



TP4: Protocolo IP

João Alves, A91646

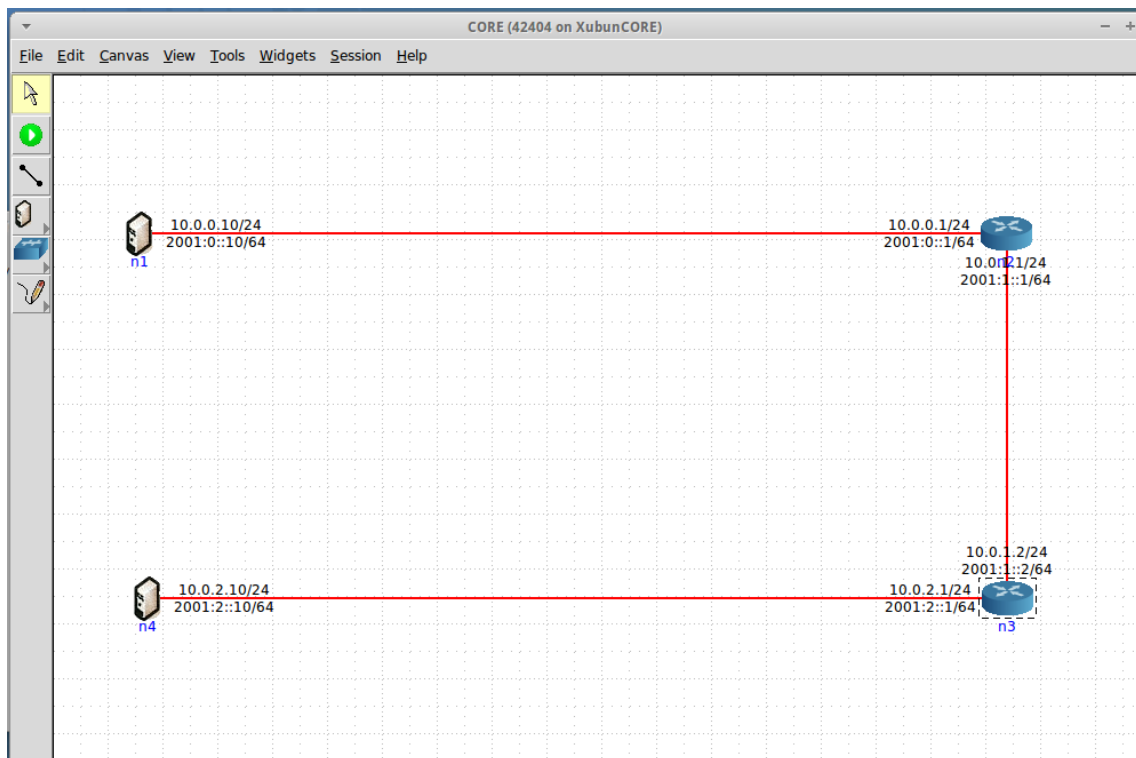
Ivo Lima, A90214

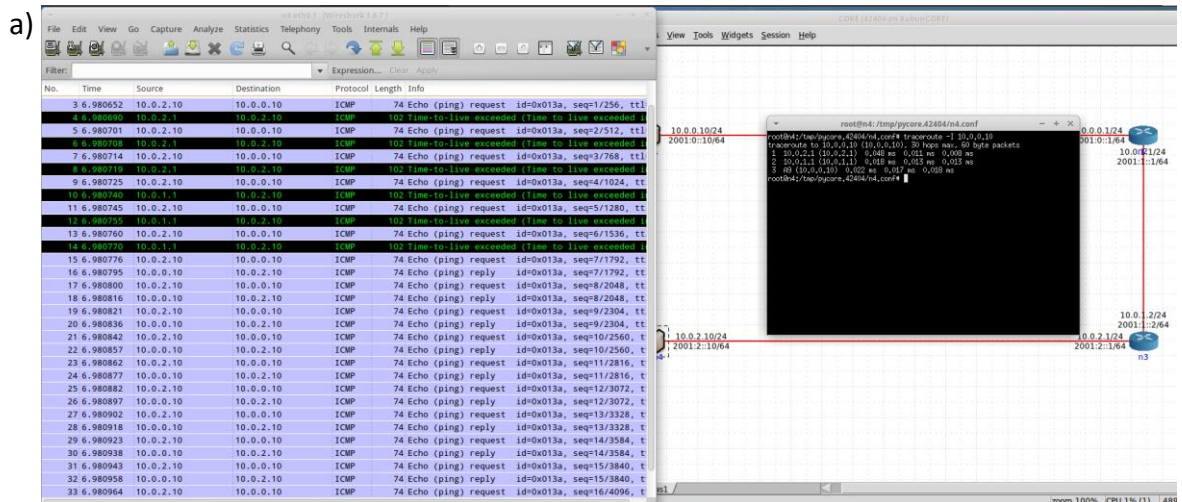
Rúben Machado, A91656

PL2 Grupo 5

Parte 1

1.





b) Para cada *tracert* existem 3 *requests* e 3 *replies*, sendo este o resultado esperado, contudo 6 *requests* não obtiveram resposta porque as portas não existem.

3	6.980652	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=1/256, ttl=64
4	6.980690	10.0.2.1	10.0.2.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
5	6.980701	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=2/512, ttl=64
6	6.980708	10.0.2.1	10.0.2.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
7	6.980714	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=3/768, ttl=64
8	6.980719	10.0.2.1	10.0.2.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
9	6.980725	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=4/1024, ttl=64
10	6.980740	10.0.1.1	10.0.2.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
11	6.980745	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=5/1280, ttl=64
12	6.980755	10.0.1.1	10.0.2.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
13	6.980760	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=6/1536, ttl=64
14	6.980770	10.0.1.1	10.0.2.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
15	6.980776	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=7/1792, ttl=64
16	6.980795	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=7/1792, ttl=64
17	6.980800	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=8/2048, ttl=64
18	6.980816	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=8/2048, ttl=64
19	6.980821	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=9/2304, ttl=64
20	6.980836	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=9/2304, ttl=64
21	6.980842	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=10/2560, ttl=64
22	6.980857	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=10/2560, ttl=64
23	6.980862	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=11/2816, ttl=64
24	6.980877	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=11/2816, ttl=64
25	6.980882	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=12/3072, ttl=64
26	6.980897	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=12/3072, ttl=64
27	6.980902	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=13/3328, ttl=64
28	6.980918	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=13/3328, ttl=64
29	6.980923	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=14/3584, ttl=64
30	6.980938	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=14/3584, ttl=64
31	6.980943	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=15/3840, ttl=64
32	6.980958	10.0.0.10	10.0.2.10	ICMP	74	Echo (ping) reply id=0x013a, seq=15/3840, ttl=64
33	6.980964	10.0.2.10	10.0.0.10	ICMP	74	Echo (ping) request id=0x013a, seq=16/4096, ttl=64

c) O valor inicial mínimo deverá ser 3, uma vez que o TTL é decrementado aquando a passagem em cada router.

d) O tempo médio de ida-e-volta é de 0.056 ms.

$$(0,048 + 0,011 + 0,008 + 0,018 + 0,013 + 0,013 + 0,022 + 0,017 + 0,018) \div 3 = 0,056$$

2. Com o auxílio *PingPlotter* 5 verificamos que o tamanho do pacote por defeito é 56 bytes.

Packet Type: ICMP Using Windows DLL (default)

Time to Wait for ping replies: 3000

Packet send delay: Auto ms

Packet size: 56 bytes

Network Interface: Default - determine best

☒ Allow packet fragmentation?

☐ Start traces as final hop only

A imagem em baixo corresponde à primeira mensagem ICMP capturada.

