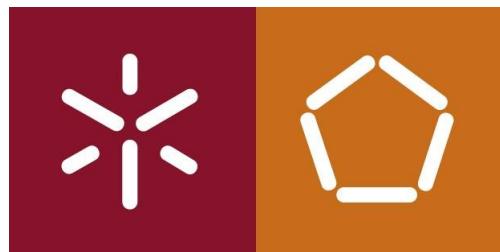


UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA



TP2 - Protocolo IPv4

REDES DE COMPUTADORES

PL 4 GRUPO 8



Carla Cruz
A80564



Diogo Sobral
A82523



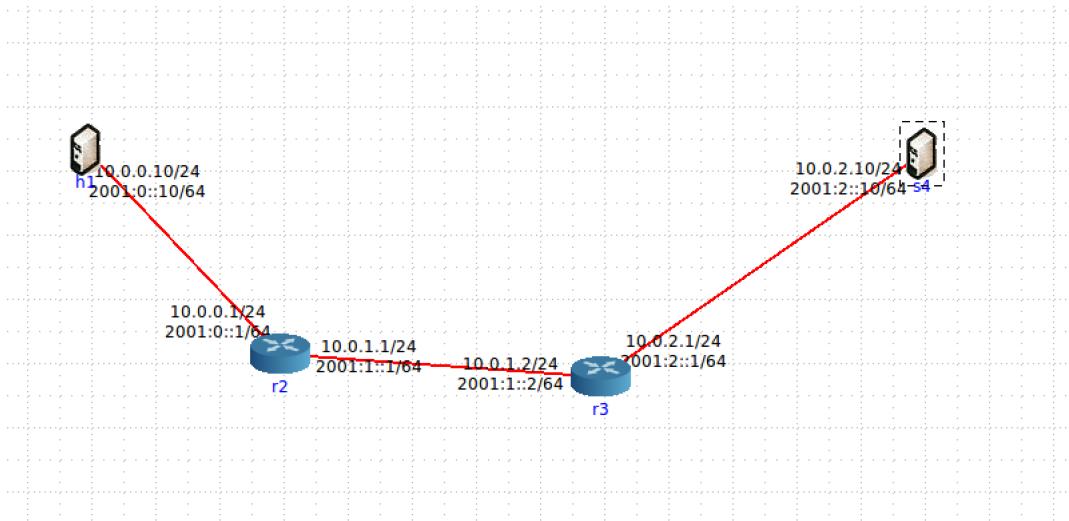
Pedro Freitas
A80975

November 12, 2018

Chapter 1

1.1 Pergunta 1

Prepare uma topologia CORE para verificar o comportamento do traceroute. Ligue um host (pc) h1 a um router r2; o router r2 a um router r3, que por sua vez, se liga a um host (servidor) s4. (Note que pode não existir conectividade IP imediata entre h1 e s4 até que o routing estabilize).



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

```
Specify "host" missing argument.
root@h1:/tmp/pycore.45869/h1.conf# traceroute -I 10.0.2.10
traceroute to 10.0.2.10 (10.0.2.10), 30 hops max, 60 byte packets
 1  A0 (10.0.0.1)  0.066 ms  0.006 ms  0.006 ms
 2  10.0.1.2 (10.0.1.2)  0.024 ms  0.007 ms  0.006 ms
 3  10.0.2.10 (10.0.2.10)  0.022 ms  0.010 ms  0.009 ms
root@h1:/tmp/pycore.45869/h1.conf#
```

- B** Registe e analise o tráfego ICMP enviado por h1 e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.

126 571.003480 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=1/256, ttl=1
127 571.003501 10.0.0.1	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
128 571.003507 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=2/512, ttl=1
129 571.003511 10.0.0.1	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
130 571.003514 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=3/768, ttl=1
131 571.003518 10.0.0.1	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
132 571.003521 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=4/1024, ttl=2
133 571.003544 10.0.1.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
134 571.003548 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=5/1280, ttl=2
135 571.003554 10.0.1.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
136 571.003556 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=6/1536, ttl=2
137 571.003562 10.0.1.2	10.0.0.10	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
138 571.003565 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=7/1792, ttl=3
139 571.003586 10.0.2.10	10.0.0.10	ICMP	74 Echo (ping) reply id=0x006f, seq=7/1792, ttl=62

Figure 1.1: Tráfego ICMP

Inicialmente, o h1 tenta comunicar com o host s4 mas não consegue visto que o TTL é inicialmente 1. O pacote chega ao r2 e é descartado, enviando uma mensagem para h1 a informar que o mesmo foi descartado. O TTL é depois aumentado sucessivamente, repetindo até chegar a TTL 3, que é o mínimo para chegar a s4.

- C** Qual deve ser o valor inicial mínimo do campo TTL para alcançar o destino s4? Verifique na prática que a sua resposta está correta.

O valor inicial do TTL deverá ser 3.

138 571.003565 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=7/1792, ttl=3
139 571.003586 10.0.2.10	10.0.0.10	ICMP	74 Echo (ping) reply id=0x006f, seq=7/1792, ttl=62

Figure 1.2: Primeiro pacote a chegar com sucesso ao h4.

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

O valor deverá ser 0.021 ms.

139 571.003586 10.0.2.10	10.0.0.10	ICMP	74 Echo (ping) reply id=0x006f, seq=7/1792, ttl=62
140 571.003590 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=8/2048, ttl=3
141 571.003598 10.0.2.10	10.0.0.10	ICMP	74 Echo (ping) reply id=0x006f, seq=8/2048, ttl=62
142 571.003601 10.0.0.10	10.0.2.10	ICMP	74 Echo (ping) request id=0x006f, seq=9/2304, ttl=3
143 571.003608 10.0.2.10	10.0.0.10	ICMP	74 Echo (ping) reply id=0x006f, seq=9/2304, ttl=62
Destination: 10.0.0.10 (10.0.0.10)			
Internet Control Message Protocol			
Type: 0 (Echo (ping) reply)			
Code: 0			
Checksum: 0x8a04 [correct]			
Identifier (BE): 111 (0x006f)			
Identifier (LE): 28416 (0x6f00)			
Sequence number (BE): 7 (0x0007)			
Sequence number (LE): 1792 (0x0700)			
[Response To: 138]			
[Response Time: 0.021 ms]			

Figure 1.3: Valor do Round-Trip Time

1.2 Pergunta 2

Selecionando a primeira mensagem ICMP capturada (referente a (i) tamanho por defeito).

13	8.955742	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
14	9.027562	172.26.254.254	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
15	9.028753	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)
16	9.035378	172.26.254.254	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
17	9.035562	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=3/768, ttl=1 (no response found!)
18	9.039060	172.26.254.254	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
19	9.039251	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=4/1024, ttl=2 (no response found!)
20	9.040889	172.16.2.1	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
76	10.054783	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=5/1280, ttl=2 (no response found!)
77	10.058471	172.16.2.1	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
78	10.058612	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=6/1536, ttl=2 (no response found!)
79	10.060270	172.16.2.1	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
80	10.060419	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=7/1792, ttl=3 (no response found!)
81	10.063689	172.16.115.252	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
90	11.079396	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=8/2048, ttl=3 (no response found!)
91	11.082755	172.16.115.252	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
92	11.082925	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=9/2304, ttl=3 (no response found!)
93	11.088665	172.16.115.252	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
94	11.088219	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=10/2560, ttl=4 (reply in 95)
95	11.090599	193.136.9.240	172.26.89.69	ICMP	86 Echo (ping) reply id=0x88cf, seq=10/2560, ttl=61 (request in 94)
98	11.094232	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=11/2816, ttl=4 (reply in 99)
99	11.096182	193.136.9.240	172.26.89.69	ICMP	86 Echo (ping) reply id=0x88cf, seq=11/2816, ttl=61 (request in 98)
100	11.096336	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=12/3072, ttl=4 (reply in 101)
101	11.098311	193.136.9.240	172.26.89.69	ICMP	86 Echo (ping) reply id=0x88cf, seq=12/3072, ttl=61 (request in 100)

Figure 1.4: Endereço IP da interface ativa

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

13	8.955742	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
14	9.027562	172.26.254.254	172.26.89.69	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
15	9.028753	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)

Figure 1.5: Endereço IP da interface ativa

Como podemos ver pela análise da imagem 1.5, o endereço IP da interface ativa do nosso computador é 172.26.89.69 .

