

UNIVERSIDADE DO MINHO
DEPARTAMENTO DE INFORMÁTICA

Redes de Computadores
TP2:Protocolo IPv4
Grupo N° 8 PL6

Gonçalo Almeida (A84610)

Emanuel Rodrigues (A84776)

Lázaro Pinheiro (A86788)

13 de Novembro de 2019

Conteúdo

1	Questões e Respostas	3
1.1	Parte 1 - Datagramas IP e Fragmentação	3
1.2	Parte 2 - Endereçamento e Encaminhamento IP	13
2	Conclusão	34

Capítulo 1

Questões e Respostas

1.1 Parte 1 - Datagramas IP e Fragmentação

1. Prepare uma topologia no CORE para verificar o comportamento do traceroute. Ligue um host (servidor) s1 a um router r2; o router r2 a um router r3, o router r3 a um router r4, que por sua vez, se liga a um host (pc) h5. (Note que pode não existir conectividade IP imediata entre s1 e h5 até que o routing estabilize). Ajuste o nome dos equipamentos atribuídos por defeito para a topologia do enunciado.

Implementamos a seguinte topologia core:

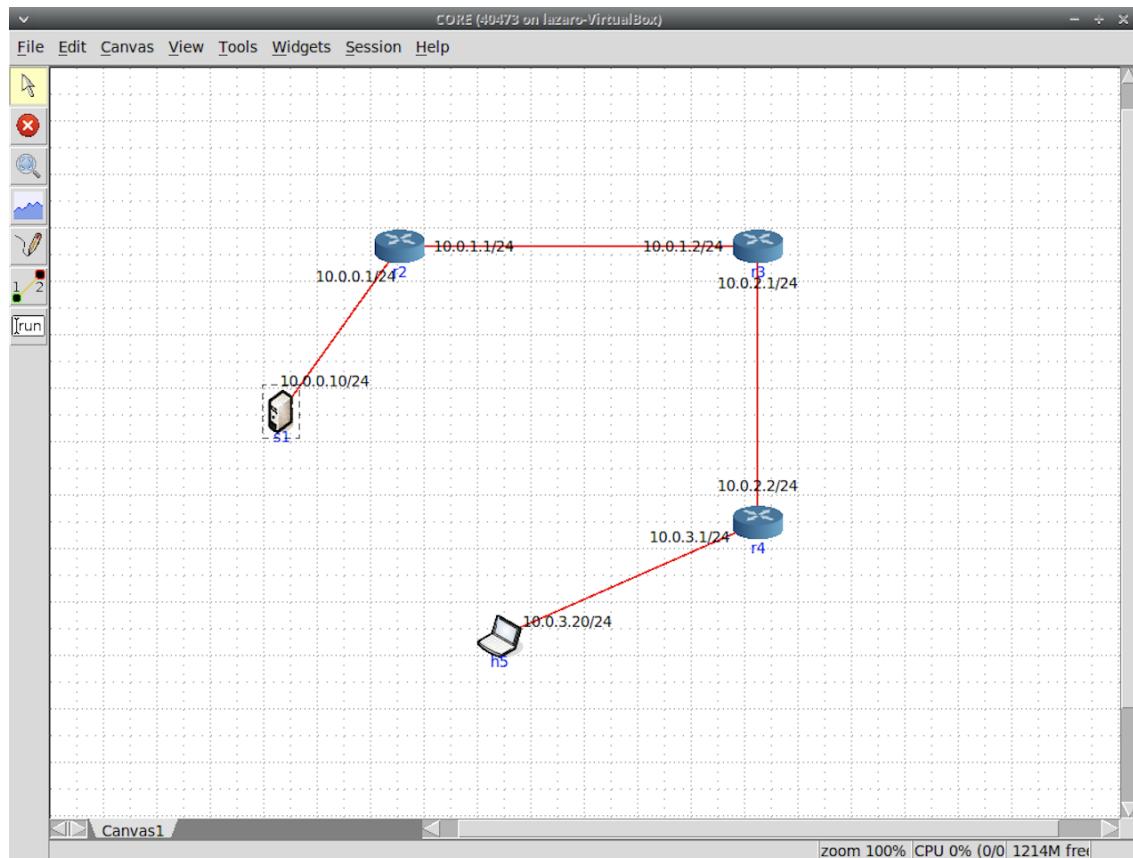


Figura 1.1: Topologia Core

- (a) Active o wireshark ou o tcpdump no pc s1. Numa shell de s1, execute o comando traceroute -I para o endereço IP do host h5.

Ativamos o wireshark no servidor s1 executando o comando traceroute -I para o endereço IP do host h5 (10.0.3.20).

```
root@s1:/tmp/pycore.40473/s1.conf# traceroute -I 10.0.3.20
traceroute to 10.0.3.20 (10.0.3.20), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  0.030 ms  0.005 ms  0.004 ms
 2  10.0.1.2 (10.0.1.2)  0.013 ms  0.007 ms  0.005 ms
 3  10.0.2.2 (10.0.2.2)  0.014 ms  0.008 ms  0.008 ms
 4  10.0.3.20 (10.0.3.20)  0.016 ms  0.010 ms  0.009 ms
root@s1:/tmp/pycore.40473/s1.conf#
```

Figura 1.2: Comando Traceroute

- (b) Registe e analise o tráfego ICMP enviado por s1 e o tráfego ICMP recebido como resposta. Comente os resultados face ao comportamento esperado.

No.	Time	Source	Destination	Protocol	Length	Info
1	70 28.542132782	10.0.3.20	10.0.0.10	ICMP	74	Echo (ping) reply id=0x0081, seq=14/3584, ttl=61 (request in 69)
	71 28.542135681	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0081, seq=15/3840, ttl=5 (reply in 72)
	72 28.542144584	10.0.3.20	10.0.0.10	ICMP	74	Echo (ping) reply id=0x0081, seq=15/3840, ttl=61 (request in 71)
	73 28.542148181	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0081, seq=16/4096, ttl=6 (reply in 74)
	74 28.542157349	10.0.3.20	10.0.0.10	ICMP	74	Echo (ping) reply id=0x0081, seq=16/4096, ttl=61 (request in 73)
	75 29.983366411	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=1/256, ttl=1 (no response found!)
	76 29.983386011	10.0.0.1	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	77 29.983385759	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=2/512, ttl=1 (no response found!)
	78 29.983388544	10.0.0.1	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	79 29.983396775	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=3/768, ttl=1 (no response found!)
	80 29.983392951	10.0.0.1	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	81 29.983395536	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=4/1024, ttl=2 (no response found!)
	82 29.983405694	10.0.1.2	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	83 29.983408859	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=5/1280, ttl=2 (no response found!)
	84 29.983411955	10.0.1.2	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	85 29.983414819	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=6/1536, ttl=2 (no response found!)
	86 29.983417918	10.0.1.2	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	87 29.983426478	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=7/1792, ttl=3 (no response found!)
	88 29.983431838	10.0.2.2	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	89 29.983434265	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=8/2848, ttl=3 (no response found!)
	90 29.983440971	10.0.2.2	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	91 29.983442395	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=9/2384, ttl=3 (no response found!)
	92 29.983448287	10.0.2.2	10.0.0.10	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
	93 29.983459494	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=10/2560, ttl=4 (reply in 94)
	94 29.983462526	10.0.3.20	10.0.0.10	ICMP	74	Echo (ping) reply id=0x0082, seq=10/2560, ttl=61 (request in 93)
	95 29.983465717	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=11/2816, ttl=4 (reply in 96)
	96 29.983472494	10.0.3.20	10.0.0.10	ICMP	74	Echo (ping) reply id=0x0082, seq=11/2816, ttl=61 (request in 95)
	97 29.983474832	10.0.0.10	10.0.3.20	ICMP	74	Echo (ping) request id=0x0082, seq=12/3072, ttl=4 (reply in 98)

Figura 1.3: Tráfego ICMP enviado e recebido por s1

O s1 envia request com TTL = 1 para o h5 e o r2 responde que o TTL foi excedido, repetindo este processo 3 vezes.

O s1 envia request com TTL = 2 para o h5 e o r3 responde que o TTL foi excedido, repetindo este processo 3 vezes.

O s1 envia request com TTL = 3 para o h5 e o r4 responde que o TTL foi excedido, repetindo este processo 3 vezes.

O s1 envia request com TTL = 4 para o h5 e este responde que recebeu o pacote.

- (c) Qual deve ser o valor inicial mínimo do campo TTL para alcançar o destino h5? Verifique na prática que a sua resposta está correta.

O valor inicial mínimo para o TTL é 4, pois por cada router que passa é diminuído em 1, chegando ao host (destino) com TTL 0. Como podemos observar na figura 1.3, os pacotes com TTL inferior a 4 recebem uma mensagem de erro.

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

Pela figura 1.2 temos:

$$RTT = ((0.03 + 0.005 + 0.004)/3 + (0.013 + 0.007 + 0.005)/3 + (0.014 + 0.008 + 0.008)/3 + (0.016 + 0.010 + 0.009)/3) \times 2 = 0.086$$

2. Pretende-se agora usar o traceroute na sua máquina nativa, e gerar de data-gramas IP de diferentes tamanhos. Selecione a primeira mensagem ICMP capturada (referente a (i) tamanho por defeito) e centre a análise no nível protocolar IP (expanda o tab correspondente na janela de detalhe do wireshark).

Obtivemos o seguinte tráfego fazendo traceroute -I marco.uminho.pt

No.	Time	Source	Destination	Protocol	Length	Info
13	9.192104	172.26.29.82	193.137.16.65	DNS	83	Standard query 0x29d4 PTR 1.2.16.172.in-addr.arpa
14	9.194701	193.137.16.65	172.26.29.82	DNS	91	Standard query response 0x29d4 Refused PTR 1.2.16.172.in-addr.arpa
15	9.194883	172.26.29.82	193.137.16.145	DNS	83	Standard query 0x29d4 PTR 1.2.16.172.in-addr.arpa
16	9.197851	193.137.16.145	172.26.29.82	DNS	91	Standard query response 0x29d4 Refused PTR 1.2.16.172.in-addr.arpa
17	9.198068	172.26.29.82	193.137.16.75	DNS	83	Standard query 0x29d4 PTR 1.2.16.172.in-addr.arpa
18	9.200238	193.137.16.75	172.26.29.82	DNS	91	Standard query response 0x29d4 Refused PTR 1.2.16.172.in-addr.arpa
25	10.214583	172.26.29.82	193.137.16.65	DNS	87	Standard query 0x3243 PTR 252.115.16.172.in-addr.arpa
26	10.217051	193.137.16.65	172.26.29.82	DNS	95	Standard query response 0x3243 Refused PTR 252.115.16.172.in-addr.arpa
27	10.217220	172.26.29.82	193.137.16.145	DNS	87	Standard query 0x3243 PTR 252.115.16.172.in-addr.arpa
28	10.219972	193.137.16.145	172.26.29.82	DNS	95	Standard query response 0x3243 Refused PTR 252.115.16.172.in-addr.arpa
29	10.220192	172.26.29.82	193.137.16.75	DNS	87	Standard query 0x3243 PTR 252.115.16.172.in-addr.arpa
30	10.222262	193.137.16.75	172.26.29.82	DNS	95	Standard query response 0x3243 Refused PTR 252.115.16.172.in-addr.arpa
5	9.176739	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=1/256, ttl=1 (no response found!)
6	9.184905	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
7	9.185572	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=2/512, ttl=1 (no response found!)
8	9.187460	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
9	9.187573	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=3/768, ttl=1 (no response found!)
10	9.189396	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
11	9.189583	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=4/1024, ttl=2 (no response found!)
12	9.191490	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
19	10.201655	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=5/1280, ttl=2 (no response found!)
20	10.209622	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
21	10.289743	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=6/1536, ttl=2 (no response found!)
22	10.211694	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
23	10.211829	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=7/1792, ttl=3 (no response found!)
24	10.214020	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
31	11.225887	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=8/2048, ttl=3 (no response found!)
32	11.234149	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
33	11.234316	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=9/2304, ttl=3 (no response found!)
34	11.239514	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
35	11.239660	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=10/2560, ttl=4 (reply in 36)
36	11.242221	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=10/2560, ttl=6 (request in 35)
37	11.242802	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=11/2816, ttl=4 (reply in 38)
38	11.245339	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=11/2816, ttl=6 (request in 37)
39	11.245462	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=12/3072, ttl=4 (reply in 40)
40	11.247624	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=12/3072, ttl=6 (request in 39)
2	0.022530	216.58.201.138	172.26.29.82	TCP	66	443 → 50098 [ACK] Seq=1 Ack=40 Win=251 Len=0 TSecr=821281811
4	0.022592	172.26.29.82	216.58.201.138	TCP	66	50098 → 443 [ACK] Seq=40 Ack=40 Win=2047 Len=0 TSecr=821281833 TSecr=2378356261
1	0.000000	172.26.29.82	216.58.201.138	TLSv1...	105	Application Data
3	0.022534	216.58.201.138	172.26.29.82	TLSv1...	105	Application Data

Figura 1.4: Tráfego de pacotes

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

```

▶ Frame 5: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▶ Ethernet II, Src: Apple_87:e4:78 (88:e9:fe:87:e4:78), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
▼ Internet Protocol Version 4, Src: 172.26.29.82, Dst: 193.136.9.240
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
▶ Differentiated Services Field: 0x00 (DSSCP: CS0, ECN: Not-ECT)
  Total Length: 72
  Identification: 0x8493 (33939)
▼ 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)
  Header checksum: 0xa03d [validation disabled]
  [Header checksum status: Unverified]
  Source: 172.26.29.82
  Destination: 193.136.9.240
▶ Internet Control Message Protocol

```

0000	00	d0	03	ff	94	00	88	e9	fe	87	e4	78	08	00	45	00x..E.
0010	00	48	84	93	00	00	01	01	a0	3d	ac	1a	1d	52	c1	88	H.....	=...R..
0020	09	f0	08	00	73	6c	84	92	00	01	00	00	00	00	00	00sl..
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figura 1.5: Datagrama 1

O endereço IP da interface ativa do computador é 172.26.29.82 indicado no campo Source.