391	132.89...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=103/26368, ttl=255 (reply in 392)
392	132.89...	193.136.9.240	172.26.1.65	ICMP	70 Echo (ping) reply	id=0x0001, seq=103/26368, ttl=61 (request in 391)
393	132.94...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=104/26624, ttl=1 (no response found!)
394	132.94...	172.26.254.254	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
395	132.99...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=105/26880, ttl=2 (no response found!)
396	132.99...	172.26.1.65	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
397	133.04...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=106/27136, ttl=3 (no response found!)
398	133.04...	172.26.115.252	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
399	133.09...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=107/27392, ttl=4 (reply in 400)
400	133.09...	193.136.9.240	172.26.1.65	ICMP	70 Echo (ping) reply	id=0x0001, seq=107/27392, ttl=61 (request in 399)
401	135.39...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=108/27648, ttl=255 (reply in 402)
402	135.39...	193.136.9.240	172.26.1.65	ICMP	70 Echo (ping) reply	id=0x0001, seq=108/27648, ttl=61 (request in 401)

a) O IP da interface ativa é 172.26.1.65

Source Address: 172.26.1.65															
Destination Address: 193.136.9.240															
Internet Control Message Protocol															
00 d0 03 ff 94 00 3c 6a a7 07 88 a2 08 00 45 00<jE..															
00 38 fe 0c 00 00 01 01 42 e5 ac 1a 01 41 c1 88 -8.....B....A..															
09 f0 08 00 35 d5 00 01 00 68 20 20 20 20 20 205...-h															
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20															
20 20 20 20 20 20															

b) O valor do campo protocolo é 1, e identifica o ICMP. Neste caso, o ICMP (*Internet Control Message Protocol*) é usado para testar a conectividade entre a origem e o destino.

Fragment Offset: 0															
Time to Live: 1															
> [Expert Info (Note/Sequence): "Time To Live" only 1]															
Protocol: ICMP (1)															
Header Checksum: 0x42e5 [validation disabled]															
[Header checksum status: Unverified]															
Source Address: 172.26.1.65															
Destination Address: 193.136.9.240															
Internet Control Message Protocol															
000 00 d0 03 ff 94 00 3c 6a a7 07 88 a2 08 00 45 00<jE..															
010 00 38 fe 0c 00 00 01 01 42 e5 ac 1a 01 41 c1 88 -8.....B....A..															
020 09 f0 08 00 35 d5 00 01 00 68 20 20 20 20 20 205...-h															
030 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20															
040 20 20 20 20 20 20															

c) O cabeçalho tem 20 bytes. O campo de dados tem 28 bytes. O tamanho do *payload* calcula-se através do comprimento TCP.

[illegible]

d) Não é fragmentado.

```
> Flags: 0x00
  Fragment Offset: 0
> Time to Live: 1
  Protocol: ICMP (1)
  Header Checksum: 0x42e5 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 172.26.1.65
```

```
0000  00 d0 03 ff 94 00 3c 6a a7 07 88 a2 08 00 45 00  ....<j.....E.
0010  00 38 fe 0c 00 01 42 e5 ac 1a 01 41 c1 88  -8...B...A..
0020  09 f0 08 00 35 d5 00 01 00 68 20 20 20 20 20  ....5...-h
0030  20 20 20 20 20 20 20 20 20 20 20 20 20 20
0040  20 20 20 20 20 20
```

e) A minha máquina com o IP 172.26.1.65 tenta comunicar com o router 193.136.9.240, porém para conseguir comunicar com este necessita de 3 chamadas *TTL*, o que lhe permitem identificar o caminho desde a origem (máquina) até ao destino (router). O padrão em questão consiste em enviar o máximo de *TTLs* possíveis para tentar chegar ao destino (255) e o mesmo ocorre no sentido inverso com o envio de 61 *TTLs*. Depois de existir esta confirmação de que há comunicação possível, a máquina tenta chegar ao destino com o menor número de *TTLs* possível, que neste caso são 4. E este padrão repete-se ao longo do tempo.

393	132.94...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=104/26624, ttl=1 (no response found!)
394	132.94...	172.26.254.254	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
395	132.99...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=105/26880, ttl=2 (no response found!)
396	132.99...	172.16.2.1	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
397	133.04...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=106/27136, ttl=3 (no response found!)
398	133.04...	172.16.115.252	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
399	133.09...	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=107/27392, ttl=4 (reply in 400)
400	133.09...	193.136.9.240	172.26.1.65	ICMP	70 Echo (ping) reply	id=0x0001, seq=107/27392, ttl=61 (request in 399)

- f) O campo TTL vai sendo alterado de 255 para 253, ou seja, decrementa. Isto acontece, pois na comunicação entre a minha máquina (172.26.1.65) e o router (193.136.9.240) existem mais 3 dispositivos (172.26.254.254, 172.16.2.1, 172.16.115.252). Sendo que se, por exemplo, o pacote chegasse ao destino com valor TTL 1, então o router deveria descartá-lo tornando o valor TTL 0 e enviaria de volta um ICMP de *type* 11 e de código 0 (representa o *TTL Exceeded*). No nosso caso o TTL é iniciado com valor 255 que sendo decrementado em 1 conforme passa pelos diferentes dispositivos. Uma vez que eles são 2 o valor será decrementado 2 vezes (porque começa no 0).

```
394 132.94... 172.26.254.254 172.26.1.65 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
```

Internet Protocol Version 4, Src: 172.26.254.254, Dst: 172.26.1.65

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)

Total Length: 56

Identification: 0x5da3 (23971)

> Flags: 0x00

Fragment Offset: 0

Time to Live: 255

Protocol: ICMP (1)

Header Checksum: 0x04ed [validation disabled]

[Header checksum status: Unverified]

0000	3c 6a a7 07 88 a2 00 d0 03 ff 94 00 08 00 45 c0	<j.....E.
0010	00 38 5d a3 00 00 ff 01 04 ed ac 1a fe fe ac 1a	.8]... ..
0020	01 41 0b 00 b6 c1 00 00 00 00 45 00 00 38 fe 0c	.A.....E..8..
0030	00 00 01 01 42 e5 ac 1a 01 41 c1 88 09 f0 08 00	...B...-A.....
0040	35 d5 00 01 00 68	5....h

```
396 132.99... 172.16.2.1 172.26.1.65 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
```

Internet Protocol Version 4, Src: 172.16.2.1, Dst: 172.26.1.65

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 56

Identification: 0xb231 (45617)

> Flags: 0x00

Fragment Offset: 0

Time to Live: 254

Protocol: ICMP (1)

Header Checksum: 0xaf26 [validation disabled]

[Header checksum status: Unverified]

0000	3c 6a a7 07 88 a2 00 d0 03 ff 94 00 08 00 45 00	<j.....E.
0010	00 38 b2 31 00 00 fe 01 af 26 ac 10 02 01 ac 1a	.8.1... ..&.....
0020	01 41 0b 00 b6 c1 00 00 00 00 45 00 00 38 fe 0d	.A.....E..8..
0030	00 00 01 01 42 e4 ac 1a 01 41 c1 88 09 f0 08 00	...B...-A.....
0040	35 d4 00 01 00 69	5....i


```

Internet Protocol Version 4, Src: 172.16.115.252, Dst: 172.26.1.65
  0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x34b8 (13496)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 253
    Protocol: ICMP (1)
    Header Checksum: 0xbba4 [validation disabled]
    [Header checksum status: Unverified]
0000  3c 6a a7 07 88 a2 00 d0 03 ff 94 00 08 00 45 00  <j.....E.
0010  00 38 34 b8 00 00 fd 01 bb a4 ac 10 73 fc ac 1a  -84...s...
0020  01 41 0b 00 00 b6 c1 00 00 00 45 00 00 38 fe 0e  -A.....E-8-
0030  00 00 01 01 42 e3 ac 1a 01 41 c1 88 09 f0 08 00  .....B...A...
0040  35 d3 00 01 00 6a                                5...j

```

391	132.890151	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=103/26368, ttl=255 (reply in 392)
392	132.893663	193.136.9.240	172.26.1.65	ICMP	70 Echo (ping) reply	id=0x0001, seq=103/26368, ttl=61 (request in 391)
393	132.940171	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=104/26624, ttl=1 (no response found!)
394	132.945382	172.26.254.254	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
395	132.990566	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=105/26880, ttl=2 (no response found!)
396	132.993150	172.26.1.65	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
397	133.041629	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=106/27136, ttl=3 (no response found!)
398	133.043595	172.26.115.252	172.26.1.65	ICMP	70 Time-to-live exceeded	(Time to live exceeded in transit)
399	133.092819	172.26.1.65	193.136.9.240	ICMP	70 Echo (ping) request	id=0x0001, seq=107/27392, ttl=4 (reply in 400)
400	133.093759	193.136.9.240	172.26.1.65	ICMP	70 Echo (ping) reply	id=0x0001, seq=107/27392, ttl=61 (request in 399)

General

Auto-Save Data

CloudConnect

Web Server

Default Settings

Display

Engine

Packet Type

ICMP Using Windows DLL (default)

Time to Wait for ping replies

3000

Packet send delay

Auto

ms

Packet size

3025

bytes

Network Interface

Default - determine best

☒ Allow packet fragmentation?