B Qual é o valor do campo protocolo? O que identifica?

```
> Frame 13: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
> Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
> Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 72
  Identification: 0x88d0 (35024)
> Flags: 0x0000
> Time to live: 1
  Protocol: ICMP (1)
  Header checksum: 0x600d [validation disabled]
  [Header checksum status: Unverified]
  Source: 172.26.89.69
  Destination: 193.136.9.240
```

Figure 1.6: Valor do campo protocolo

Como podemos ver nos valores do datagrama da figura 1.6, o campo do protocolo tem o valor 1 representando este o protocolo ICMP - Internet Control Message Protocol.

C Quantos bytes tem o cabeçalho IP(v4)? Quantos bytes tem o campo de dados(payload) do datagrama? Como se calculado tamanho do payload?

```
Frame 13: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 72
    Identification: 0x88d0 (35024)
    ▶ Flags: 0x0000
```

Figure 1.7: Header Length

Como podemos ver na figura acima, na parte a escurecido, o cabeçalho IP(v4) tem 20 bytes.

```
Frame 13: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 72
    Identification: 0x88d0 (35024)
    ▶ Flags: 0x0000
    ▶ Time to live: 1
    Protocol: ICMP (1)
```

Figure 1.8: Total Length

O campo de dados do datagrama tem 52 bytes. Este valor é calculado fazendo a diferença entre o tamanho total (72 bytes) e o tamanho do cabeçalho (20 bytes).

D O datagrama IP foi fragmentado? Justifique.

```
▶ Frame 13: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▶ Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
▼ Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 72
    Identification: 0x88d0 (35024)
    ▶ Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0... .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    ▶ Time to live: 1
```

Figure 1.9: Flags

Na figura acima podemos ver que o valor das flags e o valor do offset estão a 0 e, por isso, podemos concluir que o pacote não foi fragmentado.

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.

13 8.955742	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
15 9.028753	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)
17 9.035562	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=3/768, ttl=1 (no response found!)
19 9.039251	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=4/1024, ttl=2 (no response found!)
76 10.054783	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=5/1280, ttl=2 (no response found!)
78 10.058612	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=6/1536, ttl=2 (no response found!)
80 10.060419	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=7/1792, ttl=3 (no response found!)
90 11.079396	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=8/2048, ttl=3 (no response found!)
92 11.082925	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=9/2304, ttl=3 (no response found!)
94 11.088219	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=10/2560, ttl=4 (reply in 95)
98 11.094232	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=11/2816, ttl=4 (reply in 99)
100 11.096336	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=12/3072, ttl=4 (reply in 101)

Figure 1.10: Pacotes Ordenados

13 8.955742	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
15 9.028753	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)
17 9.035562	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=3/768, ttl=1 (no response found!)
19 9.039251	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=4/1024, ttl=2 (no response found!)
76 10.054783	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=5/1280, ttl=2 (no response found!)
78 10.058612	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=6/1536, ttl=2 (no response found!)
80 10.060419	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=7/1792, ttl=3 (no response found!)
90 11.079396	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=8/2048, ttl=3 (no response found!)
92 11.082925	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=9/2304, ttl=3 (no response found!)
94 11.088219	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=10/2560, ttl=4 (reply in 95)
98 11.094232	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=11/2816, ttl=4 (reply in 99)
100 11.096336	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=12/3072, ttl=4 (reply in 101)

Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdEnt_ff:94:00 (00:d0:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 72
Identification: 0x88d0 (35024)
Flags: 0x0000
0... = Reserved bit: Not set
.0... = Don't fragment: Not set
..0... = More fragments: Not set
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 1
Protocol: ICMP (1)

Figure 1.11: Pacote 1

78 10.058612	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=6/1536, ttl=2 (no response found!)
80 10.060419	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=7/1792, ttl=3 (no response found!)
90 11.079396	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=8/2048, ttl=3 (no response found!)
92 11.082925	172.26.89.69	193.136.9.240	ICMP	86 Echo (ping) request id=0x88cf, seq=9/2304, ttl=3 (no response found!)

Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdEnt_ff:94:00 (00:d0:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 72
Identification: 0x88d5 (35029)
Flags: 0x0000
0... = Reserved bit: Not set
.0... = Don't fragment: Not set
..0... = More fragments: Not set
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 2
Protocol: ICMP (1)

Figure 1.12: Pacote 2

Pela análise das imagens, vemos que o valor dos campos de Identification e de TTL variam ao longo do tempo.

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

```

13 8.955742 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
15 9.028753 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)
17 9.035562 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=3/768, ttl=1 (no response found!)
19 9.039251 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=4/1024, ttl=2 (no response found!)
76 10.054783 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=5/1280, ttl=2 (no response found!)
78 10.058612 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=6/1536, ttl=2 (no response found!)
80 10.060419 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=7/1792, ttl=3 (no response found!)
90 11.079396 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=8/2048, ttl=3 (no response found!)
92 11.082925 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=9/2304, ttl=3 (no response found!)
94 11.088219 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=10/2560, ttl=4 (reply in 95)
98 11.094232 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=11/2816, ttl=4 (reply in 99)
100 11.096336 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=12/3072, ttl=4 (reply in 101)
95 11.090599 193.136.9.240 172.26.89.69 ICMP 86 Echo (ping) reply id=0x88cf, seq=10/2560, ttl=61 (request in 94)

frame 13: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:00:00:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 72
Identification: 0x88d0 (35024)

```

Figure 1.13: Pacote 3

```

13 8.955742 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
15 9.028753 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)
17 9.035562 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=3/768, ttl=1 (no response found!)
19 9.039251 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=4/1024, ttl=2 (no response found!)
76 10.054783 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=5/1280, ttl=2 (no response found!)
78 10.058612 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=6/1536, ttl=2 (no response found!)
80 10.060419 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=7/1792, ttl=3 (no response found!)
90 11.079396 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=8/2048, ttl=3 (no response found!)
92 11.082925 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=9/2304, ttl=3 (no response found!)
94 11.088219 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=10/2560, ttl=4 (reply in 95)
98 11.094232 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=11/2816, ttl=4 (reply in 99)
100 11.096336 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=12/3072, ttl=4 (reply in 101)
95 11.090599 193.136.9.240 172.26.89.69 ICMP 86 Echo (ping) reply id=0x88cf, seq=10/2560, ttl=61 (request in 94)

frame 15: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:00:00:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 72
Identification: 0x88d1 (35025)

```

Figure 1.14: Pacote 4

```

13 8.955742 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
15 9.028753 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)
17 9.035562 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=3/768, ttl=1 (no response found!)
19 9.039251 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=4/1024, ttl=2 (no response found!)
76 10.054783 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=5/1280, ttl=2 (no response found!)
78 10.058612 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=6/1536, ttl=2 (no response found!)
80 10.060419 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=7/1792, ttl=3 (no response found!)
90 11.079396 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=8/2048, ttl=3 (no response found!)
92 11.082925 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=9/2304, ttl=3 (no response found!)
94 11.088219 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=10/2560, ttl=4 (reply in 95)
98 11.094232 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=11/2816, ttl=4 (reply in 99)
100 11.096336 172.26.89.69 193.136.9.240 ICMP 86 Echo (ping) request id=0x88cf, seq=12/3072, ttl=4 (reply in 101)
95 11.090599 193.136.9.240 172.26.89.69 ICMP 86 Echo (ping) reply id=0x88cf, seq=10/2560, ttl=61 (request in 94)

frame 15: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:00:00:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 72
Identification: 0x88d1 (35025)

```

Figure 1.15: TTL a alterar

Analizando com mais cuidado, a imagem 1.15 mostra-nos que o ttl é incrementado de 3 em 3. A imagens do Pacote 3 e do Pacote são imagens que dois pacotes consecutivos e podemos verificar que o campo da Identificação é incrementado a cada pacote enviado.