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

O valor do campo protocolo é 1 e identifica o protocolo ICMP.

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

O cabeçalho IP(v4) tem 20 bytes, indicado no campo Header Length do datagrama. O campo de dados (payload) do datagrama tem 52 bytes (Total Length – Header Length = 72 – 20 bytes).

(d) O datagrama IP foi fragmentado? Justifique.

No datagrama podemos verificar que a flag More fragments está a 0, ou seja, não existem mais fragmentos para além do atual.

(e) Ordene os pacotes capturados de acordo com o endereço IP fonte (e.g., selecionando o cabeçalho da coluna Source), e analise a sequência de tráfego ICMP gerado a partir do endereço IP atribuído à interface da sua máquina. Para a sequência de mensagens ICMP enviadas pelo seu computador, indique que campos do cabeçalho IP variam de pacote para pacote.

No.	Time	Source	Destination	Protocol	Length	Info
24	10.214020	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
32	11.234149	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
34	11.239514	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
12	9.191490	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
20	10.209622	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
22	10.211694	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
6	9.184905	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
8	9.187460	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
10	9.189396	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
5	9.176739	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=1/256, ttl=1 (no response found!)
7	9.185572	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=2/512, ttl=1 (no response found!)
9	9.187573	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=3/768, ttl=1 (no response found!)
11	9.189503	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=4/1024, ttl=2 (no response found!)
19	10.201655	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=5/1280, ttl=2 (no response found!)
21	10.209743	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=6/1536, ttl=2 (no response found!)
23	10.211829	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=7/1792, ttl=3 (no response found!)
31	11.225887	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=8/2048, ttl=3 (no response found!)
33	11.234316	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=9/2304, ttl=3 (no response found!)
35	11.239660	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=10/2560, ttl=4 (reply in 36)
37	11.242802	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=11/2816, ttl=4 (reply in 38)
39	11.245462	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=12/3072, ttl=4 (reply in 40)
36	11.242221	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=10/2560, ttl=61 (request in 35)
38	11.245339	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=11/2816, ttl=61 (request in 37)
40	11.247624	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=12/3072, ttl=61 (request in 39)
5	9.176739	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=1/256, ttl=1 (no response found!)
7	9.185572	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=2/512, ttl=1 (no response found!)
9	9.187573	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=3/768, ttl=1 (no response found!)
11	9.189503	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=4/1024, ttl=2 (no response found!)
19	10.201655	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=5/1280, ttl=2 (no response found!)
21	10.209743	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=6/1536, ttl=2 (no response found!)
23	10.211829	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=7/1792, ttl=3 (no response found!)
31	11.225887	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=8/2048, ttl=3 (no response found!)
33	11.234316	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=9/2304, ttl=3 (no response found!)
35	11.239660	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=10/2560, ttl=4 (reply in 36)
37	11.242802	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=11/2816, ttl=4 (reply in 38)
39	11.245462	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=12/3072, ttl=4 (reply in 40)

Figura 1.6: Pacotes capturados ordenados de acordo com o endereço IP da fonte

Os campos do cabeçalho IP, que variam de pacote para pacote, são o TTL no campo Time to Live e o identificador do pacote no campo Identification.

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

Ambos aumentam, sendo que o valor do TTL aumenta 1 unidade a cada 3 diagramas e o valor do identificador do pacote aumenta 1 unidade a cada diagrama.

(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?

No.	Time	Source	Destination	Protocol	Length	Info
6	9.184905	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
8	9.187460	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
10	9.189396	172.26.254.254	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
12	9.191490	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
20	10.209622	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
22	10.211694	172.16.2.1	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
24	10.214020	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
32	11.234149	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
34	11.239514	172.16.115.252	172.26.29.82	ICMP	78	Time-to-live exceeded (Time to live exceeded in transit)
36	11.242221	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=10/2560, ttl=61 (request in 35)
38	11.245339	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=11/2816, ttl=61 (request in 37)
40	11.247624	193.136.9.240	172.26.29.82	ICMP	94	Echo (ping) reply id=0x8492, seq=12/3072, ttl=61 (request in 39)
5	9.176739	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=1/256, ttl=1 (no response found!)
7	9.185572	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=2/512, ttl=1 (no response found!)
9	9.187573	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=3/768, ttl=1 (no response found!)
11	9.189503	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=4/1024, ttl=2 (no response found!)
19	10.201655	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=5/1280, ttl=2 (no response found!)
21	10.209743	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=6/1536, ttl=2 (no response found!)
23	10.211829	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=7/1792, ttl=3 (no response found!)
31	11.225887	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=8/2048, ttl=3 (no response found!)
33	11.234316	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=9/2304, ttl=3 (no response found!)
35	11.239660	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=10/2560, ttl=4 (reply in 36)
37	11.242802	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=11/2816, ttl=4 (reply in 38)
39	11.245462	172.26.29.82	193.136.9.240	ICMP	86	Echo (ping) request id=0x8492, seq=12/3072, ttl=4 (reply in 40)

Figura 1.7: Tráfego capturado ordenado por endereço destino

```

▶ Frame 6: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
▶ Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00), Dst: Apple_87:e4:78 (88:e9:fe:87:e4:78)
▼ Internet Protocol Version 4, Src: 172.26.254.254, Dst: 172.26.29.82
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
▶ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
  Total Length: 56
  Identification: 0x3c73 (15475)
▼ 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: 255
  Protocol: ICMP (1)
  Header checksum: 0xa0c [validation disabled]
  [Header checksum status: Unverified]
  Source: 172.26.254.254
  Destination: 172.26.29.82
▶ Internet Control Message Protocol

```

Figura 1.8: 1º Datagrama da figura anterior

O valor do TTL vai decrementando em 1 unidade pois as mensagens de erro precisam de passar por outros routers para chegarem ao host.

3. Pretende-se agora analisar a fragmentação de pacotes IP. Reponha a ordem do tráfego capturado usando a coluna do tempo de captura. Observe o tráfego depois do tamanho de pacote ter sido definido para 42XX bytes.

```
[MacBook-Pro-de-Lazaro:~ lazarpinheiro$ traceroute -I marco.uminho.pt 4208
traceroute to marco.uminho.pt (193.136.9.240), 64 hops max, 4208 byte packets
 1  172.26.254.254 (172.26.254.254)  28.099 ms  11.187 ms  4.787 ms
 2  172.16.2.1 (172.16.2.1)  2.996 ms  9.404 ms  2.973 ms
 3  172.16.115.252 (172.16.115.252)  4.412 ms  9.679 ms  3.676 ms
 4  marco.uminho.pt (193.136.9.240)  3.570 ms  3.903 ms  3.507 ms
```

Figura 1.9: Comando traceroute

- (a) 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?

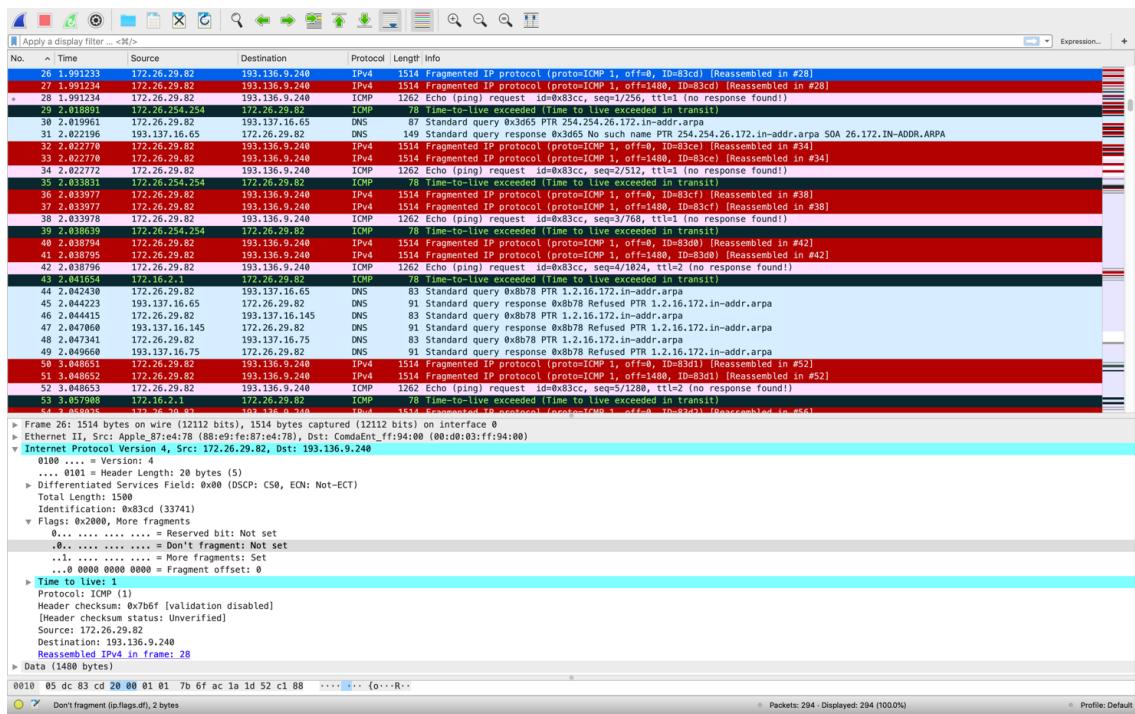


Figura 1.10: Tráfego de pacotes com datagrama do primeiro fragmento

Como podemos observar no campo Total Length, o tamanho máximo do pacote é de 1500 bytes. Como o tamanho do pacote que queremos enviar é 4208 bytes, há necessidade de o fragmentar, pois é demasiado grande para circular na rede.

- (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?

Como a flag More fragments está a 1, sabemos que existem mais fragmentos, e como a flag Fragment offset está a 0, sabemos que se trata do primeiro fragmento. Este diagrama tem 1480 bytes de tamanho (Total Length – Header Length = 1500 – 20).

- (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?

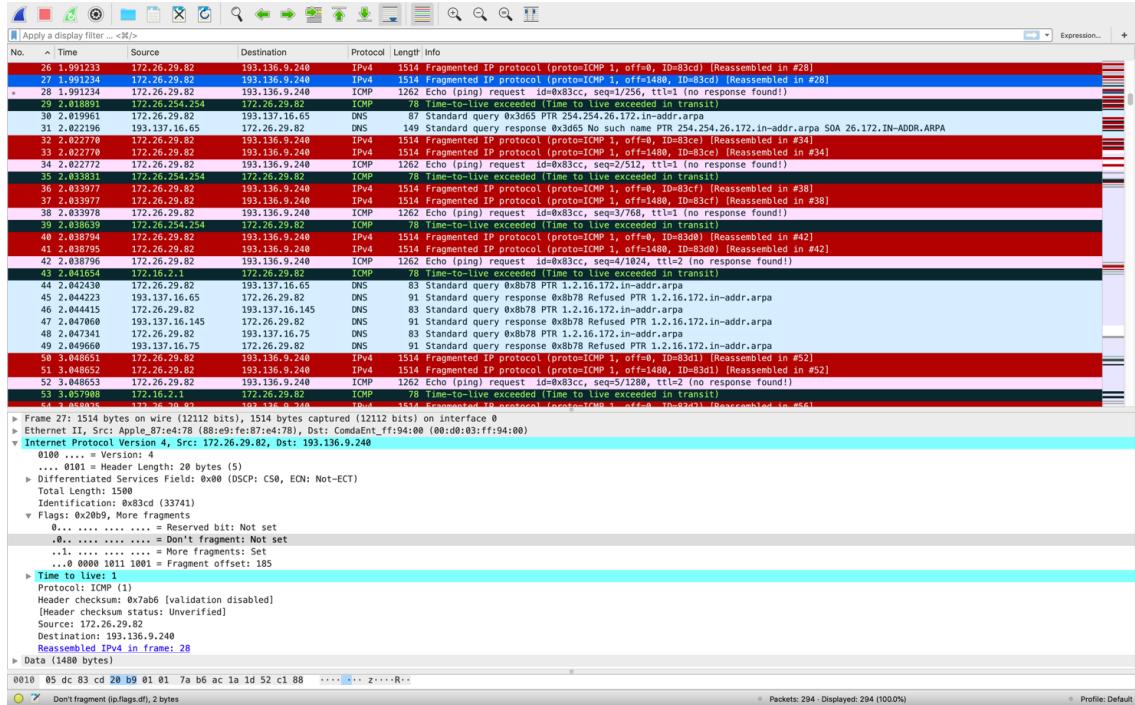


Figura 1.11: Tráfego de pacotes com datagrama do segundo fragmento

Podemos ver que não se trata do primeiro fragmento, dado que tem a flag Fragment offset a 185 e não a 0. Podemos também ver que existem mais fragmentos, visto que a flag More fragments está a 1.