☐ Start traces as final hop only

a) Na imagem em baixo está a primeira imagem ICMP e está fragmentada. Uma vez que o pacote corresponde a um *jumbo frame* pois tem mais do que 1500 bytes, o router para assegurar a transmissão deste determina qual é o maior tamanho de informação permitido pela rede, sem que perca o pacote ou a conexão à rede, e isto levou à necessidade de fragmentação da trama.

```

328 0.910081 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3826/61966, ttl=255 (reply in 328)
347 0.927233 193.136.9.240 192.168.1.92 ICMP 79 Echo (ping) reply id=0x0001, seq=3826/61966, ttl=51 (request in 328)
367 0.960272 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3827/62222, ttl=1 (no response found!)
370 0.962819 192.168.1.254 192.168.1.92 ICMP 82 Time-to-live exceeded (Time to live exceeded in transit)
393 1.010084 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3828/62478, ttl=2 (no response found!)
426 1.061774 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3829/62734, ttl=3 (no response found!)
458 1.112296 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3830/62990, ttl=4 (no response found!)
467 1.116500 195.8.30.245 192.168.1.92 ICMP 110 Time-to-live exceeded (Time to live exceeded in transit)
489 1.163604 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3831/63246, ttl=5 (no response found!)
494 1.172201 195.8.1.57 192.168.1.92 ICMP 110 Time-to-live exceeded (Time to live exceeded in transit)
518 1.213608 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3832/63502, ttl=6 (no response found!)
529 1.222764 193.136.251.1 192.168.1.92 ICMP 110 Time-to-live exceeded (Time to live exceeded in transit)
556 1.263707 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3833/63758, ttl=7 (no response found!)
560 1.272526 194.210.6.102 192.168.1.92 ICMP 110 Time-to-live exceeded (Time to live exceeded in transit)
589 1.313712 192.168.1.92 193.136.9.240 ICMP 79 Echo (ping) request id=0x0001, seq=3834/64014, ttl=8 (no response found!)

```

```

0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 65
  Identification: 0x18db (6363)
  Flags: 0x01
    Fragment Offset: 2960
    Time to Live: 255
    Protocol: ICMP (1)
    Header Checksum: 0x13f2 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.92

```

```

0000 e0 b9 e5 e6 f8 fe 3c 6a a7 07 88 a2 08 00 45 00 ...n.cj .....E
0010 00 41 18 db 01 72 ff 01 13 f2 0a 08 01 5c c1 88 ..A. [01] .....V
0020 09 f0 20 20 20 20 20 20 20 20 20 20 20 20 20 ..
0030 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0040 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20

```

- b) O facto de que o *Wireshark* estar configurado para sinalizar mais do que um segmento mostra que o datagrama foi fragmentado. Uma vez que o *Fragment Offset* é 0 este é o primeiro fragmento, pois o último tem valor de 1480. Como podemos ver embaixo o comprimento total do datagrama é 1500.

326 0.910081	192.168.1.92	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=18db) [Reassembled in #328]
327 0.910081	192.168.1.92	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=18db) [Reassembled in #328]
328 0.910081	192.168.1.92	193.136.9.240	ICMP	79	Echo (ping) request id=0x0001, seq=3826/61966, ttl=255 (reply in 347)
329 0.912268	192.168.1.92	213.163.86.202	UDP	90	65022 → 50002 Len=48
330 0.912476	162.159.130.235	192.168.1.92	TLSv1...	111	Application Data
331 0.914950	192.168.1.92	213.163.87.7	UDP	1177	65021 → 50027 Len=1135
332 0.914989	192.168.1.92	213.163.87.7	UDP	1177	65021 → 50027 Len=1135
333 0.915010	192.168.1.92	213.163.87.7	UDP	1177	65021 → 50027 Len=1135
334 0.915028	192.168.1.92	213.163.87.7	UDP	1177	65021 → 50027 Len=1135
335 0.915046	192.168.1.92	213.163.87.7	UDP	1177	65021 → 50027 Len=1135
336 0.918156	213.163.86.202	192.168.1.92	UDP	1090	50002 → 65022 Len=1048
337 0.919958	192.168.1.92	213.163.87.7	UDP	1177	65021 → 50027 Len=1135
338 0.919989	192.168.1.92	213.163.87.7	UDP	1177	65021 → 50027 Len=1135
339 0.920034	213.163.86.202	192.168.1.92	UDP	1090	50002 → 65022 Len=1048

> Frame 326: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{462EEDEE-6D83-4B18-A701-8B6D4719F7B4}, id 0
> Ethernet II, Src: IntelCor_07:88:a2 (3c:6a:a7:07:88:a2), Dst: Technico_6e:f8:fe (e0:b9:e5:6e:f8:fe)
> Internet Protocol Version 4, Src: 192.168.1.92, 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: 0x18db (6363)
> Flags: 0x20, More fragments
Fragment Offset: 0
Time to Live: 255
Protocol: ICMP (1)
Header Checksum: 0xefc8 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.92
Destination Address: 193.136.9.240
[Reassembled IPv4 in frame: 328]
> Data (1480 bytes)

- c) Trata-se do segundo fragmento uma vez que o deslocamento do *frame* é de 1480 e existem mais fragmentos, como se pode ver embaixo nas *flags* com “*More Fragments*”.

345 0.926276	192.168.1.92	213.163.87.7	UDP	85	65021 → 50027 Len=43
346 0.927233	193.136.9.240	192.168.1.92	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b718) [Reassembled in #347]
347 0.927233	193.136.9.240	192.168.1.92	ICMP	79	Echo (ping) reply id=0x0001, seq=3826/61966, ttl=51 (request in 328)
348 0.942560	192.168.1.92	213.163.87.7	UDP	1197	65021 → 50027 Len=1155
349 0.942652	192.168.1.92	213.163.87.7	UDP	1197	65021 → 50027 Len=1155
350 0.942676	192.168.1.92	213.163.87.7	UDP	1197	65021 → 50027 Len=1155
351 0.942696	192.168.1.92	213.163.87.7	UDP	1197	65021 → 50027 Len=1155
352 0.942717	192.168.1.92	213.163.87.7	UDP	1197	65021 → 50027 Len=1155
353 0.944931	213.163.86.202	192.168.1.92	UDP	85	50002 → 65022 Len=43
354 0.948214	192.168.1.92	213.163.87.7	UDP	85	65021 → 50027 Len=43
355 0.948267	192.168.1.92	213.163.87.7	UDP	1197	65021 → 50027 Len=1155
356 0.948287	192.168.1.92	213.163.87.7	UDP	1197	65021 → 50027 Len=1155
357 0.948305	192.168.1.92	213.163.87.7	UDP	1198	65021 → 50027 Len=1156
358 0.948328	192.168.1.92	213.163.87.7	UDP	1198	65021 → 50027 Len=1156

> Frame 346: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{462EEDEE-6D83-4B18-A701-8B6D4719F7B4}, id 0
> Ethernet II, Src: Technico_6e:f8:fe (e0:b9:e5:6e:f8:fe), Dst: IntelCor_07:88:a2 (3c:6a:a7:07:88:a2)
> Internet Protocol Version 4, Src: 193.136.9.240, Dst: 192.168.1.92
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x88 (DSCP: AF41, ECN: Not-ECT)
Total Length: 1500
Identification: 0xb718 (46872)
> Flags: 0x20, More fragments
Fragment Offset: 1480
Time to Live: 51
Protocol: ICMP (1)
Header Checksum: 0x1c4b [validation disabled]
[Header checksum status: Unverified]
Source Address: 193.136.9.240
Destination Address: 192.168.1.92
[Reassembled IPv4 in frame: 347]
> Data (1480 bytes)

- d) Ao mudar para 3025 bytes foram criados 3 fragmentos a partir do datagrama original, aliás podíamos verificar o valor no campo *Fragments count* que é uma informação fornecida pelo Wireshark mas não está presente no cabeçalho IP. Deteta-se que é o último fragmento pois no campo de *flags* apenas está o valor hexadecimal (0x01) e não o valor hexadecimal seguido de “*More Fragments*”.