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?

14	9.027562	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
16	9.035378	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18	9.039060	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
20	9.040889	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
77	10.058471	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
79	10.060270	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
81	10.063609	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
91	11.082755	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
93	11.088065	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

Figure 1.16: Pacotes ordenados pelo destino

16	9.035378	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18	9.039060	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
20	9.040889	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
77	10.058471	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
79	10.060270	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
81	10.063609	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
91	11.082755	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
93	11.088065	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
95	11.090599	193.136.9.240	172.26.89.69	ICMP	86	Echo (ping) reply id=0x88cf, seq=10/2560, ttl=61 (request in 94)
99	11.096182	193.136.9.240	172.26.89.69	ICMP	86	Echo (ping) reply id=0x88cf, seq=11/2816, ttl=61 (request in 98)
101	11.098311	193.136.9.240	172.26.89.69	ICMP	86	Echo (ping) reply id=0x88cf, seq=12/3072, ttl=61 (request in 100)
13	8.955742	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=1/256, ttl=1 (no response found!)
15	9.028753	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=2/512, ttl=1 (no response found!)
17	9.035562	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=3/768, ttl=1 (no response found!)
90	9.039261	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=4/1024, ttl=1 (no response found!)
92	9.040889	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=5/1280, ttl=1 (no response found!)
94	9.042801	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=6/1536, ttl=1 (no response found!)
96	9.044713	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=7/1792, ttl=1 (no response found!)
98	9.046635	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=8/2048, ttl=1 (no response found!)
100	9.048557	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=9/2304, ttl=1 (no response found!)
102	9.050479	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=10/2560, ttl=1 (no response found!)
104	9.052391	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=11/2816, ttl=1 (no response found!)
106	9.054313	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=12/3072, ttl=1 (no response found!)
108	9.056235	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=13/3328, ttl=1 (no response found!)
110	9.058157	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=14/3584, ttl=1 (no response found!)
112	9.060079	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=15/3840, ttl=1 (no response found!)
114	9.061991	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=16/4096, ttl=1 (no response found!)
116	9.063913	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=17/4352, ttl=1 (no response found!)
118	9.065835	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=18/4608, ttl=1 (no response found!)
120	9.067757	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=19/4864, ttl=1 (no response found!)
122	9.069679	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=20/5120, ttl=1 (no response found!)
124	9.071591	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=21/5376, ttl=1 (no response found!)
126	9.073513	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=22/5632, ttl=1 (no response found!)
128	9.075435	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=23/5888, ttl=1 (no response found!)
130	9.077357	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=24/6144, ttl=1 (no response found!)
132	9.079279	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=25/6400, ttl=1 (no response found!)
134	9.081191	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=26/6656, ttl=1 (no response found!)
136	9.083113	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=27/6912, ttl=1 (no response found!)
138	9.085035	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=28/7168, ttl=1 (no response found!)
140	9.086957	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=29/7424, ttl=1 (no response found!)
142	9.088879	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=30/7680, ttl=1 (no response found!)
144	9.090791	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=31/7936, ttl=1 (no response found!)
146	9.092713	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=32/8192, ttl=1 (no response found!)
148	9.094635	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=33/8448, ttl=1 (no response found!)
150	9.096557	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=34/8704, ttl=1 (no response found!)
152	9.098479	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=35/8960, ttl=1 (no response found!)
154	9.100391	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=36/9216, ttl=1 (no response found!)
156	9.102313	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=37/9472, ttl=1 (no response found!)
158	9.104235	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=38/9728, ttl=1 (no response found!)
160	9.106157	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=39/9984, ttl=1 (no response found!)
162	9.108079	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=40/10240, ttl=1 (no response found!)
164	9.110001	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=41/10496, ttl=1 (no response found!)
166	9.111923	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=42/10752, ttl=1 (no response found!)
168	9.113845	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=43/11008, ttl=1 (no response found!)
170	9.115767	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=44/11264, ttl=1 (no response found!)
172	9.117689	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=45/11520, ttl=1 (no response found!)
174	9.119611	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=46/11776, ttl=1 (no response found!)
176	9.121533	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=47/12032, ttl=1 (no response found!)
178	9.123455	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=48/12288, ttl=1 (no response found!)
180	9.125377	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=49/12544, ttl=1 (no response found!)
182	9.127299	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=50/12800, ttl=1 (no response found!)
184	9.129221	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=51/13056, ttl=1 (no response found!)
186	9.131143	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=52/13312, ttl=1 (no response found!)
188	9.133065	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=53/13568, ttl=1 (no response found!)
190	9.134987	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=54/13824, ttl=1 (no response found!)
192	9.136909	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=55/14080, ttl=1 (no response found!)
194	9.138831	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=56/14336, ttl=1 (no response found!)
196	9.140753	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=57/14592, ttl=1 (no response found!)
198	9.142675	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=58/14848, ttl=1 (no response found!)
200	9.144597	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=59/15104, ttl=1 (no response found!)
202	9.146519	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=60/15360, ttl=1 (no response found!)
204	9.148441	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=61/15616, ttl=1 (no response found!)
206	9.150363	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=62/15872, ttl=1 (no response found!)
208	9.152285	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=63/16128, ttl=1 (no response found!)
210	9.154207	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=64/16384, ttl=1 (no response found!)
212	9.156129	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=65/16640, ttl=1 (no response found!)
214	9.158051	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=66/16896, ttl=1 (no response found!)
216	9.160973	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=67/17152, ttl=1 (no response found!)
218	9.162895	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=68/17408, ttl=1 (no response found!)
220	9.164817	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=69/17664, ttl=1 (no response found!)
222	9.166739	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=70/17920, ttl=1 (no response found!)
224	9.168661	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=71/18176, ttl=1 (no response found!)
226	9.170583	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=72/18432, ttl=1 (no response found!)
228	9.172505	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=73/18688, ttl=1 (no response found!)
230	9.174427	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=74/18944, ttl=1 (no response found!)
232	9.176349	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=75/19200, ttl=1 (no response found!)
234	9.178271	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=76/19456, ttl=1 (no response found!)
236	9.180193	172.26.89.69	193.136.9.240	ICMP	86	Echo (ping) request id=0x88cf, seq=77/19712, ttl=1 (no response found!)
238	9.182115	172.26.89.69	193.136.9.240	ICMP	86	

As figuras 1.17 e 1.18 mostram os valores dos ttl. Os primeiros três pacotes têm ttl 255, os segundos três 254 e os últimos têm o valor 253. Anteriormente foi dito que os valores do ttl variam ao longo do tempo. Esta variação deve-se ao facto que à medida que os pacotes são enviados pelo nosso router, o ttl também é aumentado fazendo com que os mesmos cheguem cada vez mais longe. Quando é preciso emitir uma mensagem de erro, os pacotes vêm de routers mais distantes fazendo com que ,no caminho de regresso para o nosso router, o pacote passe por mais routers intermédios e, por isso, o ttl das mensagens de erro seja cada vez menor, já que a cada router por onde passa o seu ttl é decrementado.