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

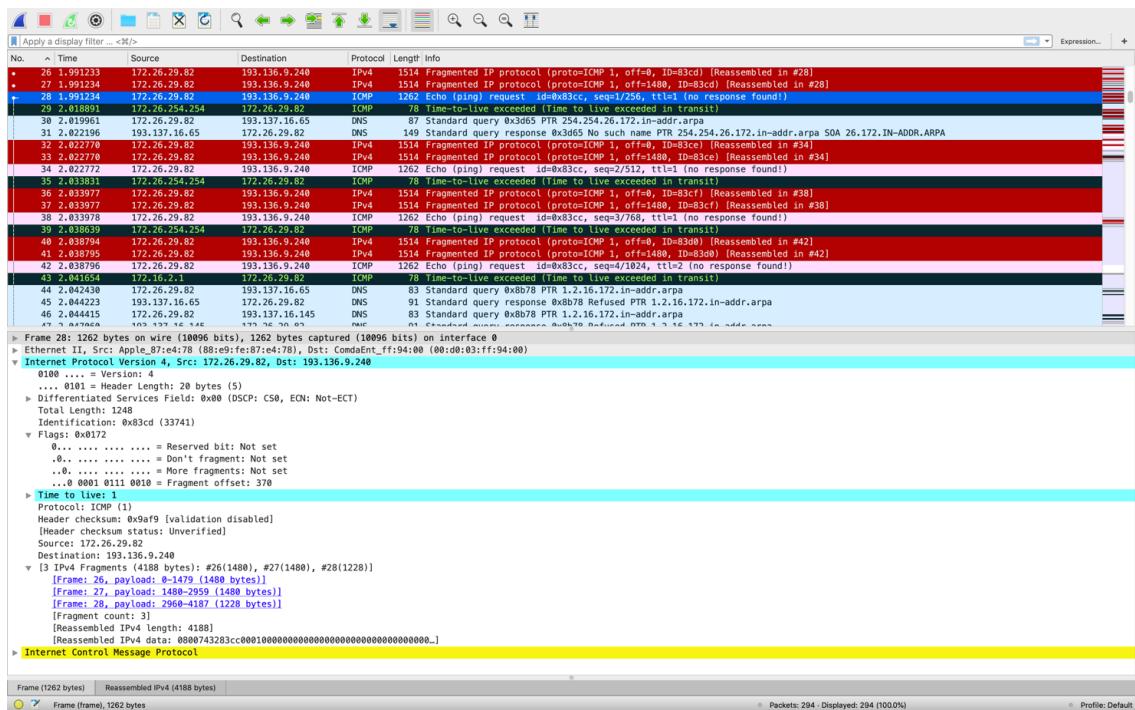


Figura 1.12: Tráfego de pacotes com datagrama do terceiro fragmento

A partir do datagrama original foram criados 3 fragmentos (evidenciado no campo Fragment Count: 3). Também podemos ver isto porque somando os tamanhos dos fragmentos #26, #27 e #28 ($1480 + 1480 + 1228$) o resultado da soma é 4208, que é o tamanho por defeito. O último fragmento é detetado quando a flag More fragments está a 0. Outra maneira de detetar que é o último fragmento, é através da soma dos offsets dos fragmentos anteriores, que é igual ao offset do último fragmento, e o facto da flag More fragments estar a 0.

- (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.

Os campos que mudam entre os diferentes fragmentos são as flags More fragments e Fragment offset.

1.2 Parte 2 - Endereçamento e Encaminhamento IP

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

- (a) Indique que endereços IP e máscaras de rede foram atribuídos pelo CORE a cada equipamento. Para simplificar, pode incluir uma imagem que ilustre de forma clara a topologia definida e o endereçamento usado.

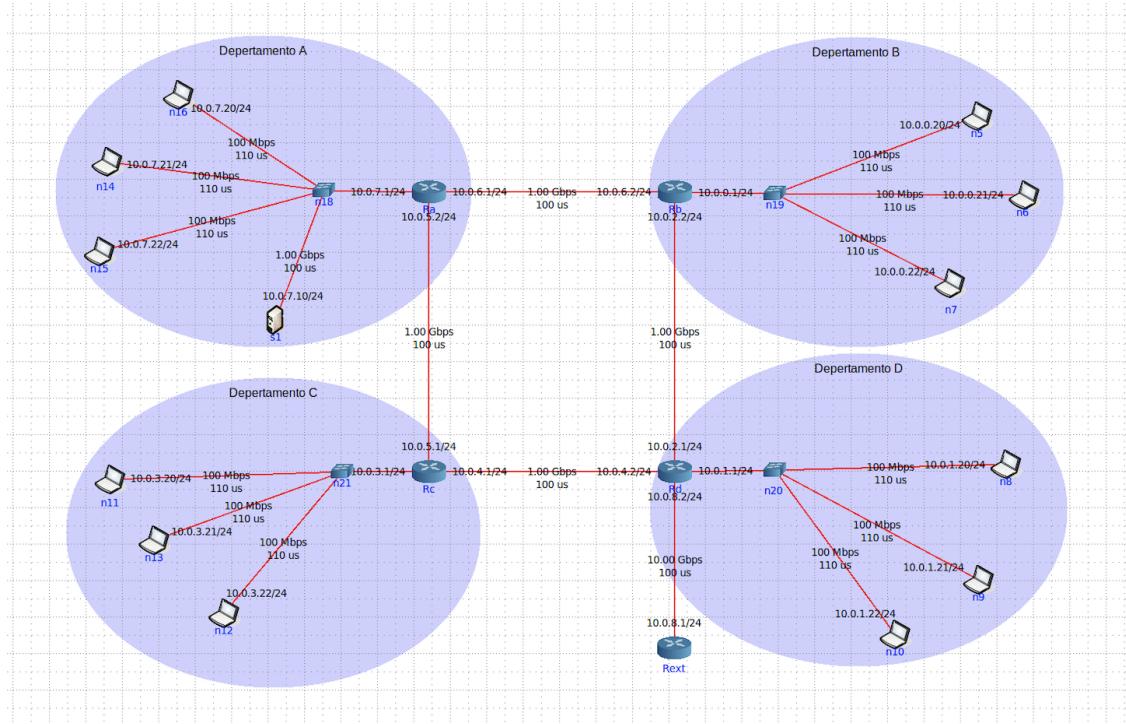


Figura 1.13: Topologia Core

Na figura 1.13 podemos ver os endereços IP de cada equipamento e também verificar que a máscara de rede é 255.255.255.0, visto que na notação CIDR o número de bits é 24 (/24).

- (b) Trata-se de endereços públicos ou privados? Porquê?

Como o intervalo de endereços 10.0.0.0 – 10.255.255.8 se trata de um intervalo de endereços privado e os endereços dos equipamentos estão incluídos nessa gama, podemos afirmar que são privados.

- (c) Por que razão não é atribuído um endereço IP aos switches?

Como os endereços IP apenas são atribuídos a entidades que operam na camada 3 (network layer) e, visto que os switches operam na camada 2 (link layer), estes não estão configurados com um endereço IP.