795	1.615676	192.168.1.92	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=18e9) [Reassembled in #797]
796	1.615676	192.168.1.92	193.136.9.240	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=18e9) [Reassembled in #797]
797	1.615676	192.168.1.92	193.136.9.240	ICMP	79	Echo (ping) request id=0x0001, seq=3840/15, ttl=14 (reply in 813)
798	1.618682	192.168.1.92	213.163.87.7	UDP	1228	65021 → 50027 Len=1186
799	1.618743	192.168.1.92	213.163.87.7	UDP	1228	65021 → 50027 Len=1186
800	1.618766	192.168.1.92	213.163.87.7	UDP	1228	65021 → 50027 Len=1186
801	1.620610	213.163.86.202	192.168.1.92	UDP	1089	50002 → 65022 Len=1047
802	1.622511	213.163.86.202	192.168.1.92	UDP	1090	50002 → 65022 Len=1048
803	1.622511	213.163.86.202	192.168.1.92	UDP	1090	50002 → 65022 Len=1048
804	1.622511	213.163.86.202	192.168.1.92	UDP	1090	50002 → 65022 Len=1048
805	1.623561	192.168.1.92	213.163.86.202	UDP	94	65022 → 50002 Len=52


```

> Frame 797: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface \Device\NPF_{462EEDE-6D83-4B18-A701-8B6D4719F7B4}, id 0
> Ethernet II, Src: IntelCor_07:88:a2 (3c:6a:a7:07:88:a2), Dst: Technico_6e:f8:fe (e0:b9:e5:6e:f8:fe)
> Internet Protocol Version 4, Src: 192.168.1.92, 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: 65
  Identification: 0x18e9 (6377)
  > Flags: 0x01
    Fragment Offset: 2960
    Time to Live: 14
    Protocol: ICMP (1)
    Header Checksum: 0x04e5 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.92
    Destination Address: 193.136.9.240
  > [3 IPv4 Fragments (3005 bytes): #795(1480), #796(1480), #797(45)]
> Internet Control Message Protocol

```



```

0010  00 41 18 e9 01 72 0e 01 04 e5 c0 a8 01 5c c1 88  .A..F...
0020  09 f0 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ..
0030  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20

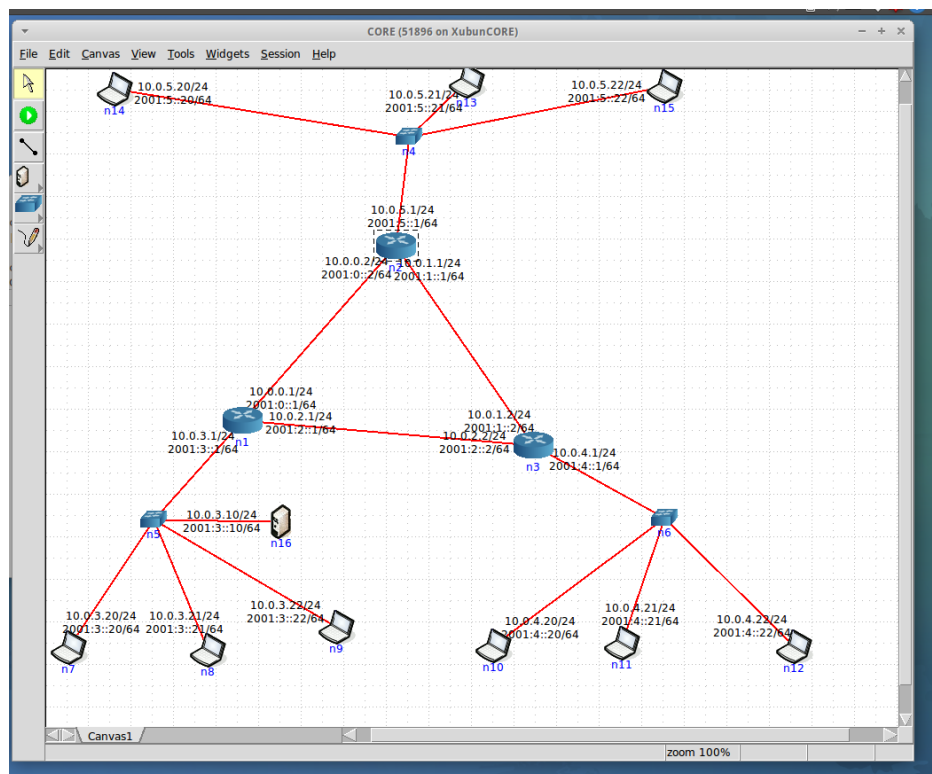
```

- e) Comparando os dois primeiros fragmentos podemos notar que as principais diferenças estão nos campos *Fragment offset* e no *Header checksum*. Comparando o segundo fragmento com o último as principais diferenças estão nos campos *Total Length*, *More fragments*, *Fragment offset* e no *Header checksum*.
- O campo *Identification* é igual em todos os fragmentos pelo que no destino é possível saber quais dos fragmentos vão originar um datagrama. Logo o comprimento entre o primeiro e o segundo fragmentos são os mesmos e também têm sinalizadores de que existem mais fragmentos, enquanto que o último é o que tem menor comprimento e não tem nenhum sinalizador na *flag*. Depois de encontrarmos o primeiro fragmento que está sinalizado através do *Fragment Offset* 0, temos ainda a indicação das *flags* de que existem mais fragmentos. Uma vez que o comprimento do primeiro fragmento é 1500 bytes e analisando o Wireshark descobrimos que o segundo fragmento se encontra na posição 347 e esta além de ocupar o mesmo que a anterior ainda nos dá a indicação de que existe mais de 1 fragmento. Procurando esse fragmento acabamos por encontra-lo na posição 797 tendo o menor comprimento de apenas 65 bytes. Concluimos ainda que este é o último pois nas *flags* não existe indicação de que ainda há mais fragmentos, e deste modo com o auxílio destes e outros parâmetros que vão alternando conseguimos construir o datagrama original.

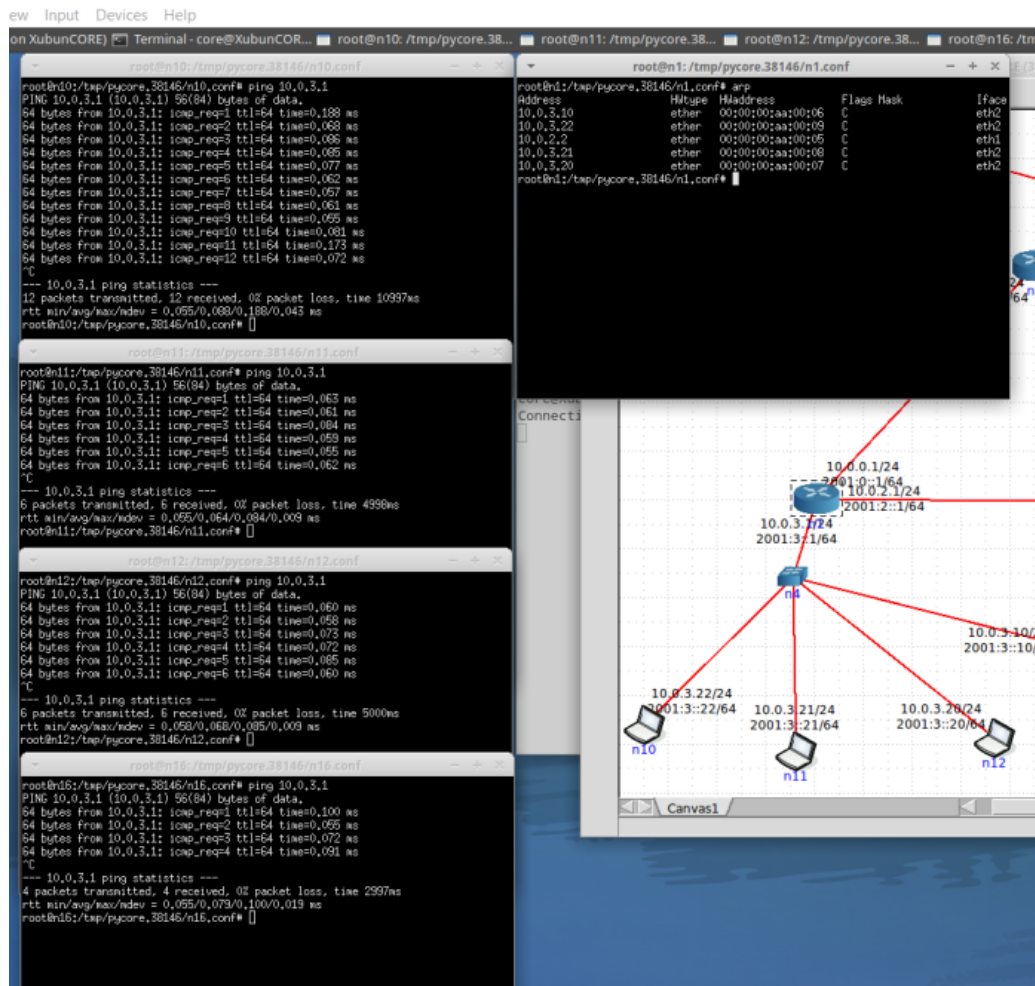
Parte 2

1)

a)



