisfeitos com o mais à-vontade sobre tópicos como endereçamento de pacotes, subnneting e tabelas de endereçamento. As dúvidas que restaram de aulas teóricas foram completamente extintas com a elaboração deste trabalho prático e esperamos que ocorra o mesmo nos próximos trabalhos de modo a construir uma base robusta para eventuais avanços na dificuldade.