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

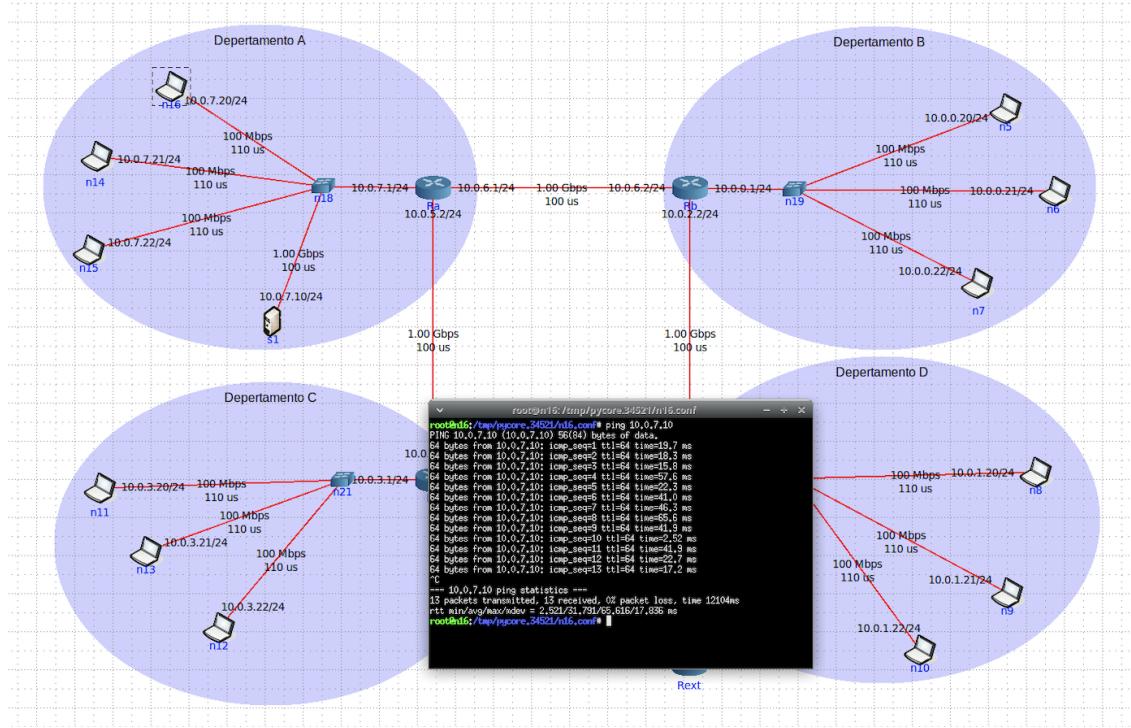


Figura 1.14: Conetividade entre o laptop do Departamento A e o servidor S1

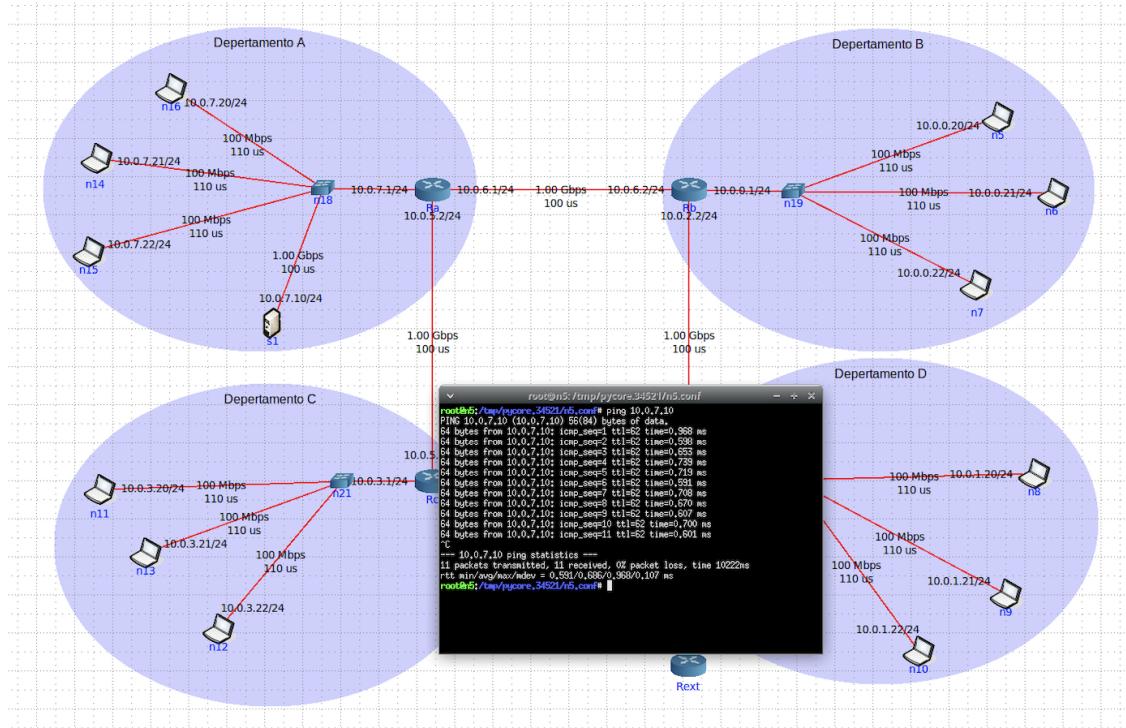


Figura 1.15: Conetividade entre o laptop do Departamento B e o servidor S1

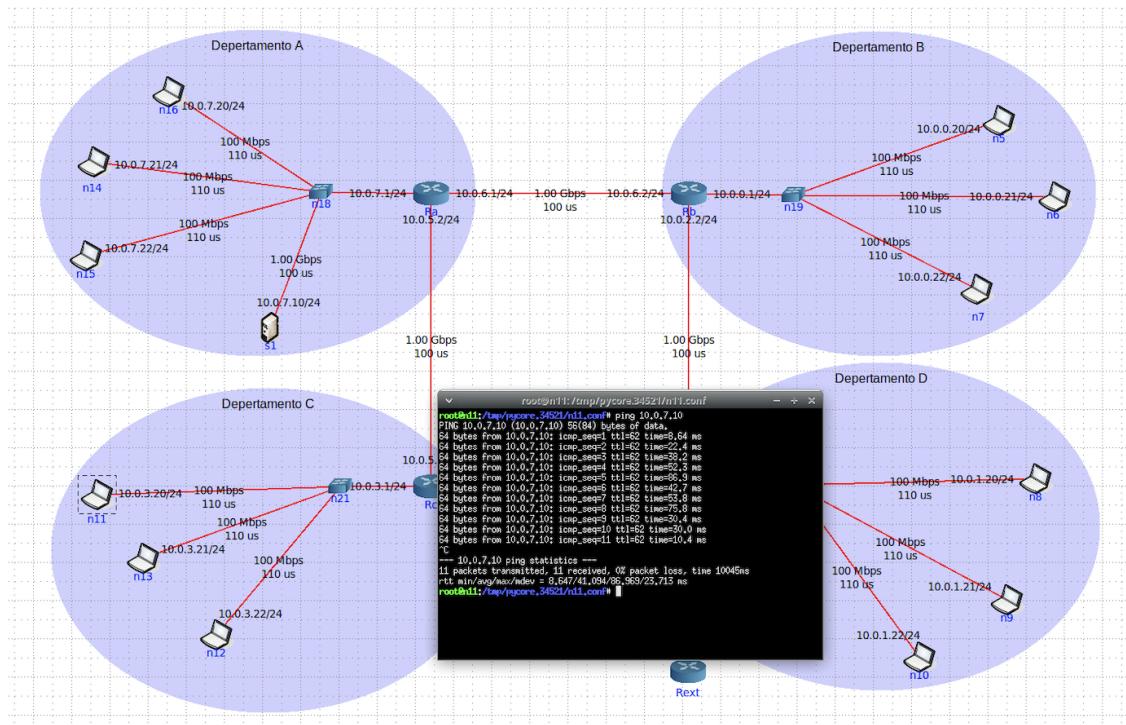


Figura 1.16: Conetividade entre o laptop do Departamento C e o servidor S1

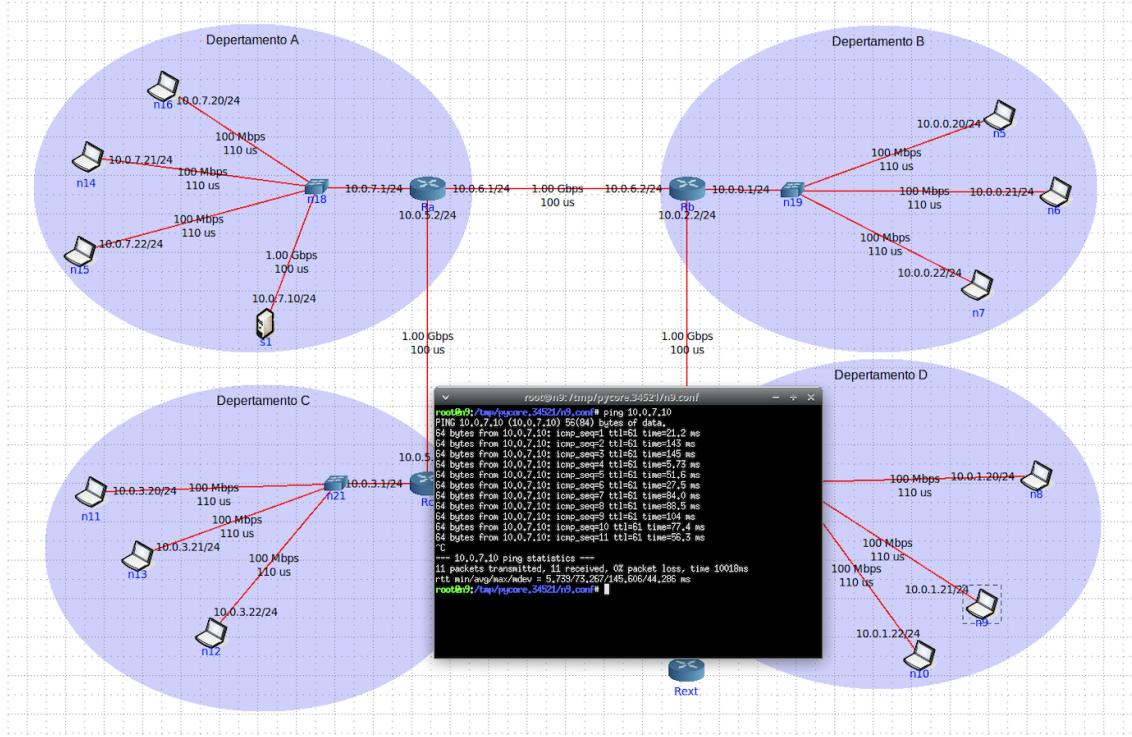


Figura 1.17: Conetividade entre o laptop do Departamento D e o servidor S1

Observando as figuras anteriores constatamos que os laptops de cada departamento enviam pacotes para o servidor S1 e este envia mensagens de volta com informação relativa ao tempo de ida e volta, demonstrando conetividade entre os laptops e o servidor. Caso não houvesse conetividade entre os laptops e o servidor, as mensagens que poderíamos observar seriam de erro.

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

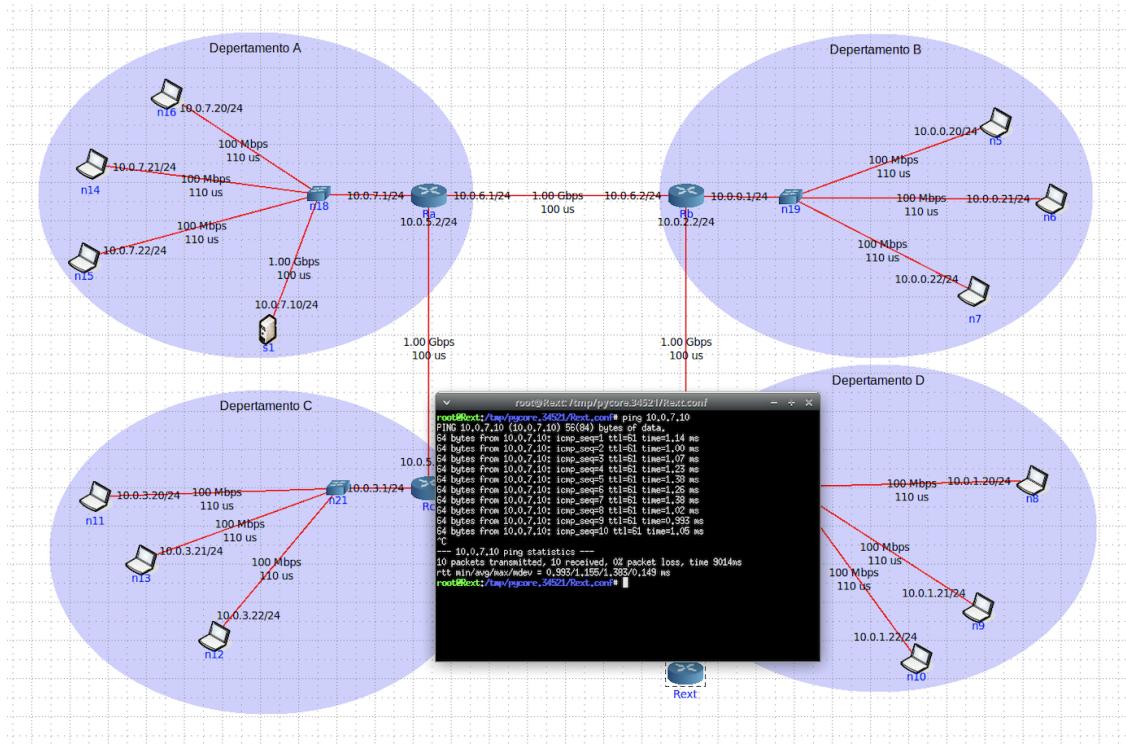


Figura 1.18: Conetividade entre o router de acesso Rext e o servidor S1

2. Para o router e um laptop do departamento B:

- (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`).

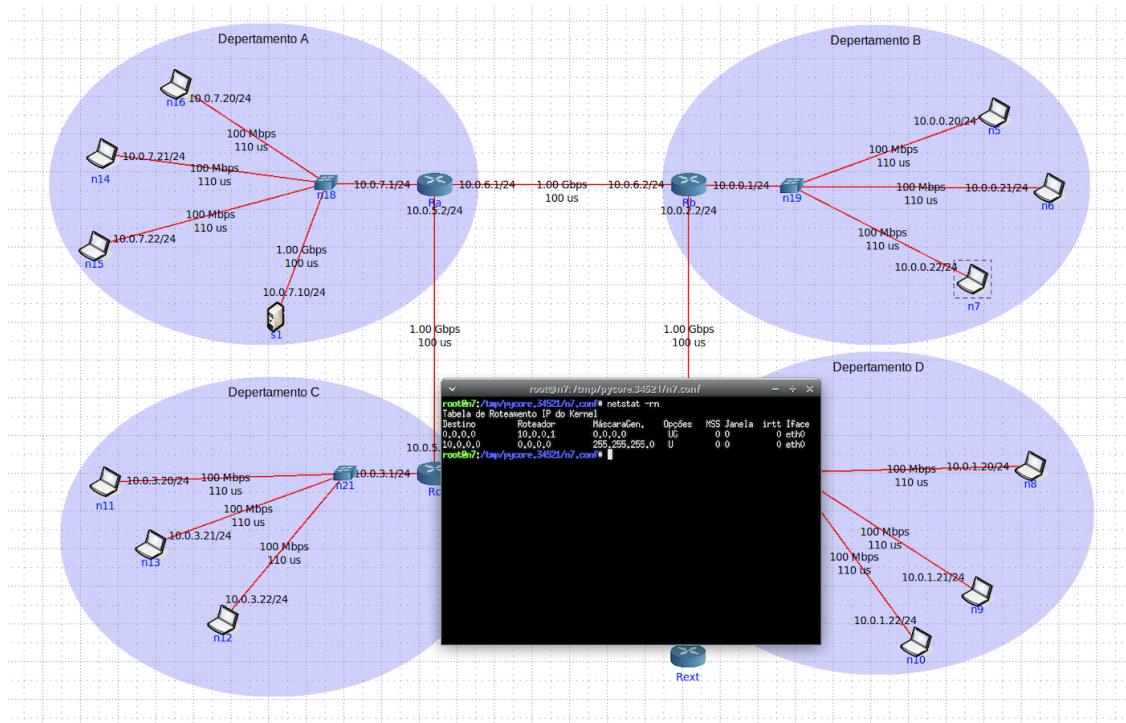


Figura 1.19: Tabela de encaminhamento do laptop n7

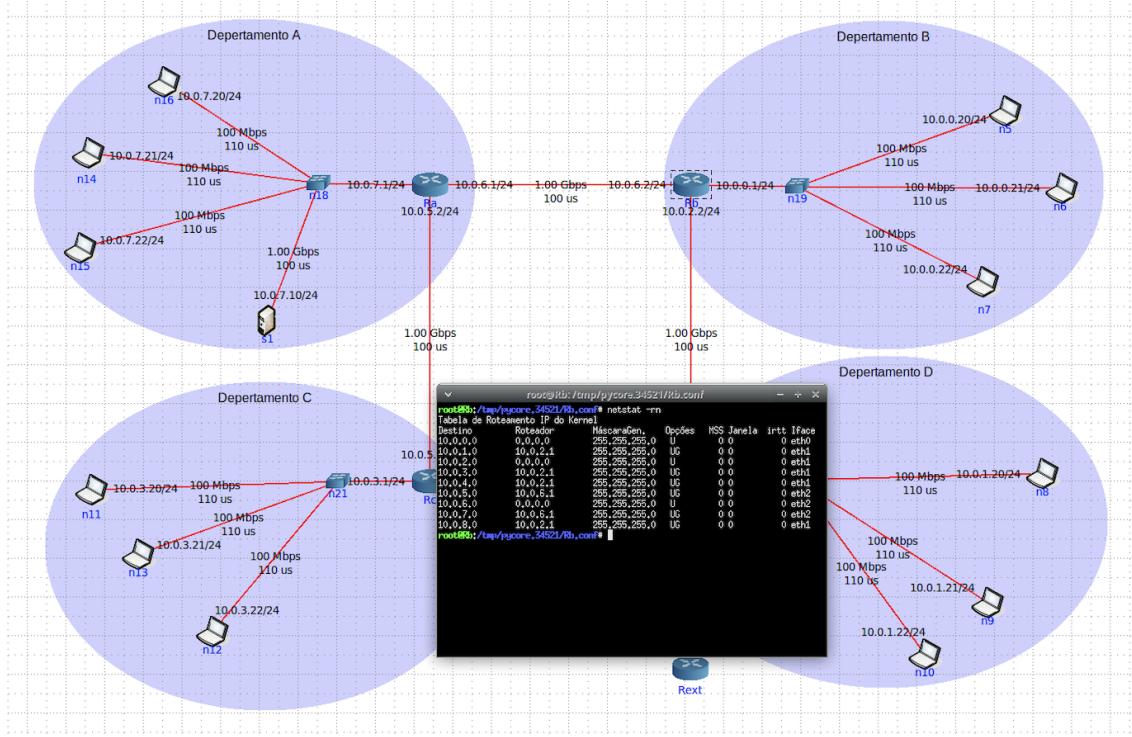


Figura 1.20: Tabela de encaminhamento do router Rb

Das tabelas de encaminhamento (figuras 1.19 e 1.20) retira-se informação sobre a rota do pacote. A coluna Destino indica a sub-rede destino, a coluna Roteador indica informação do equipamento por onde o pacote passa e a coluna MáscaraGen indica o tipo de máscara utilizada. Na primeira entrada da tabela de encaminhamento do laptop n7 podemos ver que, independentemente do endereço de destino, o pacote será entregue na interface de endereço 10.0.0.1 (endereço do router Rb), sendo este o endereço default. Na segunda entrada podemos observar que um pacote destinado à rede 10.0.0.0 (endereço da sub-rede do Departamento B) poderá optar por um destino.

Na primeira entrada da tabela de encaminhamento do router Rb podemos ver que pacotes com Roteador 0.0.0.0 podem seguir por qualquer rota. Nas seguintes entradas observa-se que os pacotes destinados às redes 10.0.1.0, 10.0.3.0, 10.0.4.0 e 10.0.8.0 serão entregues à interface de endereço 10.0.2.1 (router Rd) e os pacotes destinados às redes 10.0.5.0 e 10.0.7.0 serão entregues à interface de endereço 10.0.6.1 (router Ra).