- b) Podemos concluir que são endereços privados pois, segundo o RFC1918, a *Internet Assigned Numbers Authority (IANA)* reservou o espaço de endereçamento 10.0.0.0 – 10.255.255.255 para conexões privadas. Ora como se pode verificar na imagem a cima todos os equipamentos estão endereçados dentro desse espaço.
- c) Uma vez que o *Switch* se trata de um equipamento que trabalha estritamente no nível 2 (a camada *link*) este não precisa de endereço IP uma vez que esse é um requisito do nível 3 (a camada de rede).
- d) Como podemos ver a na imagem em baixo existe conectividade entre os laptops dos utilizadores e o servidor do departamento A.



2)

- a) Na tabela de encaminhamento do laptop n10 temos unicamente como endereços de saída o router do departamento por onde os *packets* destinados a outros departamentos ou redes saem. Já no caso de quererem comunicar com os laptops no próprio departamento, os datagramas serão recebidos pela interface do endereço próprio que corresponde ao 0.0.0.0. Isto tudo é feito com a ajuda do *Ethernet Switch*.

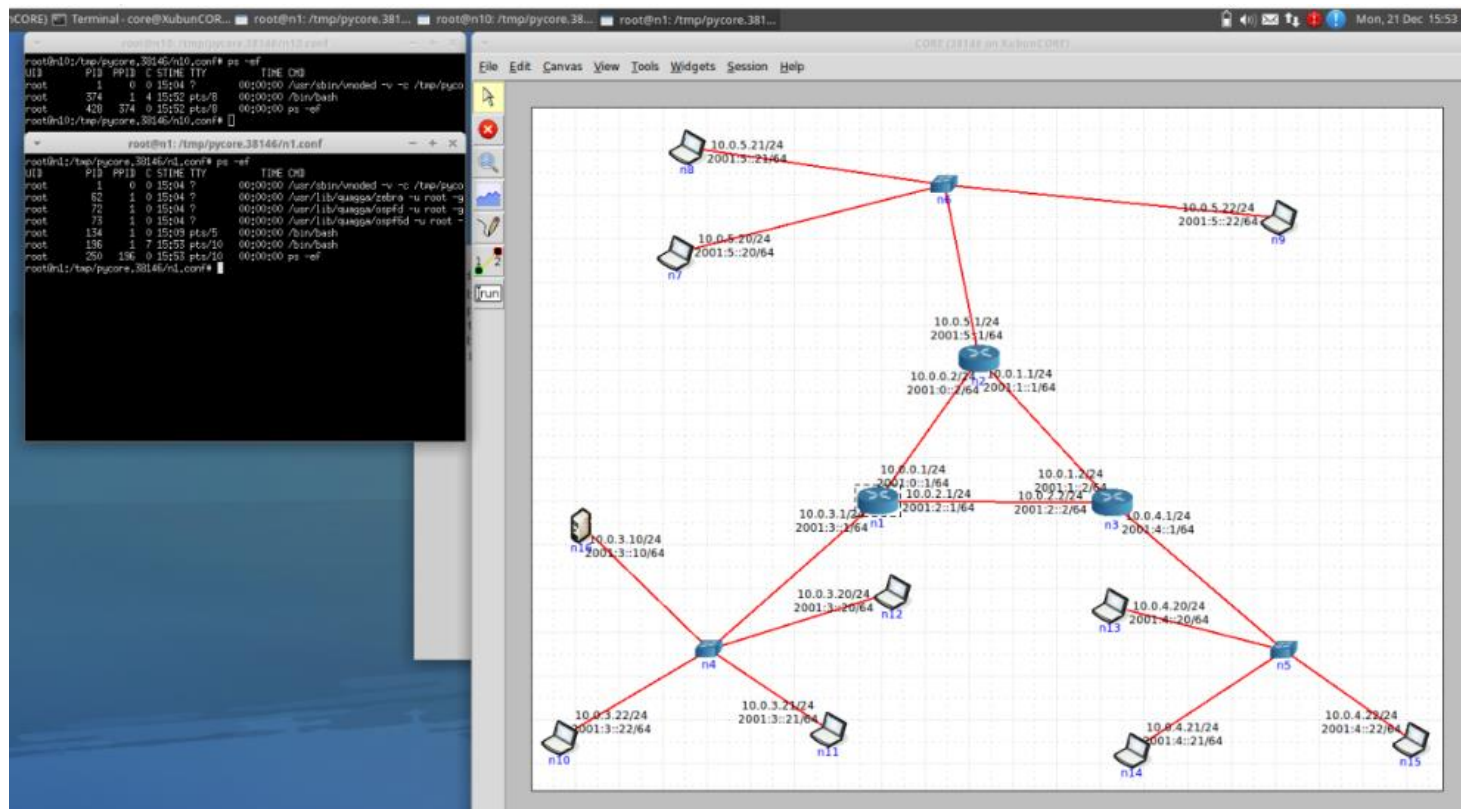
Já na tabela de encaminhamento do router n1 usamos saídas diferentes dependendo de para onde queremos enviar os datagramas. No caso de querer comunicar com qualquer um outro router usa-se a respetiva interface de endereço (por exemplo no caso do router cujo destino é 10.0.1.0 usa-se a interface do endereço correspondente ao *gateway* 10.0.0.2). E ainda caso queira comunicar com redes adjacentes usará a interface de endereço *default* (0.0.0.0).

```
Terminal - core@XubunCOR... root@n1: /tmp/pycore.38146/n1.conf root@n1: /tmp/pycore.38146/n1.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 0.0.0.0 255.255.255.0 U 0 0 0 eth2
10.0.4.0 10.0.2.2 255.255.255.0 UG 0 0 0 eth1
10.0.5.0 10.0.0.2 255.255.255.0 UG 0 0 0 eth0
root@n1: /tmp/pycore.38146/n1.conf#

root@n10: /tmp/pycore.38146/n10.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@n10: /tmp/pycore.38146/n10.conf#
```

- b) O sistema possui encaminhamento dinâmico e estático. Entre os Routers n1, n2 e n3 o encaminhamento é dinâmico, de tal forma que sempre que exista uma falha no link entre os routers existirá também uma adaptação a um novo caminho, pois as rotas são atualizadas ao longo do tempo.

Já em cada departamento o encaminhamento é realizado de forma estática, pois este é constituído por rotas pré-definidas, visto que nesta topologia de rede há maior conhecimento entre os intervenientes.