1.3 Pergunta 3

Pretende-se agora analisar a fragmentação de pacotes IP.

	Time	Source	Destination	Protocol	Length	Info
17	0.731902	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=1/256, ttl=1 (no response found!)
18	0.736576	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
21	0.737591	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=2/512, ttl=1 (no response found!)
22	0.740891	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
25	0.740999	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=3/768, ttl=1 (no response found!)
27	0.742988	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
30	0.743189	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=4/1024, ttl=2 (no response found!)
31	0.745473	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
40	1.757919	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=5/1280, ttl=2 (no response found!)
41	1.762842	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
44	1.763088	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=6/1536, ttl=2 (no response found!)
45	1.764800	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
48	1.764966	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=7/1792, ttl=3 (no response found!)
49	1.767728	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
58	2.777117	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=8/2048, ttl=3 (no response found!)
59	2.780835	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
62	2.780995	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=9/2304, ttl=3 (no response found!)
63	2.784362	172.16.115.252	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
66	2.784536	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=10/2560, ttl=4 (reply in 69)
69	2.788143	193.136.9.240	172.26.89.69	ICMP	602	Echo (ping) reply id=0x8937, seq=10/2560, ttl=61 (request in 66)
72	2.789029	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=11/2816, ttl=4 (reply in 75)
75	2.793425	193.136.9.240	172.26.89.69	ICMP	1514	Echo (ping) reply id=0x8937, seq=11/2816, ttl=61 (request in 72)
78	2.793586	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=12/3072, ttl=4 (reply in 81)
81	2.796626	193.136.9.240	172.26.89.69	ICMP	1514	Echo (ping) reply id=0x8937, seq=12/3072, ttl=61 (request in 78)

Figure 1.19: Resultado do traceroute

15	0.731901	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=8938) [Reassembled in 1514]
16	0.731902	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8938) [Reassembled in 1514]
17	0.731902	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=1/256, ttl=1 (no response found!)
18	0.736576	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
19	0.737589	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=8939) [Reassembled in 1514]
20	0.737591	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8939) [Reassembled in 1514]
21	0.737591	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=2/512, ttl=1 (no response found!)
22	0.740891	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
23	0.740997	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=893a) [Reassembled in 1514]
24	0.740998	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=893a) [Reassembled in 1514]
25	0.740999	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=3/768, ttl=1 (no response found!)
26	0.741565	172.26.89.69	224.0.0.251	MDNS	79	Standard query 0x36b3 PTR _arduino._tcp.local, "0M" question
27	0.742980	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
28	0.743108	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=893b) [Reassembled in 1514]
29	0.743109	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=893b) [Reassembled in 1514]
30	0.743109	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=12/3072, ttl=2 (no response found!)
31	0.745473	172.16.2.1	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
32	0.756540	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=893c) [Reassembled in 1514]
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240						
0100 = Version: 4					
.... 0101	= Header Length: 20 bytes (5)					
► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 1500						
Identification: 0x8938 (35128)						
▼ Flags: 0x2000, More fragments						
0... = Reserved bit: Not set						
.0.. = Don't fragment: Not set						
.1... = More fragments: Set						
...0 0000 0000 0000 0000 = Fragment offset: 0						
► Time to live: 1						
Dest-Src: 172.26.89.69 -> 193.136.9.240						

Figure 1.20: Pacotes com fragmentação

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

15	0.731901	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=8938) [Reassembled in 1514]
16	0.731902	172.26.89.69	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8938) [Reassembled in 1514]
17	0.731902	172.26.89.69	193.136.9.240	ICMP	602	Echo (ping) request id=0x8937, seq=1/256, ttl=1 (no response found!)
18	0.736576	172.26.254.254	172.26.89.69	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

Figure 1.21: Pacote 1 e fragmentação

O pacote que tentamos enviar tinha o tamanho inicial 3548. Uma vez que, a MTU que estamos a usar só consegue enviar pacotes com tamanho 1500, o nosso pacote tem que ser fragmentado para poder ser enviado.

- 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?

```

15 0.731901 172.26.89.69 193.136.9.240 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8938) [Reassembled in #]
16 0.731902 172.26.89.69 193.136.9.240 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8938) [Reassembled in #]
17 0.731902 172.26.89.69 193.136.9.240 ICMP 602 Echo (ping) request id=0x8937, seq=1/256, ttl=1 (no response found!)

Frame 15: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x8938 (35128)
    ▶ Flags: 0x2000, More fragments
        0... .... .... .... = Reserved bit: Not set
        .0... .... .... .... = Don't fragment: Not set
        ..1.... .... .... = More fragments: Set
        ...0 0000 0000 0000 = Fragment offset: 0
    ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x3a11 [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.26.89.69
    Destination: 193.136.9.240
    Reassembled IPv4 in frame: 17
Data (1480 bytes)

```

Figure 1.22: Flags do Pacote 1

O primeiro fragmento é indicado pelos valores das flags quando o offset tem o valor 0 e o valor do more fragments é 1. Como podemos ver na imagem 1.22, o fragmento selecionado tem estes valores nas flags logo é o primeiro fragmento. Sabemos que o segmento foi fragmento quanto o valor das flags é diferente de 0. O tamanho deste datagrama é 1500.

- 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?

```

15 0.731901 172.26.89.69 193.136.9.240 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8938) [Reassembled in #]
16 0.731902 172.26.89.69 193.136.9.240 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8938) [Reassembled in #]
17 0.731902 172.26.89.69 193.136.9.240 ICMP 602 Echo (ping) request id=0x8937, seq=1/256, ttl=1 (no response found!)

Frame 16: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x8938 (35128)
    ▶ Flags: 0x2000, More fragments
        0... .... .... .... = Reserved bit: Not set
        .0... .... .... .... = Don't fragment: Not set
        ..1.... .... .... = More fragments: Set
        ...0 0000 1011 1001 = Fragment offset: 185
    ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x3958 [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.26.89.69
    Destination: 193.136.9.240
    Reassembled IPv4 in frame: 17
Data (1480 bytes)

```

Figure 1.23: Segundo Fragmento

Como foi dito anteriormente, sabemos que um fragmento é o primeiro quando o offset tem o valor 0 e o more fragments é 1. Como o valor do offset é diferente de 0 significa que este fragmento não é o primeiro. Existem mais fragmentos uma vez que o valor da flag more fragments é 1.

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

O último fragmento é dado pela flag de more fragments igual a zero e o valor do offset diferente de zero. Como este fragmento é o último podemos concluir que o datagrama original foi fragmentado em 3.

```
• 15 0.731901 172.26.89.69 193.136.9.240 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8938) [Reassembled in #]
• 16 0.731902 172.26.89.69 193.136.9.240 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8938) [Reassembled in #]
• 17 0.731902 172.26.89.69 193.136.9.240 ICMP 602 Echo (ping) request id=0x8937, seq=1/256, ttl=1 (no response found!)
Frame 17: 602 bytes on wire (4816 bits), 602 bytes captured (4816 bits) on interface 0
Ethernet II, Src: Apple_d4:b0:59 (c4:b3:01:d4:b0:59), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
Internet Protocol Version 4, Src: 172.26.89.69, Dst: 193.136.9.240
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCH: CS0, ECN: Not-ECT)
        Total Length: 588
        Identification: 0x8938 (35128)
    Flags: 0x0172
        0... .... .... = Reserved bit: Not set
        .0.. .... .... = Don't fragment: Not set
        ..0. .... .... = More fragments: Not set
        ...0 0001 0111 0010 = Fragment offset: 370
    Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x5c2f [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.26.89.69
```

Figure 1.24: 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.