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

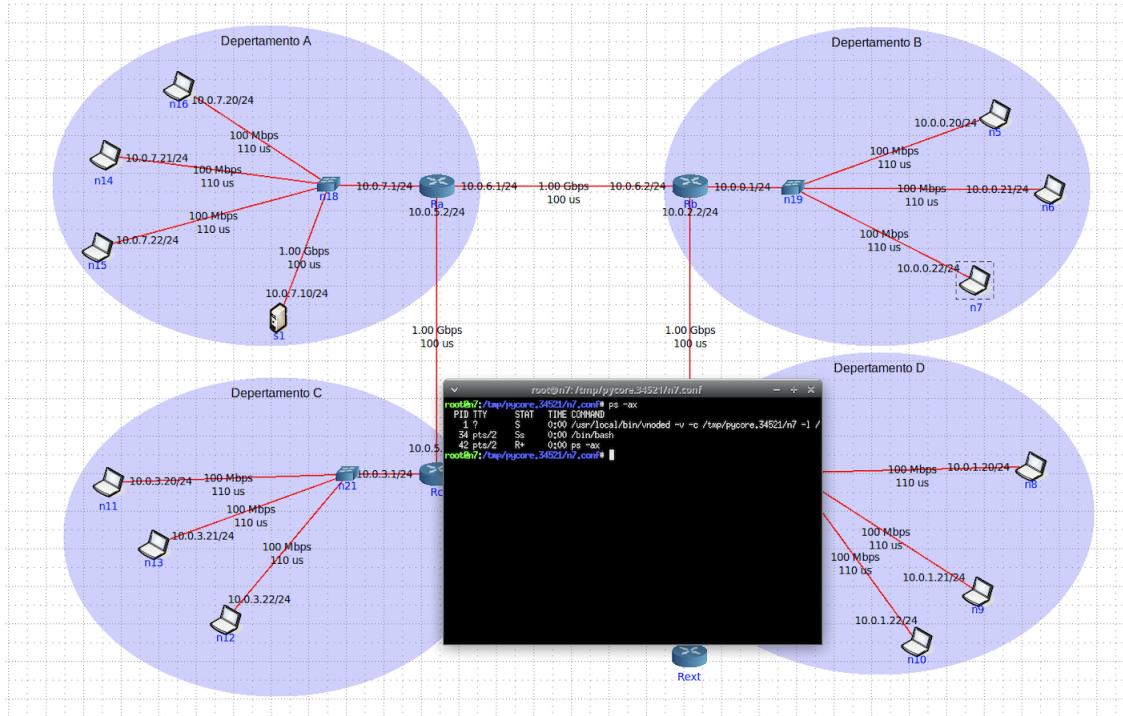


Figura 1.21: Processos a correr no laptop n7

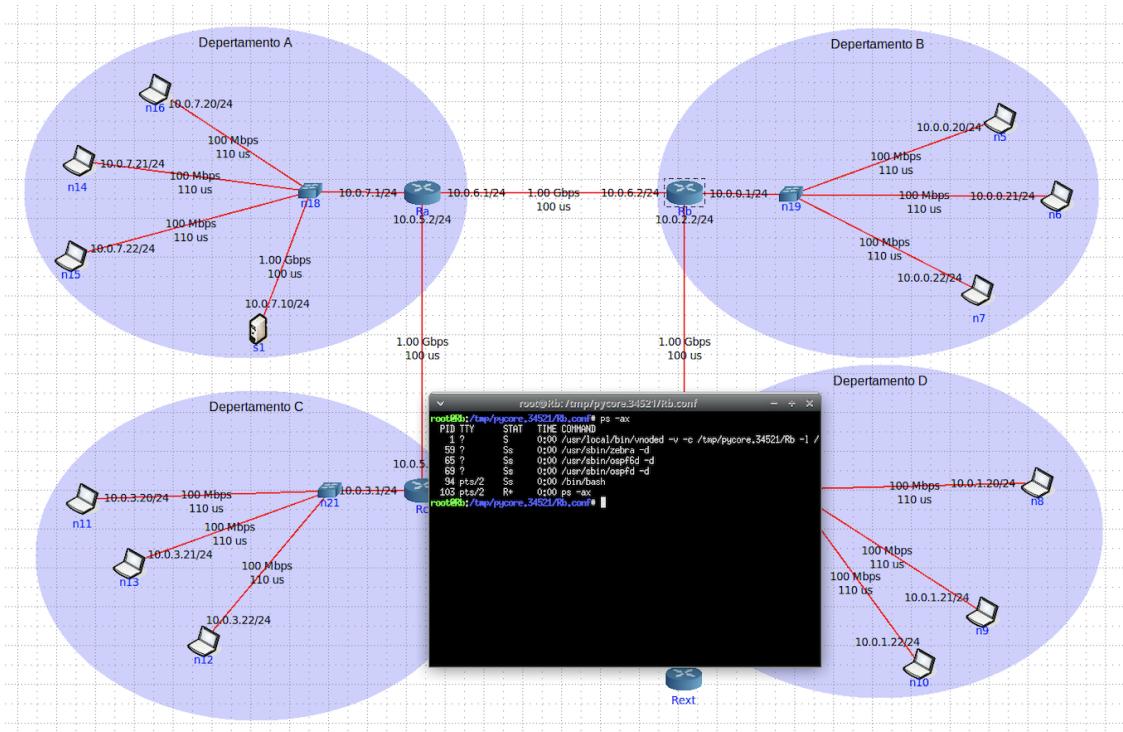


Figura 1.22: Processos a correr no router Rb

Na figura 1.22 verifica-se que na coluna Command é utilizado o protocolo ospfd, o que significa que está a ser usado encaminhamento dinâmico para o router Rb (os pacotes podem seguir por diferentes rotas quando não é possível seguir a rota esperada). Na figura 1.21 podemos ver que não é utilizado nenhum protocolo, logo está a ser usado encaminhamento estático para o laptop n7.

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

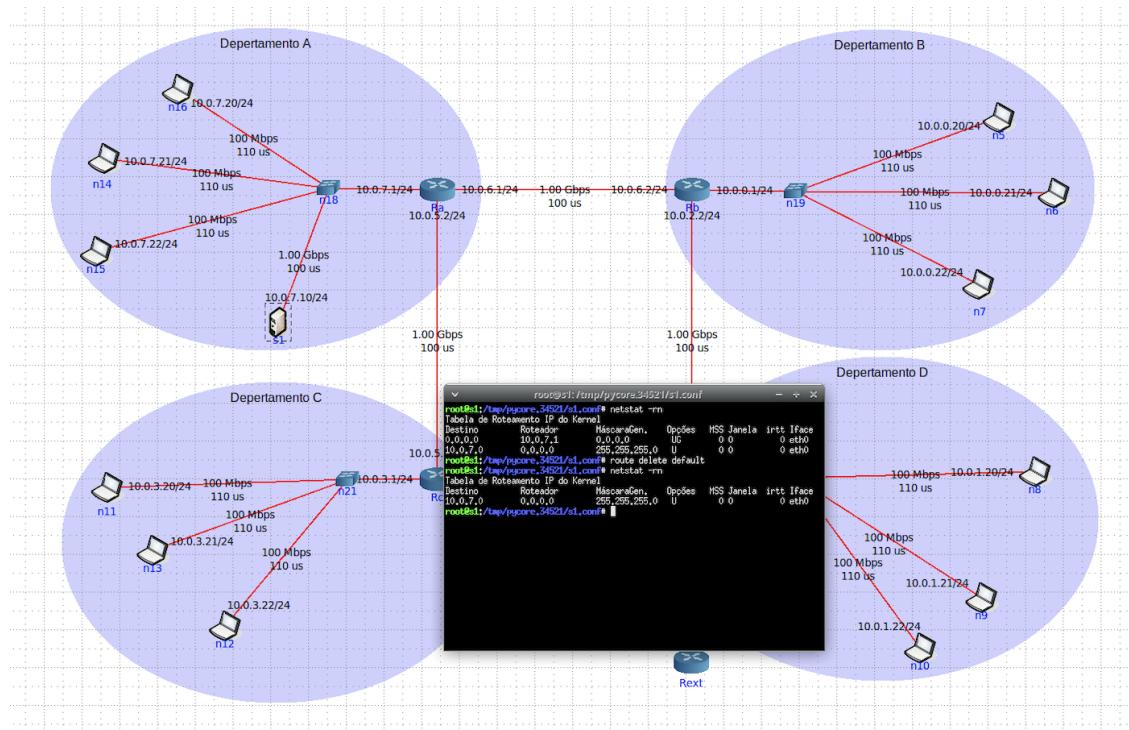


Figura 1.23: Tabela de encaminhamento do servidor S1 antes e depois de ser removida a rota default

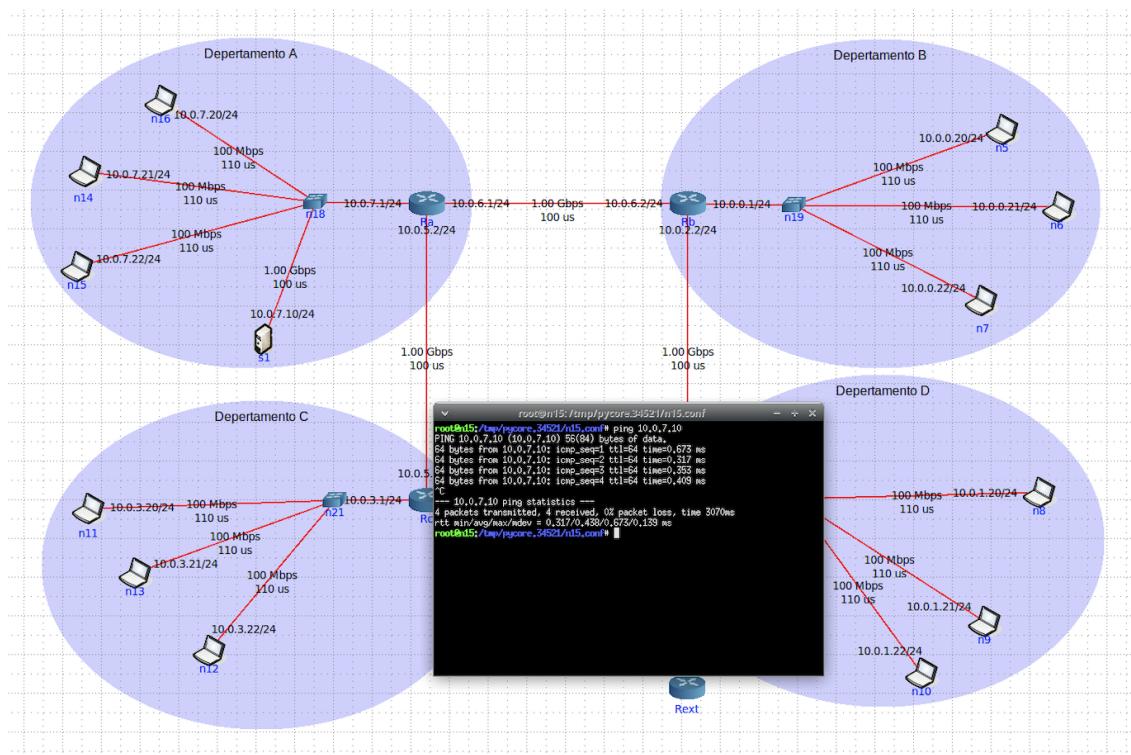
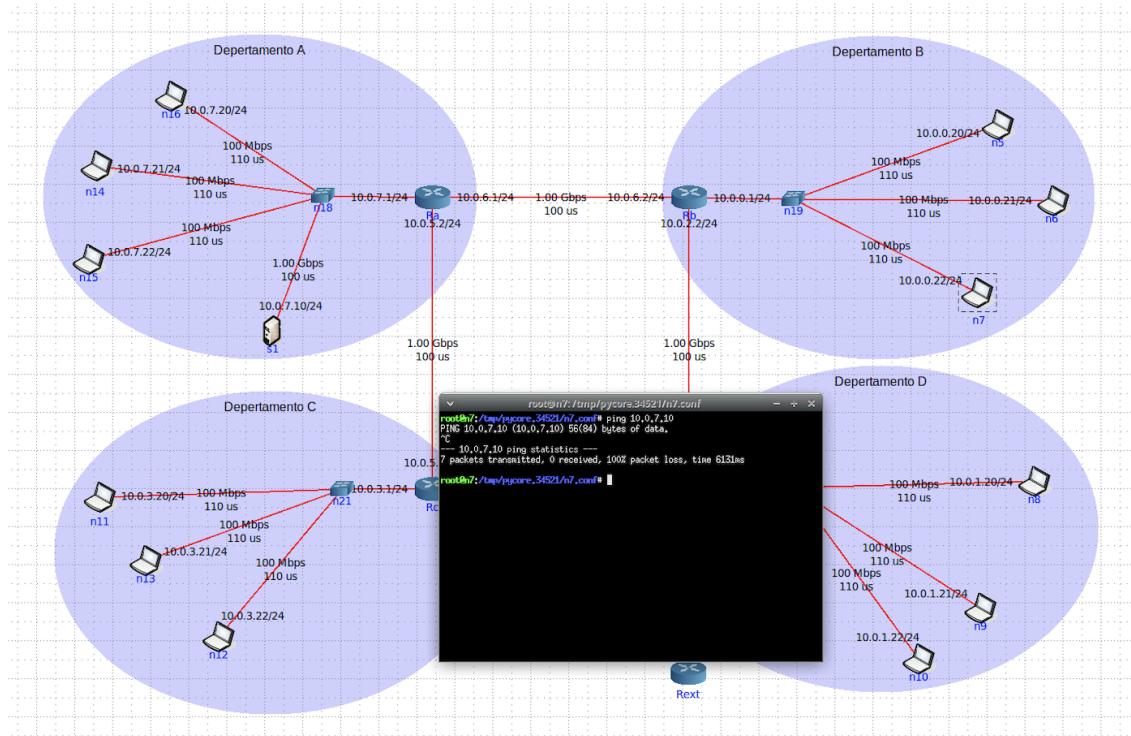