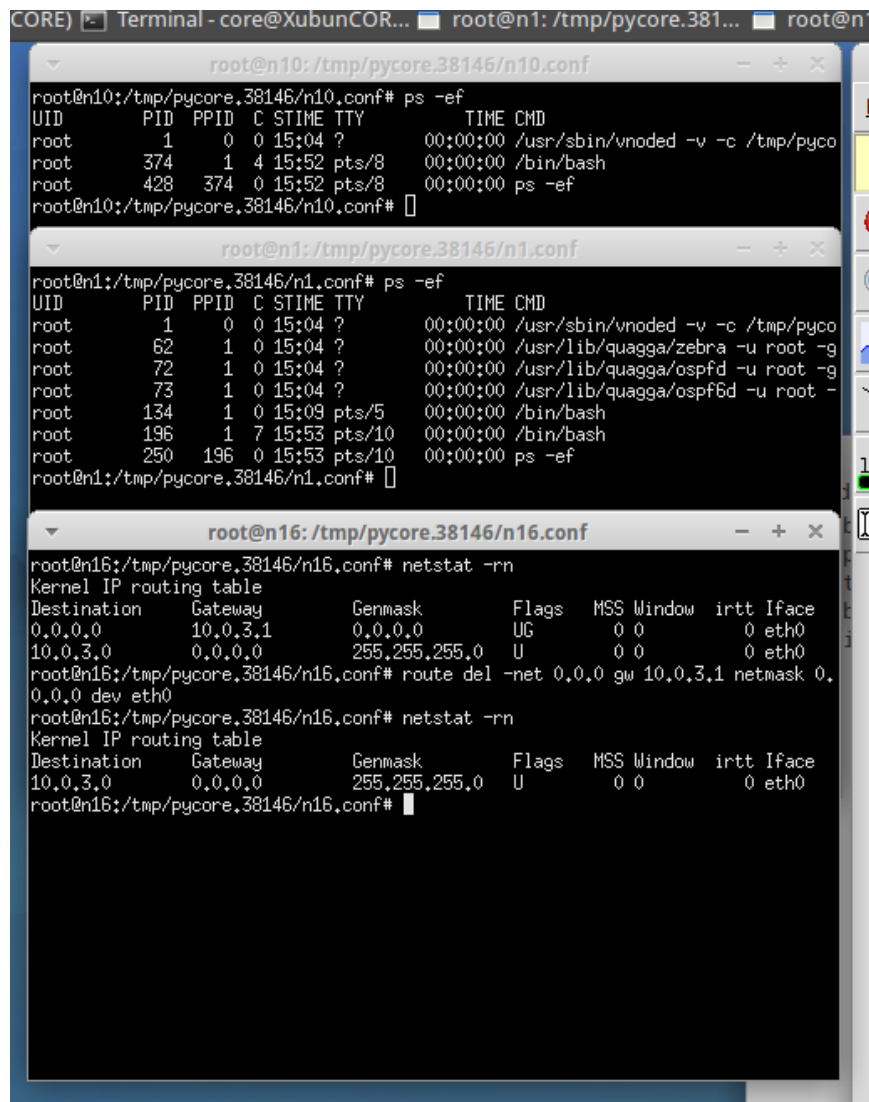


Podemos verificar no terminal de cima que apenas temos a decorrer na máquina 3 processos, sendo eles um processo pai, a *bash* e o comando *ps*.

Já no terminal de baixo 7 processos, sendo eles o processo pai, 2 *bash*, o comando *ps* e 3 *daemons*, dos quais 2 são protocolos de roteamento para IPv4 e IPv6.

Concluimos assim que nos routers existe encaminhamento dinâmico, e nos laptops encaminhamento estático.

- c) Como podemos ver em baixo, a aplicação do comando *router delete* no servidor n16 foi bem-sucedida.



The image displays three terminal windows from a Xubuntu system, showing network configuration changes across different servers.

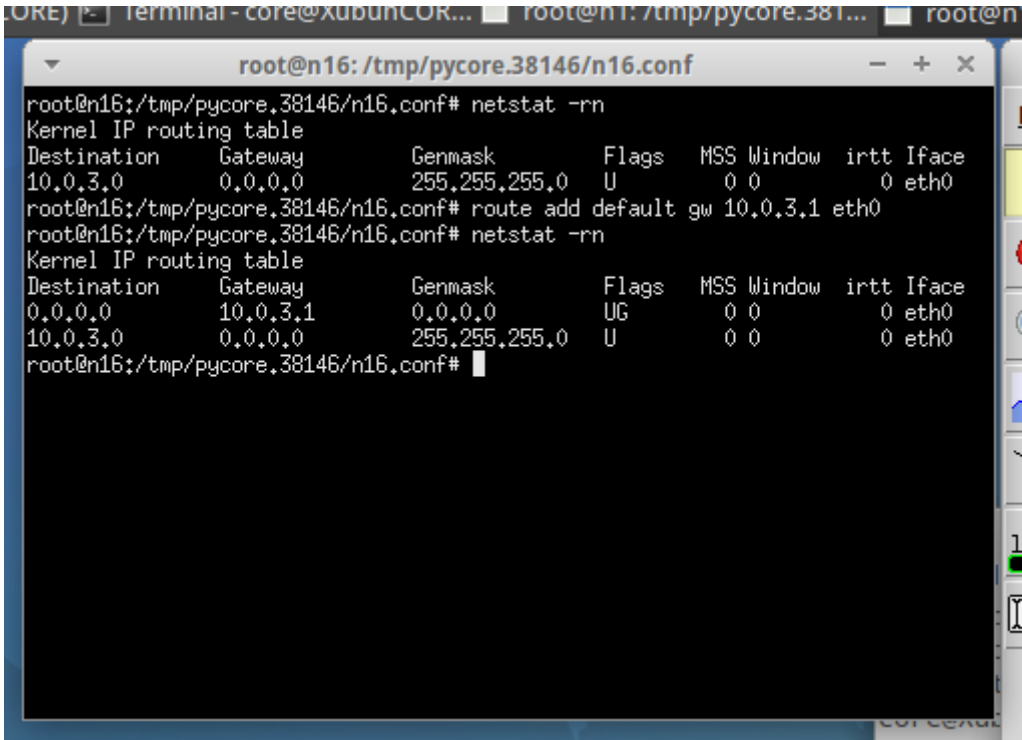
Terminal 1 (root@n10): Shows the output of the `ps -ef` command, listing processes running on server n10. The output shows three processes: `/usr/sbin/vnoded -v -c /tmp/pyco` (PID 1), `/bin/bash` (PID 374), and `ps -ef` (PID 428).

Terminal 2 (root@n1): Shows the output of the `ps -ef` command, listing processes running on server n1. The output shows several processes, including `/usr/lib/quagga/zebra -u root -g` (PID 62), `/usr/lib/quagga/ospfd -u root -g` (PID 72), `/usr/lib/quagga/ospf6d -u root -g` (PID 73), `/bin/bash` (PID 134), and `/bin/bash` (PID 196).

Terminal 3 (root@n16): Shows the output of the `netstat -rn` command, displaying the kernel IP routing table for server n16. The table shows the destination, gateway, and flags for various routes. The output shows that the route to the destination `10.0.3.0` via gateway `10.0.3.1` has been removed, as indicated by the `route del` command and the subsequent `netstat -rn` output.

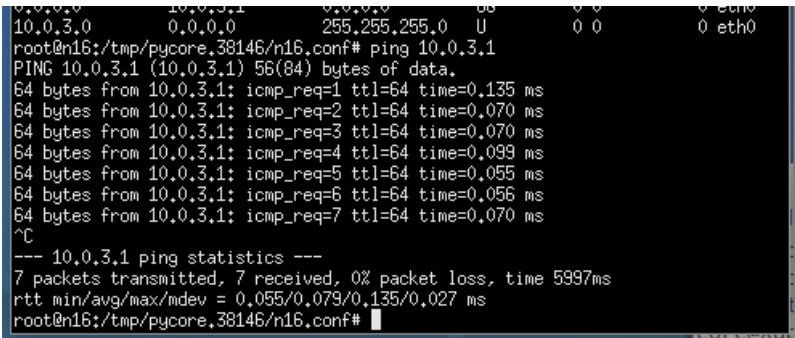
Ao tirar por defeito a rota do servidor n16, os utilizadores deixam de ter acesso, pois foi tirado o endereço de saída 10.0.3.1 que ligava o servidor ao *switch* que por sua vez se ligava ao respetivo router. Sempre que os utilizadores tentarem comunicar com o servidor enviarão os *packets* porém não irão receber resposta, pois já não têm como comunicar para fora do departamento C, logo não há rota definida e por isso a comunicação não é bem sucedida.

- d) Pela imagem a baixo podemos confirmar o sucesso do comando *Route Add* no router n16.