Entre os diferentes segmentos, os campos que mudam são os valores das flags nomeadamente, o valor do offset e da flag more segments. A flag more segments permite-se saber se ainda há ou não segmentos do datagrama original a circular na rede. O campo offset serve para saber por que ordem devem ser juntos os fragmentos de modo a ter o datagrama original. A ordem dos fragmentos é dada pelos valores do offset por ordem crescente.

Chapter 2

2.1 Pergunta 1

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

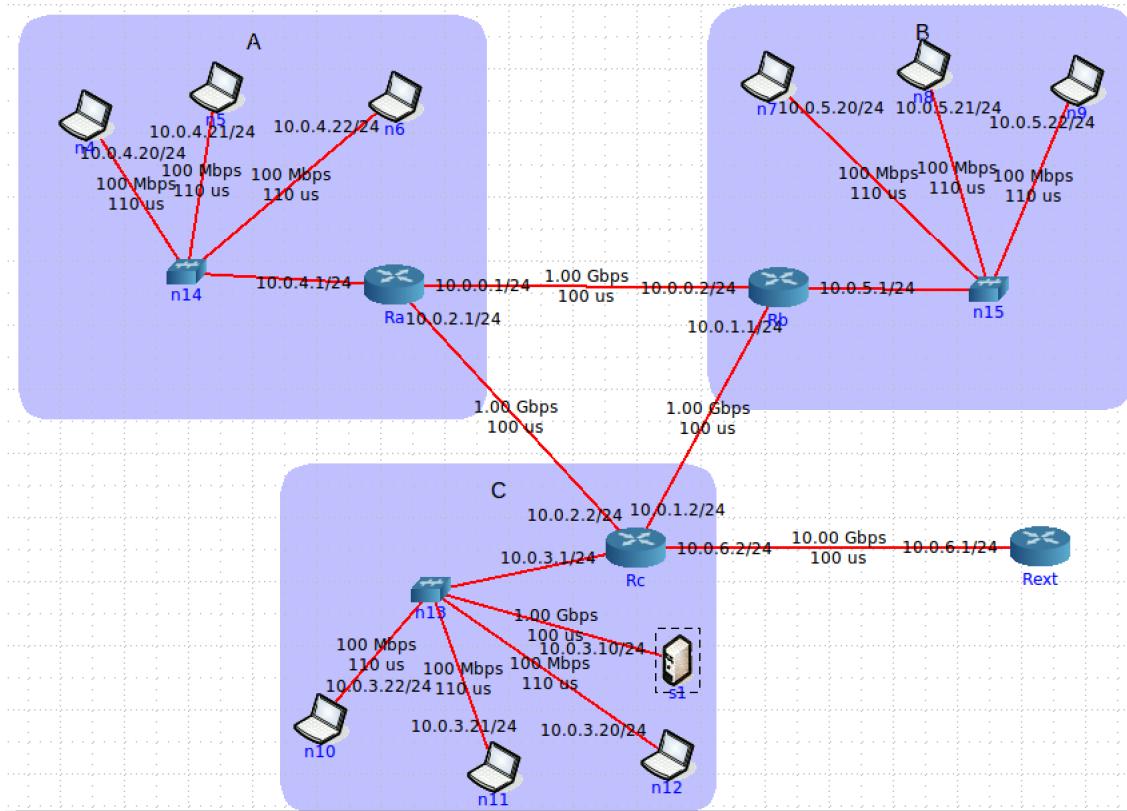


Figure 2.1: Equipamentos e Departamentos

- 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.

A máscara utilizada foi: 255.255.255.0 , visto que todos os equipamentos contêm nos seus endereços /24 . Os endereços de cada equipamento podem ser vistos na figura 2.1.

B Tratam-se de endereços públicos ou privados? Porquê?

Todos os endereços entre 10.0.0.0 e 10.255.255.255 são endereço privados. Como podemos ver na figura 2.1, todos os equipamento começam com 10. logo são todos endereços privados.

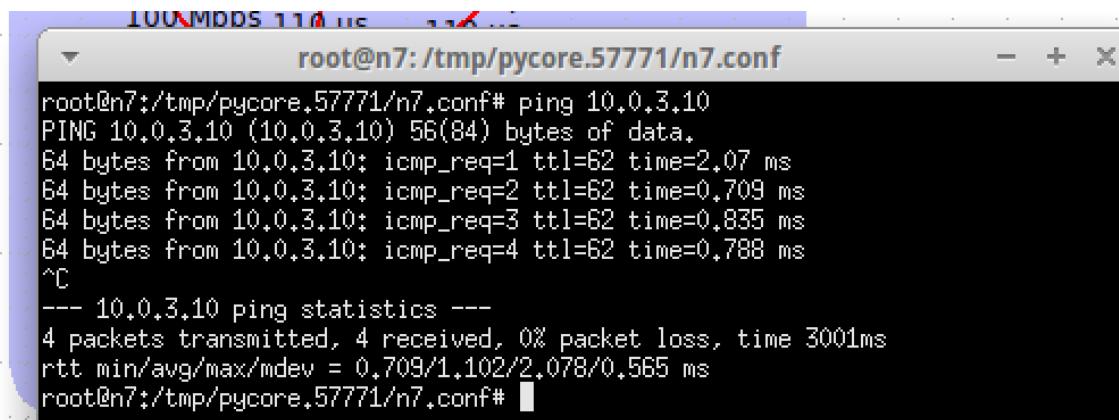
C Porque razão não é atribuído um endereço IP aos switches?

Um switch faz a interligação de equipamentos de uma rede, sendo esta uma das suas principais funcionalidades. Estes registam o endereço MAC dos dispositivos que se encontram ligados a si . Depois este endereço é analisado e associa as máquinas a que está ligado às respetivas entradas físicas do equipamento, sendo a informação diretamente enviada para o destino correspondente. Dado este funcionamento, não existe necessidade de atribuir um endereço IP ao switch, pois este apenas decide para onde vão os pacotes após ter sido realizada a análise ao endereço MAC de cada equipamento ligado a si.

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).