Figura 1.24: Conetividade entre servidor S1 e laptop do Departamento A



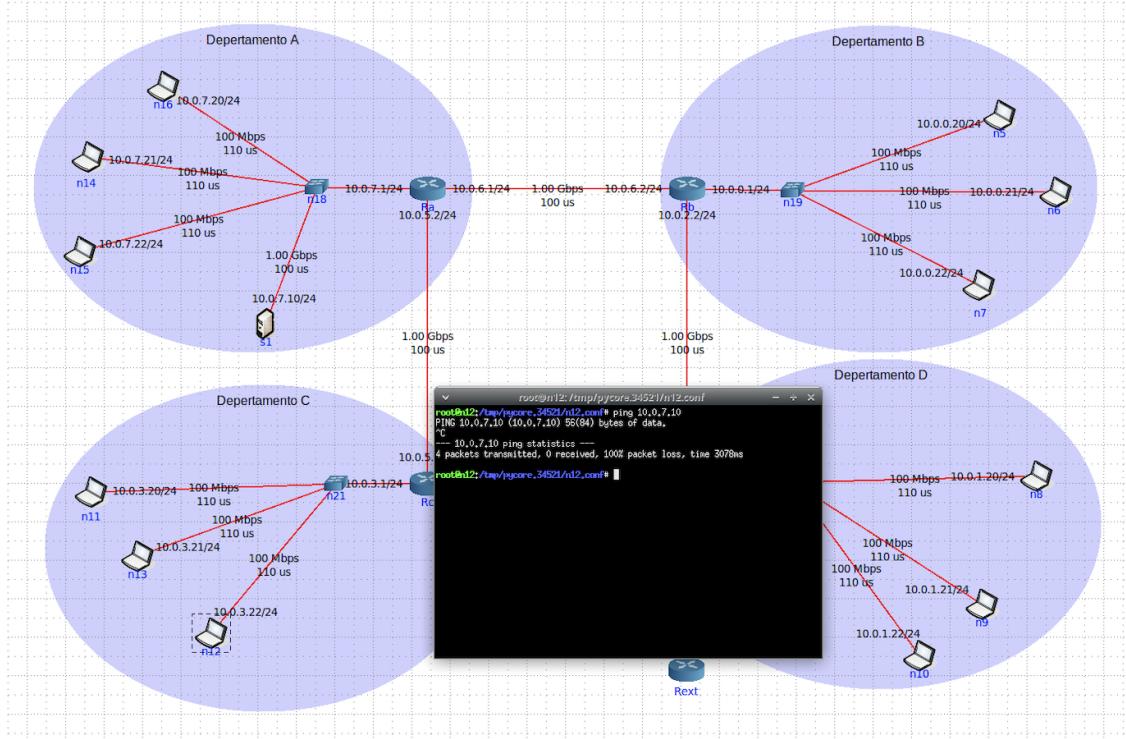


Figura 1.26: Conetividade entre servidor S1 e laptop do Departamento C

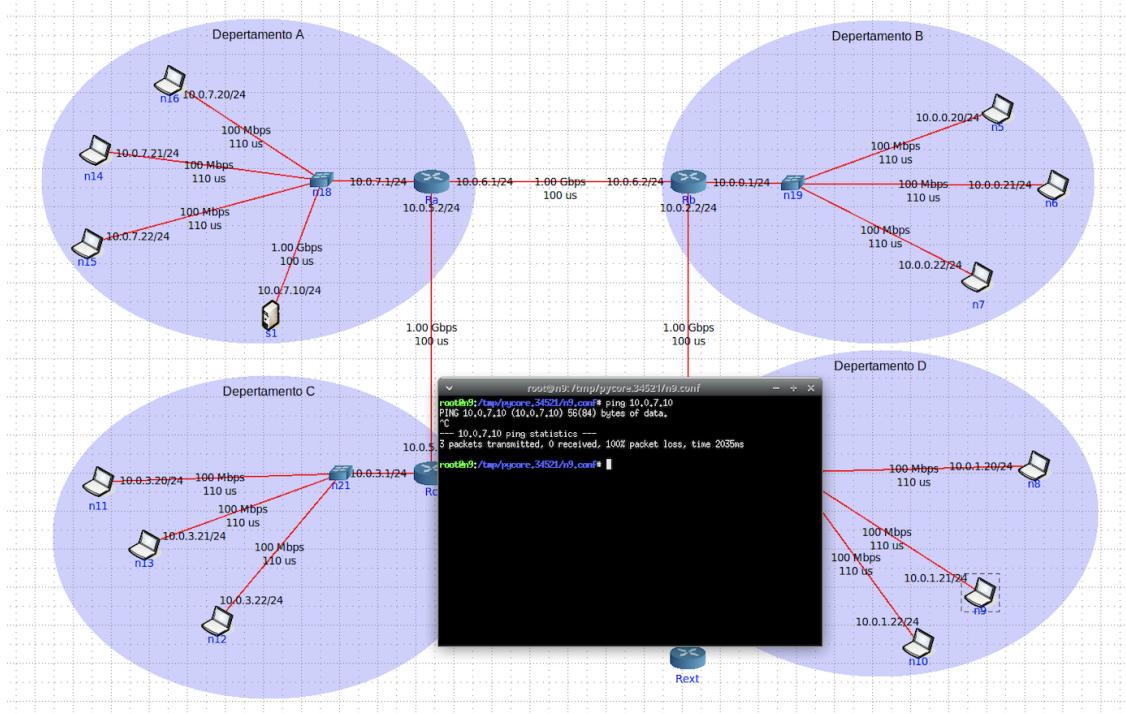


Figura 1.27: Conetividade entre servidor S1 e laptop do Departamento D

Nas figuras anteriores verificamos que, após remover a rota default, apenas os laptops do mesmo departamento do servidor S1 têm conectividade com este, visto que nos laptops de outros departamentos nenhum dos pacotes enviados ao servidor é recebido.

- (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.**

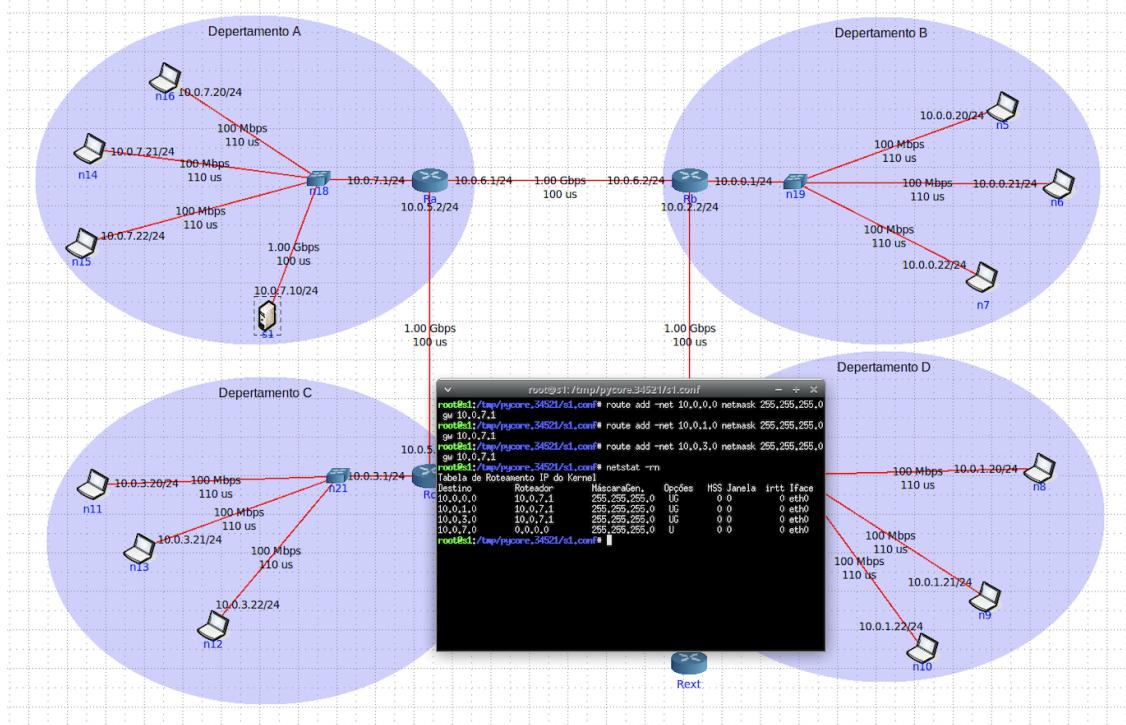


Figura 1.28: Tabela de encaminhamento do servidor S1 depois de serem adicionadas as rotas

Os pacotes com Destino a sub-rede de endereço 10.0.0.0 (sub-rede do Departamento B) e máscara /24 passam pelo Roteador com endereço 10.0.7.1 (router Ra).

Os pacotes com Destino a sub-rede de endereço 10.0.1.0 (sub-rede do Departamento D) e máscara /24 passam pelo Roteador com endereço 10.0.7.1 (router Ra).

Os pacotes com Destino a sub-rede de endereço 10.0.3.0 (sub-rede do Departamento C) e máscara /24 passam pelo Roteador com endereço 10.0.7.1 (router Ra).