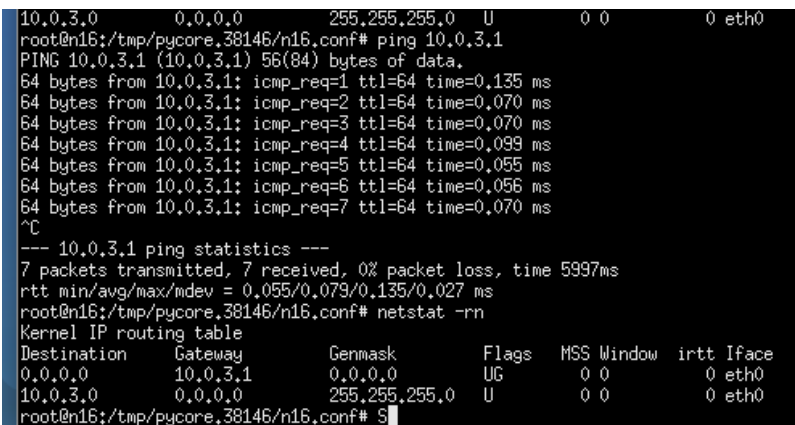


```
root@n16:/tmp/pycore.38146/n16.conf# netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask        Flags   MSS Window  irtt Iface
10.0.3.0         0.0.0.0        255.255.255.0  U       0 0        0 eth0
root@n16:/tmp/pycore.38146/n16.conf# route add default gw 10.0.3.1 eth0
root@n16:/tmp/pycore.38146/n16.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@n16:/tmp/pycore.38146/n16.conf#
```

- e) Pela imagem a baixo podemos confirmar que a nova política de encaminhamento funciona tornando o servidor novamente acessível. Tal é possível concluir através do teste de conectividade de um laptop neste departamento com o laptop de outro departamento.



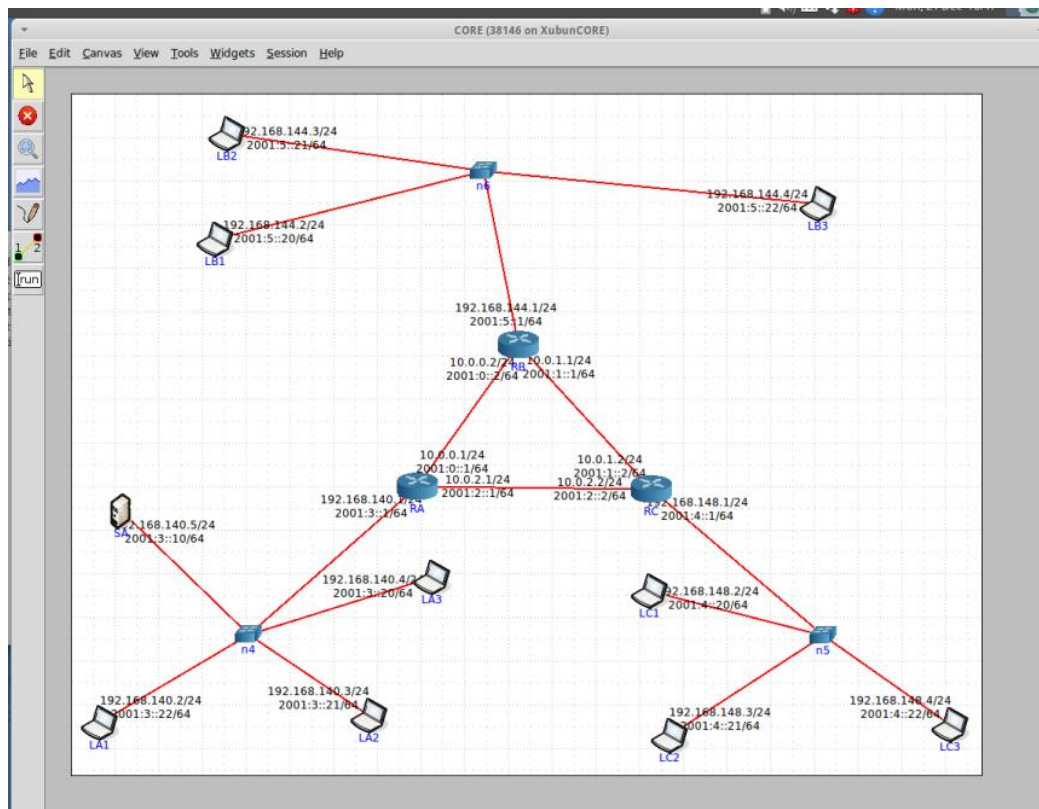
```
root@n16:/tmp/pycore.38146/n16.conf# ping 10.0.3.1
PING 10.0.3.1 (10.0.3.1) 56(84) bytes of data:
64 bytes from 10.0.3.1: icmp_req=1 ttl=64 time=0.135 ms
64 bytes from 10.0.3.1: icmp_req=2 ttl=64 time=0.070 ms
64 bytes from 10.0.3.1: icmp_req=3 ttl=64 time=0.070 ms
64 bytes from 10.0.3.1: icmp_req=4 ttl=64 time=0.099 ms
64 bytes from 10.0.3.1: icmp_req=5 ttl=64 time=0.055 ms
64 bytes from 10.0.3.1: icmp_req=6 ttl=64 time=0.056 ms
64 bytes from 10.0.3.1: icmp_req=7 ttl=64 time=0.070 ms
^C
--- 10.0.3.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5997ms
rtt min/avg/max/mdev = 0.055/0.079/0.135/0.027 ms
root@n16:/tmp/pycore.38146/n16.conf#
```



```
root@n16:/tmp/pycore.38146/n16.conf# ping 10.0.3.1
PING 10.0.3.1 (10.0.3.1) 56(84) bytes of data:
64 bytes from 10.0.3.1: icmp_req=1 ttl=64 time=0.135 ms
64 bytes from 10.0.3.1: icmp_req=2 ttl=64 time=0.070 ms
64 bytes from 10.0.3.1: icmp_req=3 ttl=64 time=0.070 ms
64 bytes from 10.0.3.1: icmp_req=4 ttl=64 time=0.099 ms
64 bytes from 10.0.3.1: icmp_req=5 ttl=64 time=0.055 ms
64 bytes from 10.0.3.1: icmp_req=6 ttl=64 time=0.056 ms
64 bytes from 10.0.3.1: icmp_req=7 ttl=64 time=0.070 ms
^C
--- 10.0.3.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5997ms
rtt min/avg/max/mdev = 0.055/0.079/0.135/0.027 ms
root@n16:/tmp/pycore.38146/n16.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@n16:/tmp/pycore.38146/n16.conf#
```