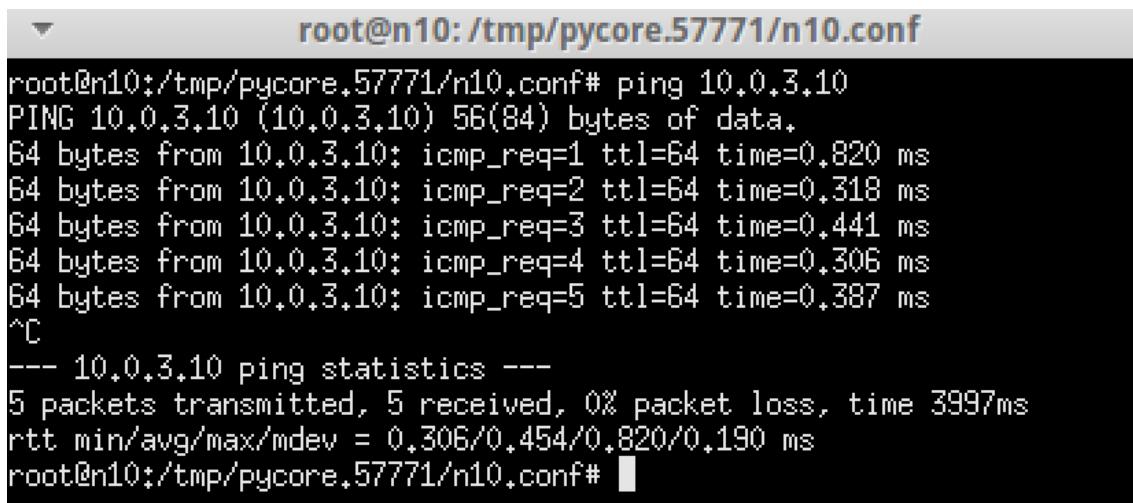
```
root@n6:/tmp/pycore.57771/n6.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_req=1 ttl=62 time=0.825 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=62 time=0.786 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=62 time=0.889 ms
64 bytes from 10.0.3.10: icmp_req=4 ttl=62 time=0.640 ms
64 bytes from 10.0.3.10: icmp_req=5 ttl=62 time=0.581 ms
64 bytes from 10.0.3.10: icmp_req=6 ttl=62 time=0.790 ms
64 bytes from 10.0.3.10: icmp_req=7 ttl=62 time=0.841 ms
64 bytes from 10.0.3.10: icmp_req=8 ttl=62 time=0.597 ms
64 bytes from 10.0.3.10: icmp_req=9 ttl=62 time=0.741 ms
64 bytes from 10.0.3.10: icmp_req=10 ttl=62 time=0.645 ms
64 bytes from 10.0.3.10: icmp_req=11 ttl=62 time=0.783 ms
^C
--- 10.0.3.10 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10000ms
rtt min/avg/max/mdev = 0.581/0.738/0.889/0.100 ms
root@n6:/tmp/pycore.57771/n6.conf# █
```

Figure 2.2: Ping Departamento A para S1



```
root@n7:/tmp/pycore.57771/n7.conf
root@n7:/tmp/pycore.57771/n7.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_req=1 ttl=62 time=2.07 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=62 time=0.709 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=62 time=0.835 ms
64 bytes from 10.0.3.10: icmp_req=4 ttl=62 time=0.788 ms
^C
--- 10.0.3.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.709/1.102/2.078/0.565 ms
root@n7:/tmp/pycore.57771/n7.conf#
```

Figure 2.3: Ping Departamento B para S1



```
root@n10:/tmp/pycore.57771/n10.conf
root@n10:/tmp/pycore.57771/n10.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_req=1 ttl=64 time=0.820 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=64 time=0.318 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=64 time=0.441 ms
64 bytes from 10.0.3.10: icmp_req=4 ttl=64 time=0.306 ms
64 bytes from 10.0.3.10: icmp_req=5 ttl=64 time=0.387 ms
^C
--- 10.0.3.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.306/0.454/0.820/0.190 ms
root@n10:/tmp/pycore.57771/n10.conf#
```

Figure 2.4: Ping Departamento C para S1

Como podemos ver pelas imagens acima apresentadas(2.2,2.3,2.4), equipamentos de cada departamento conseguem receber e transmitir packets entre eles e o servidor do departamento C, certificando-se assim a existência de conectividade entre os mesmos.

E Verifique se existe conectividade IP do router de acesso Rext para o servidor S1.

Aplicando-se o mesmo raciocínio da pergunta anterior, verificamos a existência de conectividade do router de acesso (Rext) e o servidor (S1).

```

root@Rext:/tmp/pycore.57771/Rext.conf
root@Rext:/tmp/pycore.57771/Rext.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_req=1 ttl=63 time=0.869 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=63 time=0.630 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=63 time=0.628 ms
64 bytes from 10.0.3.10: icmp_req=4 ttl=63 time=0.629 ms
^C
--- 10.0.3.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.628/0.689/0.869/0.103 ms
root@Rext:/tmp/pycore.57771/Rext.conf#

```

Figure 2.5: Ping Rext para S1

2.2 Pergunta 2

Para o router e um laptop do departamento A:

- 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@n6:/tmp/pycore.57771/n6.conf
root@n6:/tmp/pycore.57771/n6.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@n6:/tmp/pycore.57771/n6.conf#

```

Figure 2.6: Tabela de encaminhamento laptop A

```

root@Ra:/tmp/pycore.57771/Ra.conf
root@Ra:/tmp/pycore.57771/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        0.0.0.0        255.255.255.0  U         0 0          0 eth1
10.0.3.0        10.0.2.2       255.255.255.0  UG       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        10.0.2.2       255.255.255.0  UG       0 0          0 eth1
root@Ra:/tmp/pycore.57771/Ra.conf#

```

Figure 2.7: Tabela de encaminhamento router A

A tabela do laptop A (2.6) tem apenas duas entradas. A linha com destination 0.0.0.0 é o endereço default que é usado quando não se sabe para onde deve ser enviado um pacote. A outra linha tem a destination 10.0.4.0 que é utilizada quando o laptop pretende enviar um pacote para a própria rede.

A tabela de endereçamento do router tem as redes a que consegue chegar e os respetivos gateways. Quanto o destino do pacote é uma das redes presentes na coluna Destination o pacote é enviado pelo respetivo valor do Gateway.

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

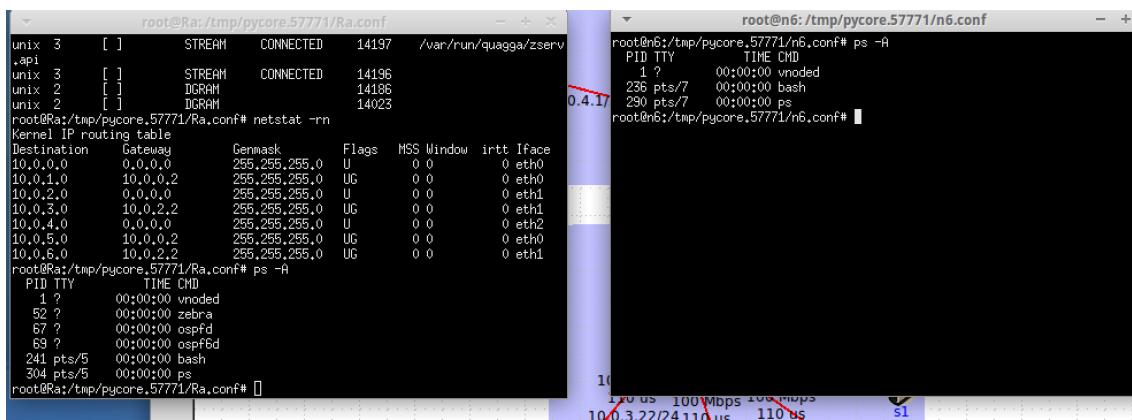


Figure 2.8: Processos a correr no router (esquerda) e no laptop (direita)

O router está a usar encaminhamento dinâmico enquanto que o laptop está a usar encaminhamento estático.

No encaminhamento dinâmico os routers trocam informação de routing entre si, sendo as rotas atualizadas ao longo do tempo. Esta atualização de rotas é obtida através de protocolos específicos de encaminhamento. Como podemos ver na imagem da esquerda, vemos que existem processos a correr os protocolos OSPF e ZEBRA, o que nos permite concluir que de facto o router está a usar encaminhamento dinâmico.

No encaminhamento estático as rotas permanecem fixas e são baseadas nas rotas pré-definidas. Por isso não existe nenhum processo a correr além dos da própria máquina. Analisando a imagem da direita, vemos que os processos que estão a correr são os processos básicos da máquina, podendo concluir assim que o laptop está a usar encaminhamento estático.

- 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.