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

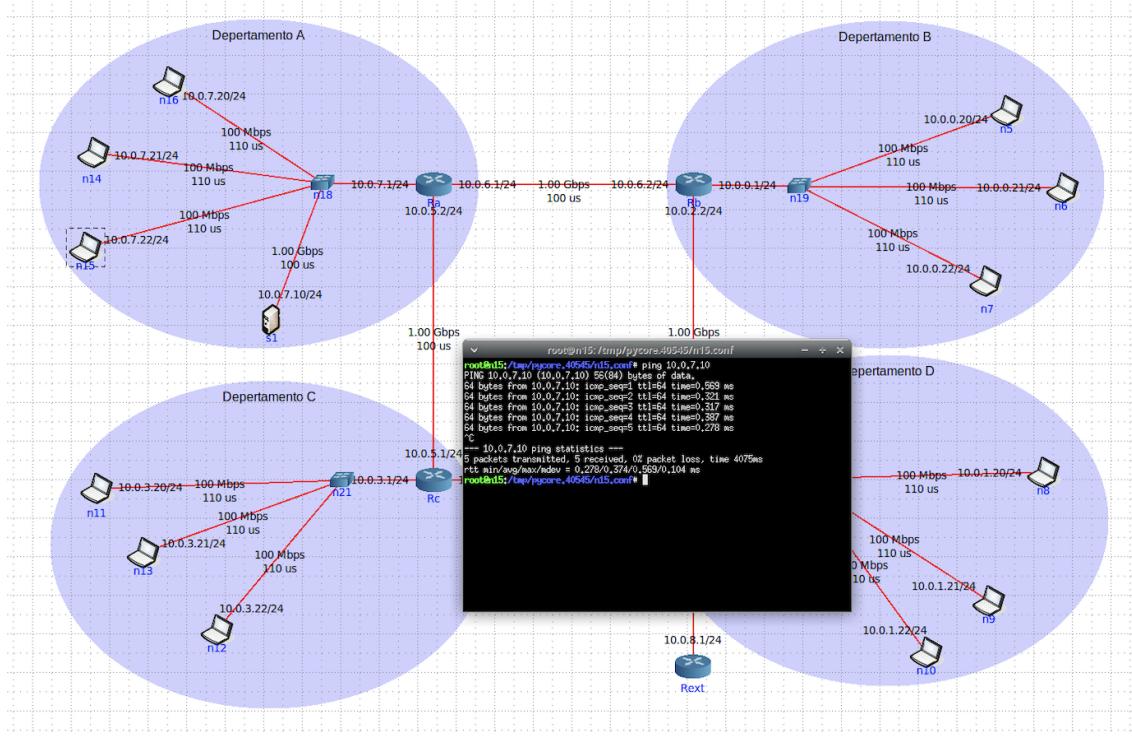


Figura 1.29: Conetividade entre servidor S1 e laptop do Departamento A

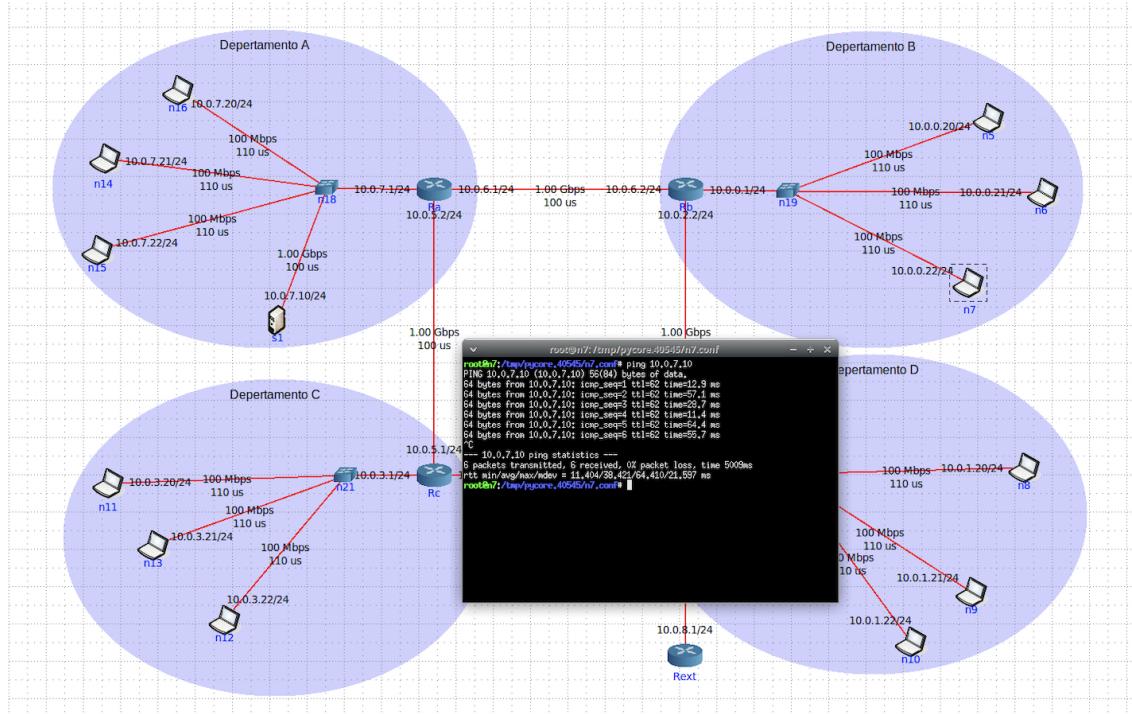


Figura 1.30: Conetividade entre servidor S1 e laptop do Departamento B

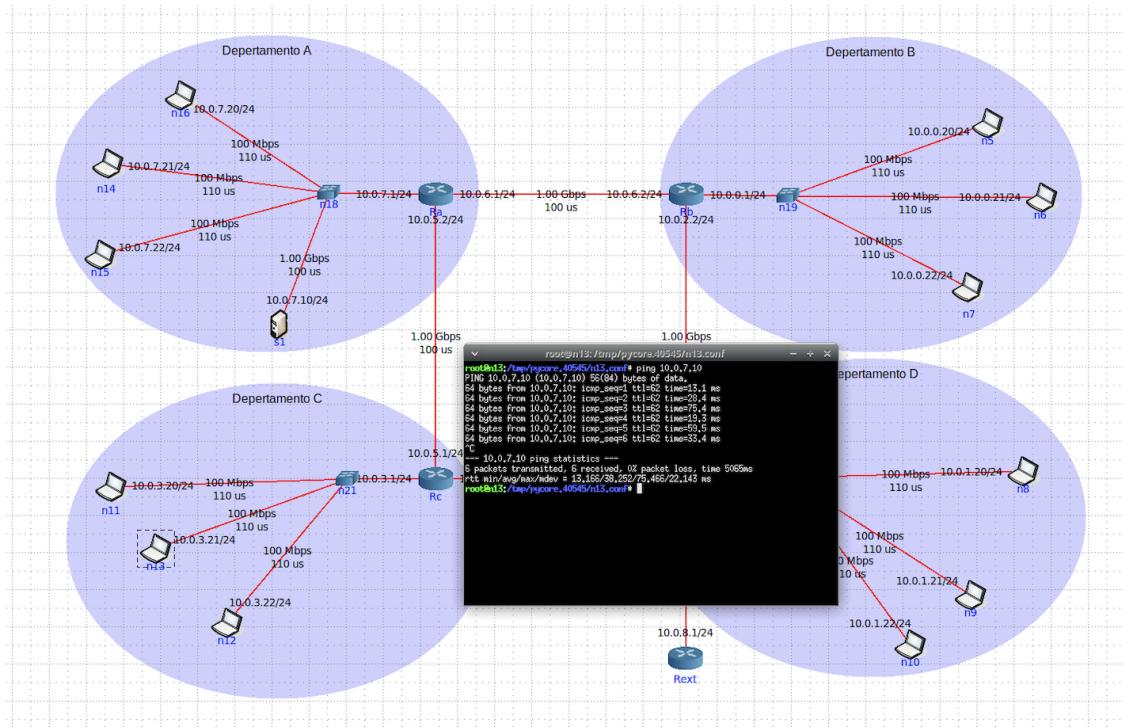


Figura 1.31: Conetividade entre servidor S1 e laptop do Departamento C

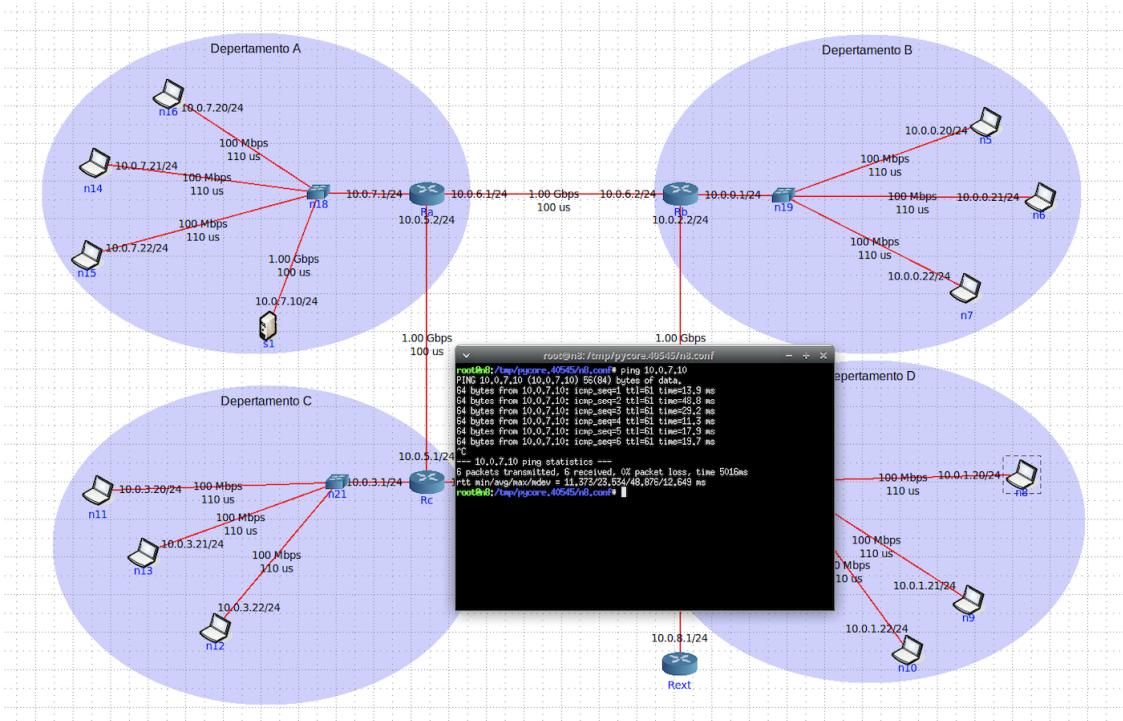


Figura 1.32: Conetividade entre servidor S1 e laptop do Departamento D

Pelas figuras anteriores podemos ver que já existe conectividade entre os laptops de todos os departamentos e o servidor S1.

3. Definição de Sub-redes

- (a) Considere que dispõe apenas do endereço de rede IP 172.yyx.32.0/20, em que “yy” são os dígitos correspondendo ao seu número de grupo (Gyy) e “x” é o dígito correspondente ao seu turno prático (PLx). Defina um novo esquema de endereçamento para as redes dos departamentos (mantendo a rede de acesso e core inalteradas) e atribua endereços às interfaces dos vários sistemas envolvidos. Deve justificar as opções usadas.

Estendendo o endereço 172.86.32.0/20 ficamos com 172.86.00100000.0/20 sendo o identificador de rede 172.86.0010.

Não podemos usar 2 bits para subnetting pois o número de sub-redes seria $2^2 - 2 = 2 < 4$. Usamos então 3 bits subnetting pois o número de sub-redes possíveis é $2^3 - 2 = 6 \geq 4$.

Ficamos então com os seguintes endereços:

172.86.0010₀₀₁0.0/23 → 172.86.34.0/23

172.86.0010₀₁₀0.0/23 → 172.86.36.0/23

172.86.0010₀₁₁0.0/23 → 172.86.38.0/23

172.86.0010₁₀₀0.0/23 → 172.86.40.0/23

172.86.0010₁₀₁0.0/23 → 172.86.42.0/23

172.86.0010₁₁₀0.0/23 → 172.86.44.0/23

Os endereços:

172.86.0010₀₀₀0.0/23 → 172.86.32.0/23

172.86.0010₁₁₁0.0/23 → 172.86.46.0/23

Não podem ser utilizados porque estão reservados.

* label amarelo → ids de sub-rede possíveis

Arbitrariamente decidimos as seguintes associações endereço-departamento:

172.86.34.0/23 → departamento A

172.86.36.0/23 → departamento B

172.86.38.0/23 → departamento C

172.86.40.0/23 → departamento D

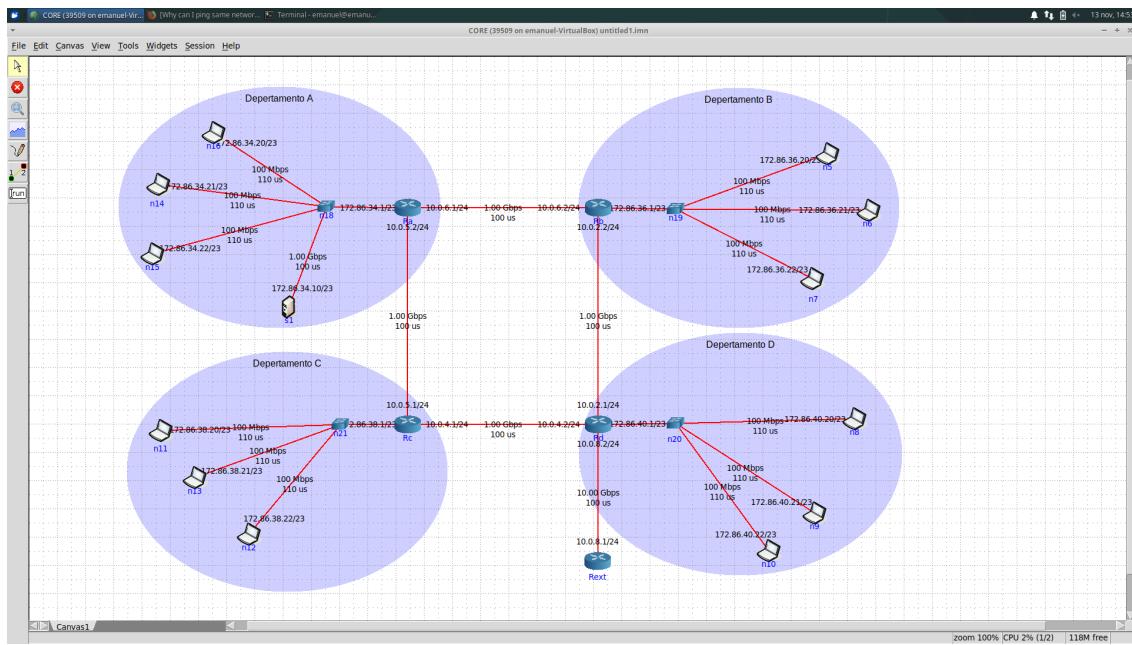


Figura 1.33: Topologia core com os endereços alterados

- (e) Qual a máscara de rede que usou (em notação decimal)? Quantos interfaces IP pode interligar em cada departamento? Justifique.

A máscara que utilizamos foi /23 (notação CIDR) que corresponde a 255.255.254.0 em formato decimal, visto que os primeiros 23 bits são da parte da subnet e os 9 bits restantes são da parte do host. Com isto podemos ter $2^9 - 2 = 510$ hosts em cada sub-rede (em cada departamento).