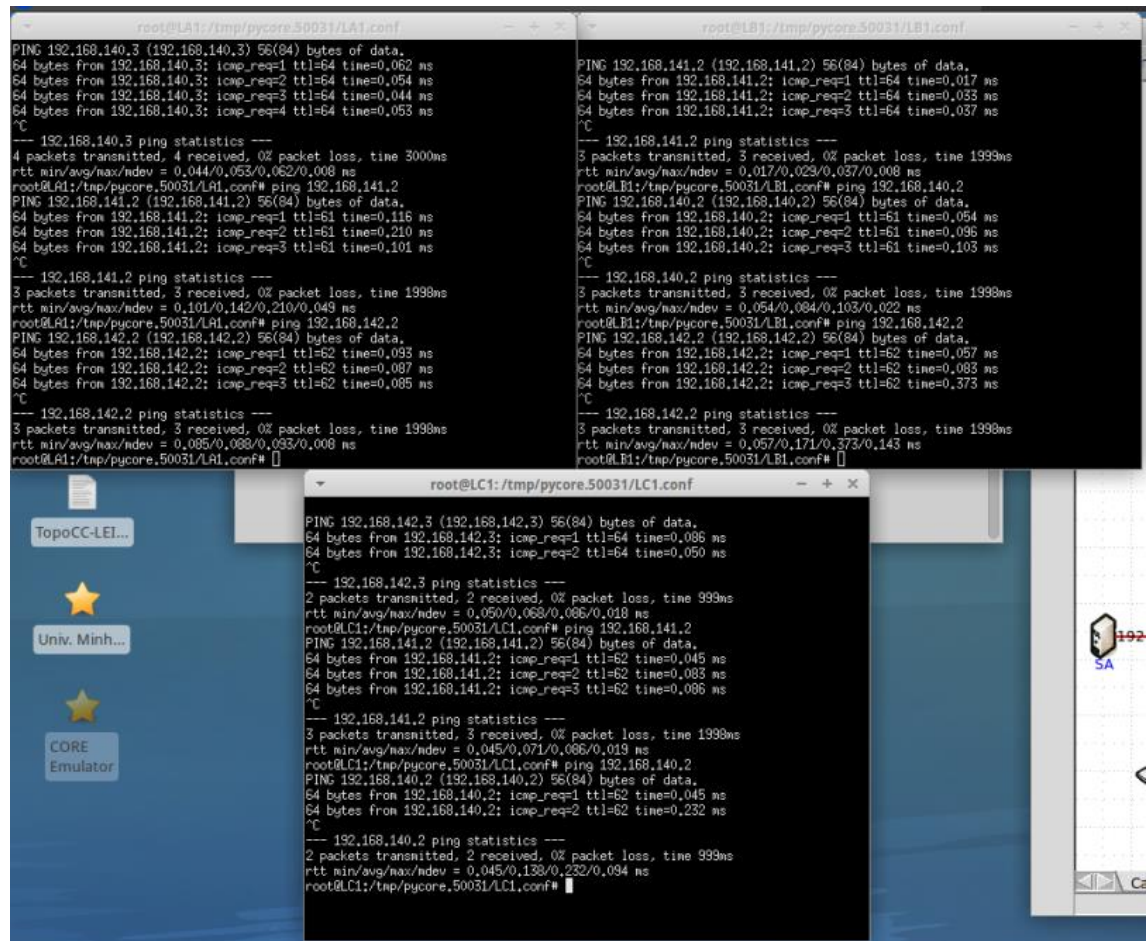
3. Definição de Sub-redes

- 1) Na figura a baixo encontra-se o novo esquema de endereçamento para as redes dos departamentos partindo do endereço IP 192.168.140.0/24.



- 2) A máscara usada é /24 que corresponde ao endereço 255.255.255.0, sendo em decimal equivalente a 24.
- 3) Em cada departamento é possível conectar 254 hosts ($(2^8)-2$). Os 2 endereços retirados (caso contrário seria 256) estão reservados para *Broadcast* e identificador de sub-rede.

- 4) No terminal superior esquerdo foi realizado um *ping* do departamento A para o B e do A para o C provando assim a comunicação entre os mesmos. Já no superior direito foi realizado um *ping* do B para o A e para o C. E por fim no inferior o *ping* foi realizado do C para o A e para o B. Podemos concluir assim que os departamentos comunicam entre si.



The screenshot displays three terminal windows from the CORE Emulator, each showing the results of a series of ping tests between three departments: A (192.168.140.3), B (192.168.141.2), and C (192.168.142.3). Each department has a corresponding configuration file (LA1.conf, LB1.conf, and LC1.conf) located in the /tmp/pycore.50031 directory.

Terminal 1 (Left): root@LA1:/tmp/pycore.50031/LA1.conf

```
PING 192.168.140.3 (192.168.140.3) 56(84) bytes of data,
64 bytes from 192.168.140.3: icmp_req=1 ttl=64 time=0.062 ms
64 bytes from 192.168.140.3: icmp_req=2 ttl=64 time=0.054 ms
64 bytes from 192.168.140.3: icmp_req=3 ttl=64 time=0.044 ms
64 bytes from 192.168.140.3: icmp_req=4 ttl=64 time=0.053 ms
^C
--- 192.168.140.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/ndev = 0.044/0.053/0.062/0.008 ms
root@LA1:/tmp/pycore.50031/LA1.conf# ping 192.168.141.2
PING 192.168.141.2 (192.168.141.2) 56(84) bytes of data,
64 bytes from 192.168.141.2: icmp_req=1 ttl=61 time=0.116 ms
64 bytes from 192.168.141.2: icmp_req=2 ttl=61 time=0.210 ms
64 bytes from 192.168.141.2: icmp_req=3 ttl=61 time=0.101 ms
^C
--- 192.168.141.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.101/0.142/0.210/0.049 ms
root@LA1:/tmp/pycore.50031/LA1.conf# ping 192.168.142.2
PING 192.168.142.2 (192.168.142.2) 56(84) bytes of data,
64 bytes from 192.168.142.2: icmp_req=1 ttl=62 time=0.093 ms
64 bytes from 192.168.142.2: icmp_req=2 ttl=62 time=0.087 ms
64 bytes from 192.168.142.2: icmp_req=3 ttl=62 time=0.085 ms
^C
--- 192.168.142.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.085/0.088/0.093/0.008 ms
root@LA1:/tmp/pycore.50031/LA1.conf#
```

Terminal 2 (Right): root@LB1:/tmp/pycore.50031/LB1.conf

```
PING 192.168.141.2 (192.168.141.2) 56(84) bytes of data,
64 bytes from 192.168.141.2: icmp_req=1 ttl=64 time=0.017 ms
64 bytes from 192.168.141.2: icmp_req=2 ttl=64 time=0.033 ms
64 bytes from 192.168.141.2: icmp_req=3 ttl=64 time=0.037 ms
^C
--- 192.168.141.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.017/0.029/0.037/0.008 ms
root@LB1:/tmp/pycore.50031/LB1.conf# ping 192.168.140.2
PING 192.168.140.2 (192.168.140.2) 56(84) bytes of data,
64 bytes from 192.168.140.2: icmp_req=1 ttl=61 time=0.054 ms
64 bytes from 192.168.140.2: icmp_req=2 ttl=61 time=0.096 ms
64 bytes from 192.168.140.2: icmp_req=3 ttl=61 time=0.103 ms
^C
--- 192.168.140.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.054/0.094/0.103/0.022 ms
root@LB1:/tmp/pycore.50031/LB1.conf# ping 192.168.142.2
PING 192.168.142.2 (192.168.142.2) 56(84) bytes of data,
64 bytes from 192.168.142.2: icmp_req=1 ttl=62 time=0.057 ms
64 bytes from 192.168.142.2: icmp_req=2 ttl=62 time=0.083 ms
64 bytes from 192.168.142.2: icmp_req=3 ttl=62 time=0.373 ms
^C
--- 192.168.142.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.057/0.171/0.373/0.143 ms
root@LB1:/tmp/pycore.50031/LB1.conf#
```

Terminal 3 (Bottom): root@LC1:/tmp/pycore.50031/LC1.conf

```
PING 192.168.142.3 (192.168.142.3) 56(84) bytes of data,
64 bytes from 192.168.142.3: icmp_req=1 ttl=64 time=0.086 ms
64 bytes from 192.168.142.3: icmp_req=2 ttl=64 time=0.050 ms
^C
--- 192.168.142.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 998ms
rtt min/avg/max/ndev = 0.050/0.068/0.086/0.018 ms
root@LC1:/tmp/pycore.50031/LC1.conf# ping 192.168.141.2
PING 192.168.141.2 (192.168.141.2) 56(84) bytes of data,
64 bytes from 192.168.141.2: icmp_req=1 ttl=62 time=0.045 ms
64 bytes from 192.168.141.2: icmp_req=2 ttl=62 time=0.083 ms
64 bytes from 192.168.141.2: icmp_req=3 ttl=62 time=0.096 ms
^C
--- 192.168.141.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.045/0.071/0.086/0.019 ms
root@LC1:/tmp/pycore.50031/LC1.conf# ping 192.168.140.2
PING 192.168.140.2 (192.168.140.2) 56(84) bytes of data,
64 bytes from 192.168.140.2: icmp_req=1 ttl=62 time=0.045 ms
64 bytes from 192.168.140.2: icmp_req=2 ttl=62 time=0.232 ms
^C
--- 192.168.140.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 998ms
rtt min/avg/max/ndev = 0.045/0.138/0.232/0.094 ms
root@LC1:/tmp/pycore.50031/LC1.conf#
```

Conclusão

Após a realização do trabalho proposto, o nosso grupo considera que esta parte prática nos proporcionou uma melhoria na aptidão (quer a nível teórico quer a nível prático) para análise e gestão de informações sobre o funcionamento dos *Internet Protocol* (IP), nomeadamente o estudo de datagramas, endereçamento e encaminhamento de IP assim como a fragmentação de pacotes de IP. Deste modo, através da utilização do Wireshark e do *PingPlotter 5*, este projeto proporcionou-se um maior aprofundamento no endereçamento e encaminhamento do IP. Dito isto, de uma forma sucinta o nosso grupo afirma que a realização deste trabalho prático foi enriquecedora e bastante útil para nos possibilitar uma melhor perspetiva das matérias lecionadas nas aulas teóricas.