```

root@s1:/tmp/pycore.51525/s1.conf#
root@S1:/tmp/pycore.51525/s1.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
0.0.0.0         10.0.3.1       0.0.0.0       UG        0 0          0 eth0
10.0.3.0        0.0.0.0        255.255.255.0 U          0 0          0 eth0
root@S1:/tmp/pycore.51525/s1.conf# route delete default
root@S1:/tmp/pycore.51525/s1.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
10.0.3.0        0.0.0.0        255.255.255.0 U          0 0          0 eth0
root@S1:/tmp/pycore.51525/s1.conf# 

```

Figure 2.9: Tabela de Encaminhamento do Servidor S1

O servidor deixa de ser acessível por fora da rede 10.0.3.0. Com o comando route delete default a rota por defeito é retirada da tabela de encaminhamento do servidor, sendo por isso, inutilizável. Assim a ligação entre o servidor e o router do departamento C é cortada. Consequentemente os outros departamentos e o router de acesso não conseguem aceder ao Servidor pois o seu ponto de acesso seria através do router do departamento C. No entanto os laptops do próprio departamento continuam a ter acesso pois estes estão conectados através do switch, e os laptops dos vários departamentos conseguem comunicar entre si na mesma.

- 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.

```

root@s1:/tmp/pycore.51525/s1.conf# route add -net 10.0.4.0 netmask 255.255.255.0
gw 10.0.3.1
root@s1:/tmp/pycore.51525/s1.conf# route add -net 10.0.5.0 netmask 255.255.255.0
gw 10.0.3.1

```

Figure 2.10: Adicionar Rota - Departamentos A e B

```

root@s1:/tmp/pycore.51525/s1.conf# route add -net 10.0.6.0 netmask 255.255.255.0
gw 10.0.3.1

```

Figure 2.11: Adicionar Rota - Router de Acesso

Como podemos ver pelas figuras 2.10 e 2.11 o comando usado para restaurar a conexão foi: route add -net 10.0.X.0 netmask 255.255.255.0 gw 10.0.3.1 . Com este comando são criadas rotas entre o servidor e os vários departamentos e router de acesso.

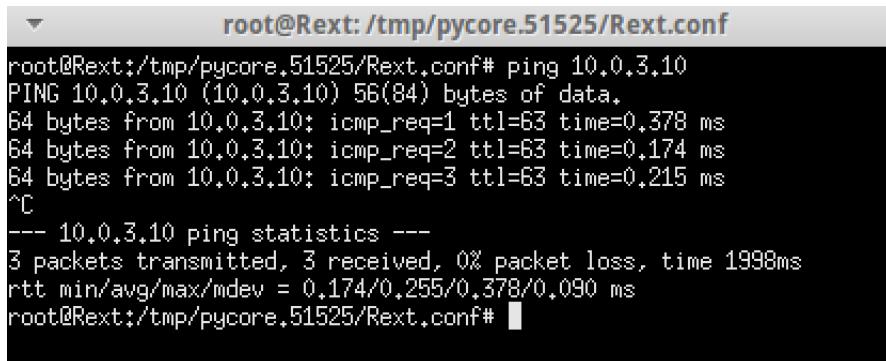
E Teste a nova política de encaminhamento garantindo que o servidor está novamente acessível, utilizando para o efeito o comando ping. Registe a nova tabela de encaminhamento do servidor.

```
root@n6:/tmp/pycore.51525/n6.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_req=1 ttl=62 time=0.910 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=62 time=0.627 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=62 time=0.622 ms
^C
--- 10.0.3.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.622/0.719/0.910/0.138 ms
root@n6:/tmp/pycore.51525/n6.conf# █
```

Figure 2.12: Ping do Departamento A para S1

```
root@n7:/tmp/pycore.51525/n7.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_req=1 ttl=62 time=0.830 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=62 time=0.951 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=62 time=0.823 ms
^C
--- 10.0.3.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.823/0.868/0.951/0.058 ms
root@n7:/tmp/pycore.51525/n7.conf# █
```

Figure 2.13: Ping do Departamento B para S1

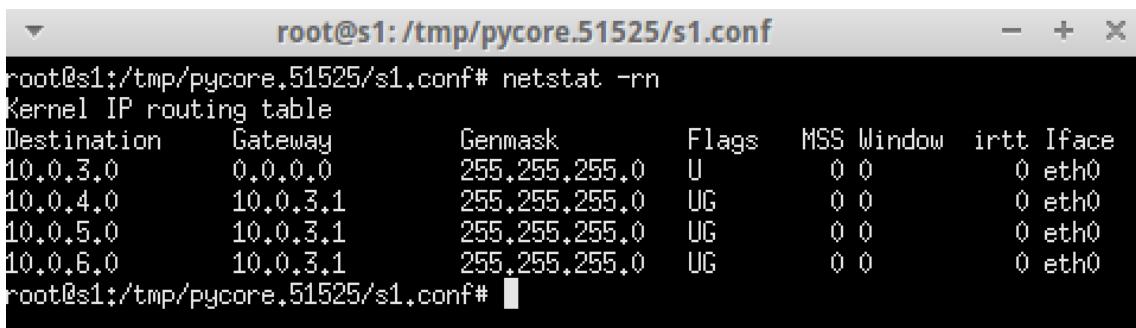


```

root@Rext:/tmp/pycore.51525/Rext.conf
root@Rext:/tmp/pycore.51525/Rext.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data.
64 bytes from 10.0.3.10: icmp_req=1 ttl=63 time=0.378 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=63 time=0.174 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=63 time=0.215 ms
^C
--- 10.0.3.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.174/0.255/0.378/0.090 ms
root@Rext:/tmp/pycore.51525/Rext.conf#

```

Figure 2.14: Ping do Rext para S1



```

root@s1:/tmp/pycore.51525/s1.conf
root@s1:/tmp/pycore.51525/s1.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
10.0.3.0        0.0.0.0       255.255.255.0 U        0 0          0 eth0
10.0.4.0        10.0.3.1       255.255.255.0 UG       0 0          0 eth0
10.0.5.0        10.0.3.1       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 eth0
root@s1:/tmp/pycore.51525/s1.conf#

```

Figure 2.15: Tabela de Encaminhamento do Servidor

Como podemos ver pelas imagens em cima a conectividade foi restabelecida. Na Figura 2.12 vemos que um laptop do Departamento A consegue transmitir pacotes entre ele e o Servidor, e este recebe-o. Por isso temos a certeza que a conectividade foi restaurada.

Pelas mesmos razões que o tópico anterior e pelas figuras 2.13 e 2.14 verificamos que a conectividade entre o Departamento B e o Servidor, e o Router de Acesso (Rext) e o servidor, respetivamente, também foram restauradas.

2.3 Pergunta 3

- A Considere que dispões 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 dadas.**

O nosso ip da rede é dado por 172.48.48.0/20. Uma vez que, apenas temos uma máscara de 20 significa que a nossa rede pode usar todos os endereços entre 172.48.48.0 e 172.48.63.255 . Como temos 3 departamentos, inicialmente, iríamos precisar de 2 bits para conseguir fazer sub-netting, no entanto como a possibilidade 00 e 11 estão reservadas ficaríamos com apenas 2 opções, 00 e 01 para representar os 3 departamentos o que torna a situação impossível. Aumentamos assim de 2 bits para 3 bits para representar as 3 redes.

172.48.0011 |XXX|0.0

000	Reservada	
001	Livre	Dep. A
010	Livre	
011	Livre	Dep. B
100	Livre	
101	Livre	Dep. C
110	Livre	
111	Reservada	

Usando 3 bits, temos 8 opções possíveis para endereços para as sub-redes, sendo que tal como acontece para 2 bit, as opções 000 e 111 ficam reservadas e ficamos assim reduzidos a 6 opções. Atribuímos, assim, uma das opções a cada departamento. Durante o processo de atribuição, deixamos sempre um de endereço disponível entre duas opções para que, no futuro, caso seja preciso aumentar o número de hosts de um departamento, seja possível manter uma coerência entre o departamento e os ips de modo a que à medida que os ips aumentam a ordem alfabética dos departamentos também o faça.

Dep.	IP	IP-Início	IP-Fim
A	172.48.50.0/23	172.48.50.0	172.48.51.255
B	172.48.54.0/23	172.48.54.0	172.48.55.255
C	172.48.58.0/23	172.48.58.0	172.48.59.255

Table 2.1: IP's de host para cada departamento

Com a ajuda da tabela 2.1, atribuímos um ip a cada interface dentro dos intervalos que cada departamento tem disponíveis para si sem ser os ip's com tudo zeros ou tudo uns.

Dep. A	IP atribuído	Dep. B	IP atribuído	Dep. C	IP atribuído
n4	172.48.50.2/23	n7	172.48.54.2/23	n10	172.48.58.5/23
n5	172.48.50.3/23	n8	172.48.54.3/23	n11	172.48.58.4/23
n6	172.48.50.4/23	n9	172.48.54.4/23	n12	172.48.58.3/23
Ra	172.48.50.1/23	Rb	172.48.54.1/23	S1	172.48.58.2/23
-	-	-	-	Rc	172.48.58.1/23

Table 2.2: Endereços atribuídos a cada dispositivo

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

Uma vez que reservamos 3 bits para fazer sub-netting a nossa máscara passa de 20 para 23 ficando o seu valor decimal em 255.255.254.0 . Como a máscara usa 23 bits ficamos com 9 bits em que podemos mexer. O número de host é então dado por $2^9 - 1$. Como cada rede guarda 2 endereço para broadcast e outro para comunicar com todos os dispositivos, o número de hosts é reduzido em 2 ficando em 509.

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

Na tabela 2.2 mostra o endereço atribuído a cada interface. Nesta fase alteramos os valores dos ip's dos dispositivos para os valores atribuídos na tabela 2.2. Para garantir a conectividade usamos o comando ping de n4 para um laptop de cada dispositivo como mostra a figura 2.17

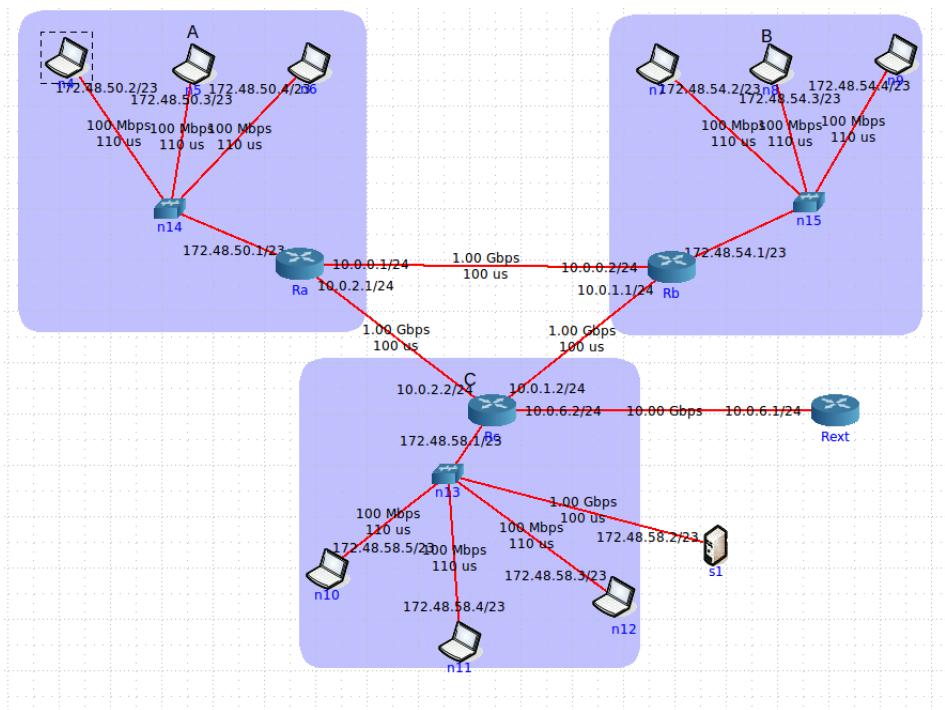


Figure 2.16: Equipamentos e Departamentos com novos IP's

```

root@n4:/tmp/pycore.59994/n4.conf
root@n4:/tmp/pycore.59994/n4.conf# ping 172.48.50.4
PING 172.48.50.4 (172.48.50.4) 56(84) bytes of data.
64 bytes from 172.48.50.4: icmp_req=1 ttl=64 time=0.732 ms
64 bytes from 172.48.50.4: icmp_req=2 ttl=64 time=0.444 ms
64 bytes from 172.48.50.4: icmp_req=3 ttl=64 time=0.362 ms
^C
--- 172.48.50.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.362/0.512/0.732/0.160 ms
root@n4:/tmp/pycore.59994/n4.conf# ping 172.48.54.2
PING 172.48.54.2 (172.48.54.2) 56(84) bytes of data.
64 bytes from 172.48.54.2: icmp_req=1 ttl=62 time=2.07 ms
64 bytes from 172.48.54.2: icmp_req=2 ttl=62 time=0.585 ms
64 bytes from 172.48.54.2: icmp_req=3 ttl=62 time=0.645 ms
^C
--- 172.48.54.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.585/1.102/2.078/0.591 ms
root@n4:/tmp/pycore.59994/n4.conf# ping 172.48.58.2
PING 172.48.58.2 (172.48.58.2) 56(84) bytes of data.
64 bytes from 172.48.58.2: icmp_req=1 ttl=62 time=1.23 ms
64 bytes from 172.48.58.2: icmp_req=2 ttl=62 time=0.917 ms
64 bytes from 172.48.58.2: icmp_req=3 ttl=62 time=0.733 ms
^C
--- 172.48.58.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.733/0.960/1.231/0.207 ms
root@n4:/tmp/pycore.59994/n4.conf# ping 10.0.6.1
PING 10.0.6.1 (10.0.6.1) 56(84) bytes of data.
64 bytes from 10.0.6.1: icmp_req=1 ttl=62 time=0.640 ms
64 bytes from 10.0.6.1: icmp_req=2 ttl=62 time=0.515 ms
64 bytes from 10.0.6.1: icmp_req=3 ttl=62 time=0.518 ms
^C
--- 10.0.6.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.515/0.557/0.640/0.064 ms
root@n4:/tmp/pycore.59994/n4.conf#

```

Figure 2.17: Ping para as diversas redes

Chapter 3

Conclusão

Na primeira parte deste trabalho fizemos uma análise detalhada do protocolo IPv4. Para esta realização foi utilizada a topologia Core na virtualbox, de modo a observar os datagramas e como é realizado o tráfego de ICMP. No envio de pacotes de dados, foi analisada se era necessária a sua fragmentação de modo a possibilitar a sua transmissão. Com isto podemos observar e perceber melhor como é que o envio e transmissão de dados é feito entre diferentes máquinas ligadas à mesma rede.

Na segunda parte deste guião foca-mo-nos mais em perceber como funciona o processo de endereçamento e encaminhamento IP. Para isto construimos um diagrama com vários departamentos e os seus diversos equipamentos. Conseguimos assim analisar se estes podiam conectar-se entre si, a forma como o faziam (com as suas rotas, com o seu tipo de encaminhamento- dinâmica ou estático) e formas de conectar ou desconectar os vários equipamentos. Numa última fase foi nos possível pôr à prova o nosso conhecimento, sendo necessário atribuir endereços a cada equipamento, diferenciando o respetivo departamento e possibilitando a conectividade.