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

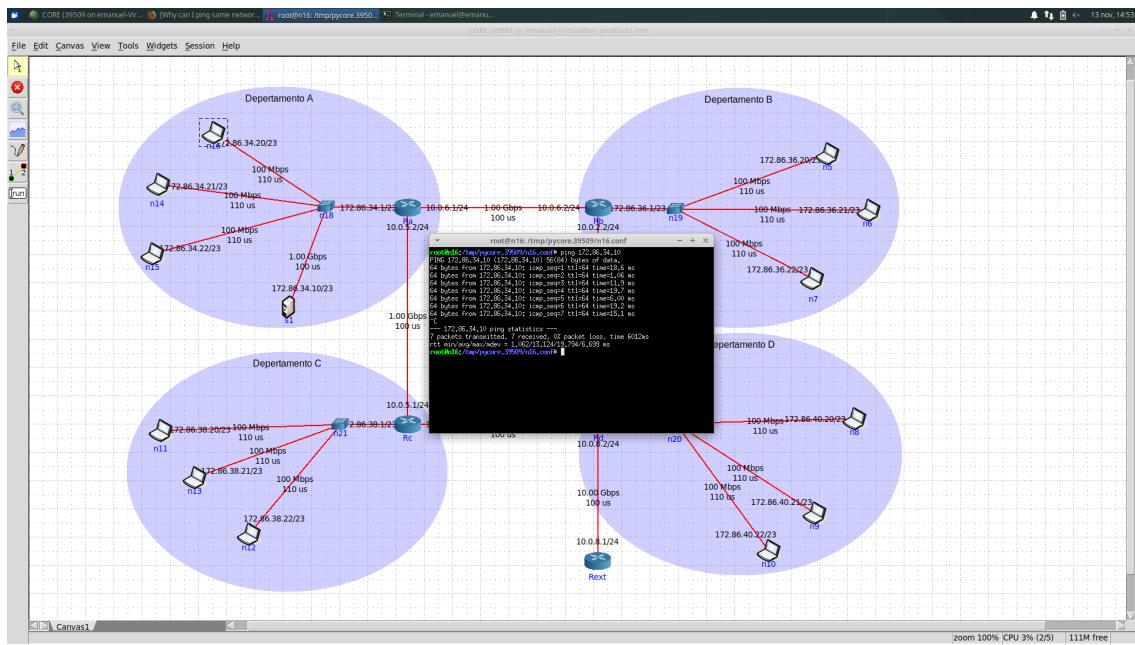


Figura 1.34: Conetividade entre o servidor S1 e laptop do Departamento A

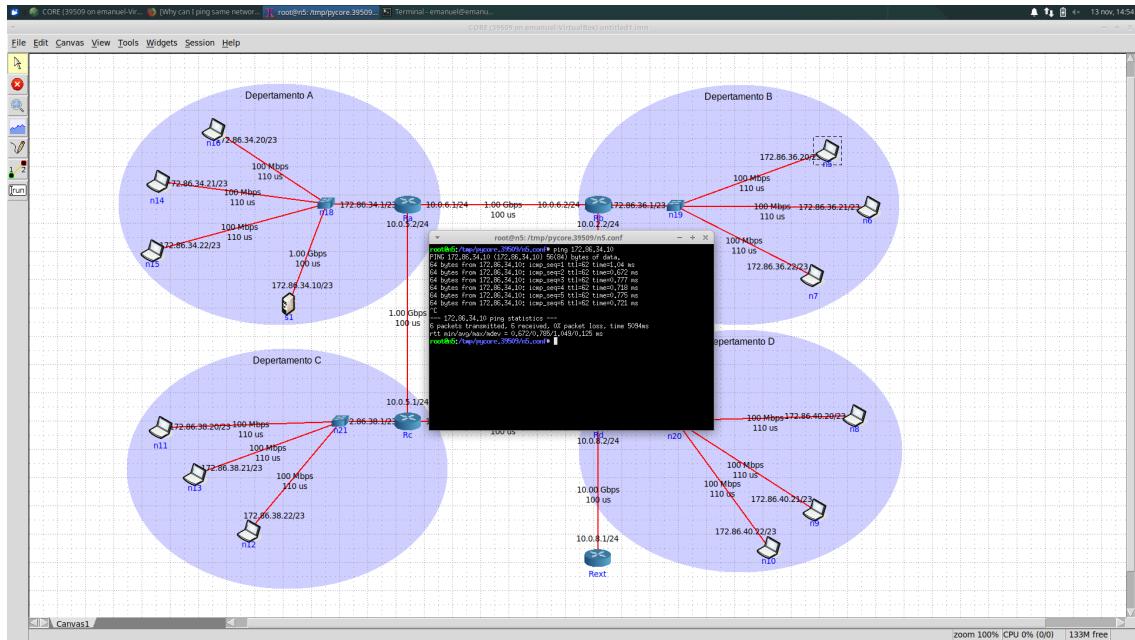


Figura 1.35: Conetividade entre o servidor S1 e laptop do Departamento B

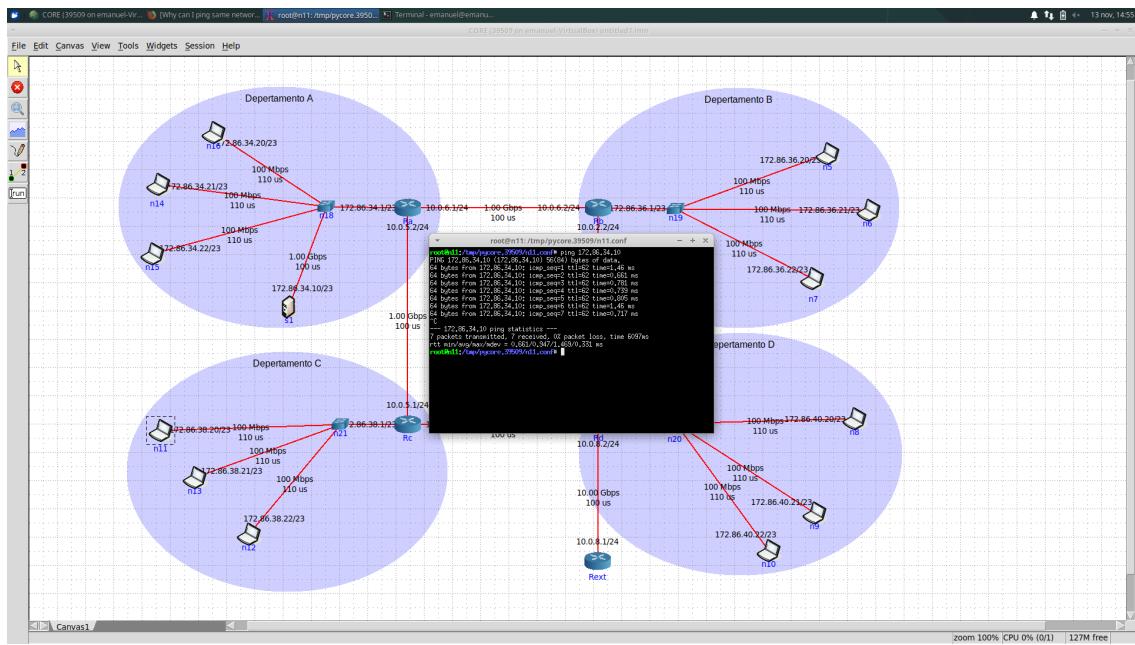


Figura 1.36: Conetividade entre o servidor S1 e laptop do Departamento C

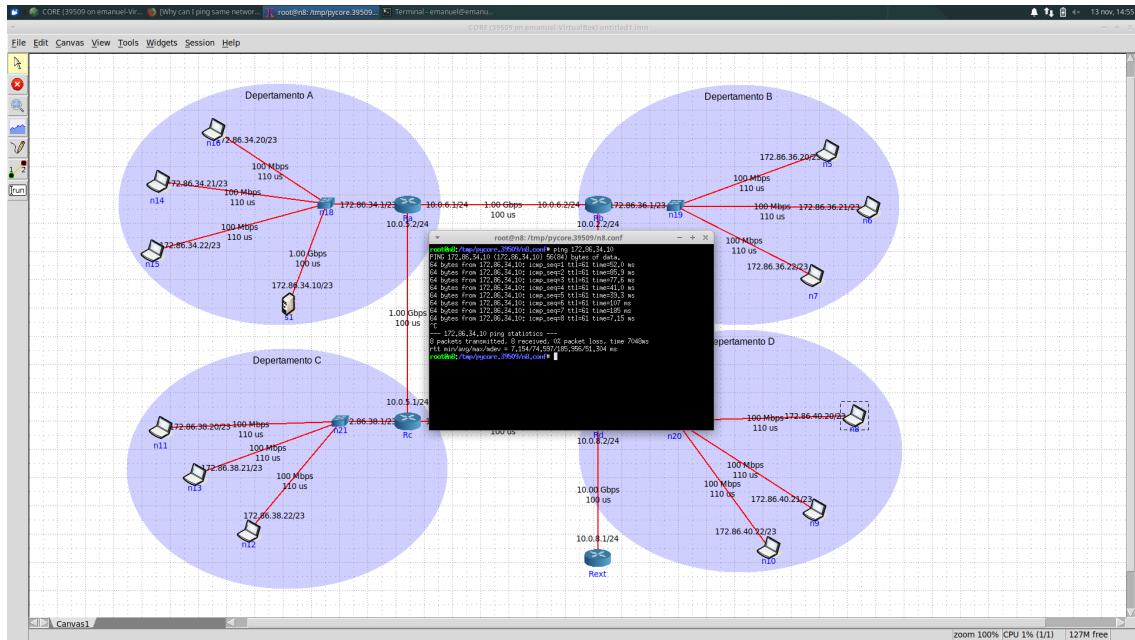


Figura 1.37: Conetividade entre o servidor S1 e laptop do Departamento D

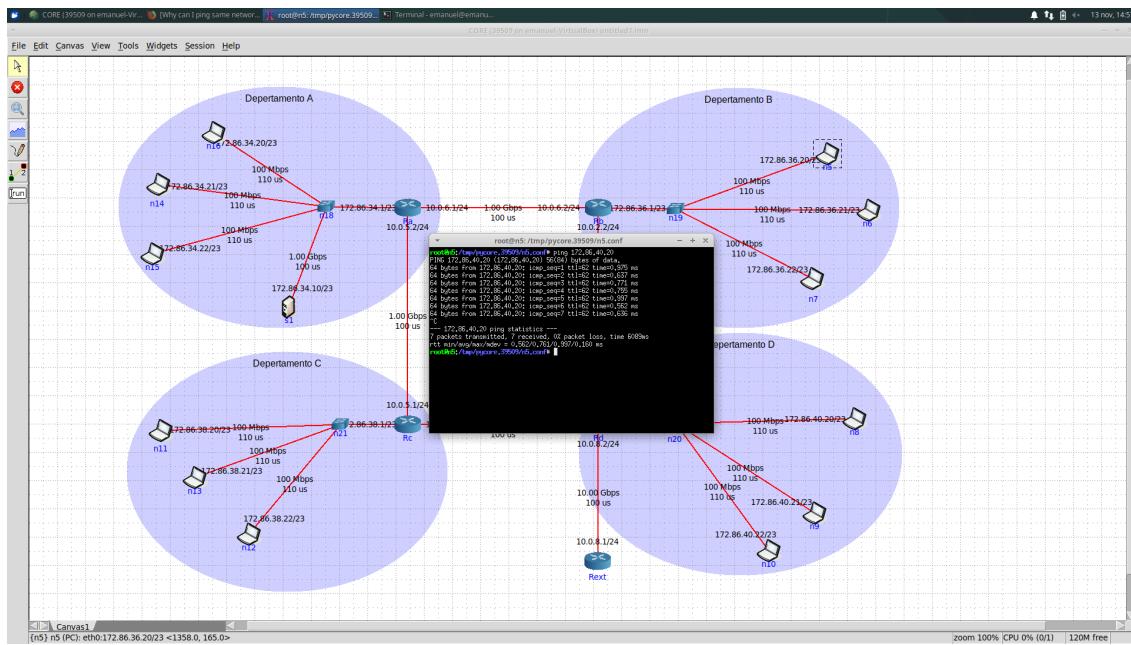


Figura 1.38: Conetividade entre laptop do Departamento B e laptop do Departamento D

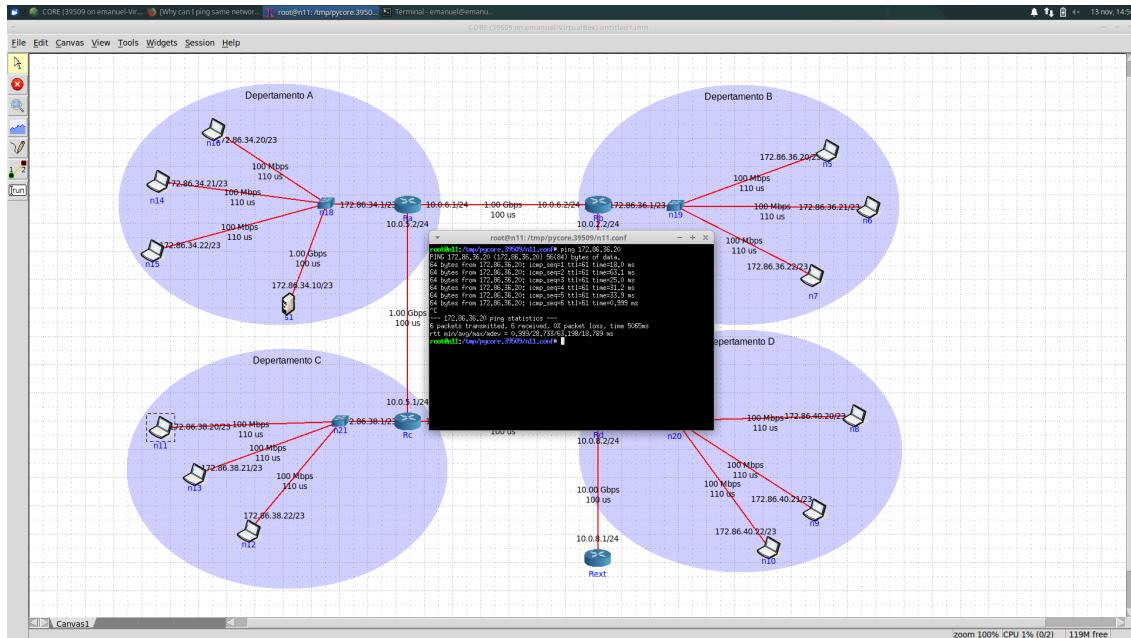


Figura 1.39: Conetividade entre laptop do Departamento B e laptop do Departamento C

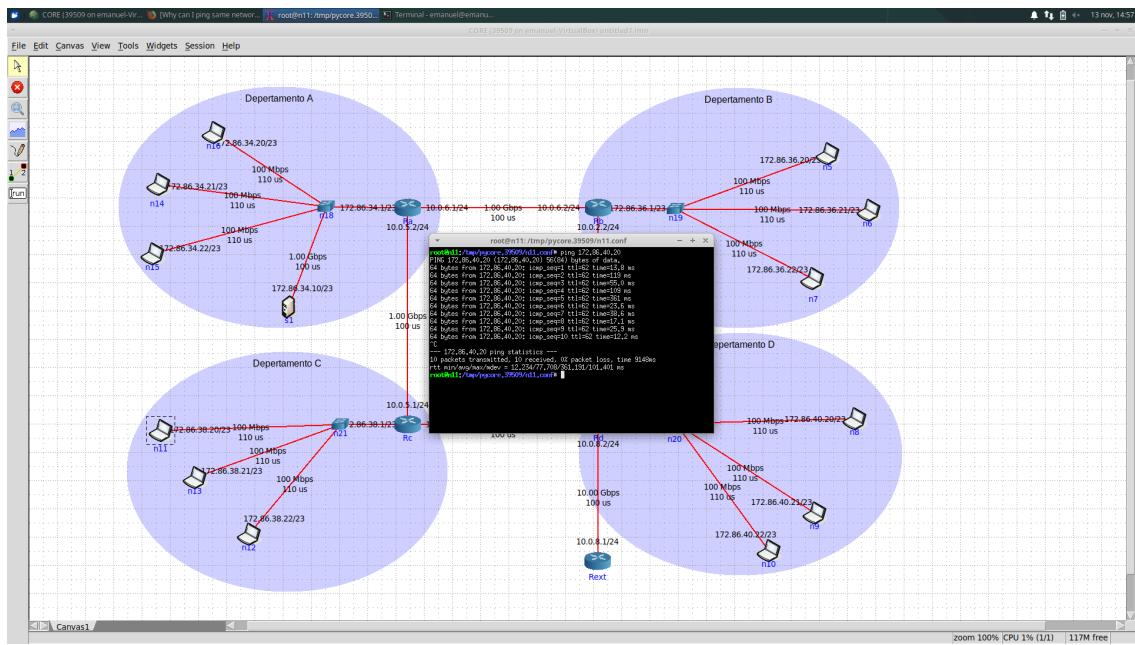


Figura 1.40: Conetividade entre laptop do Departamento C e laptop do Departamento D

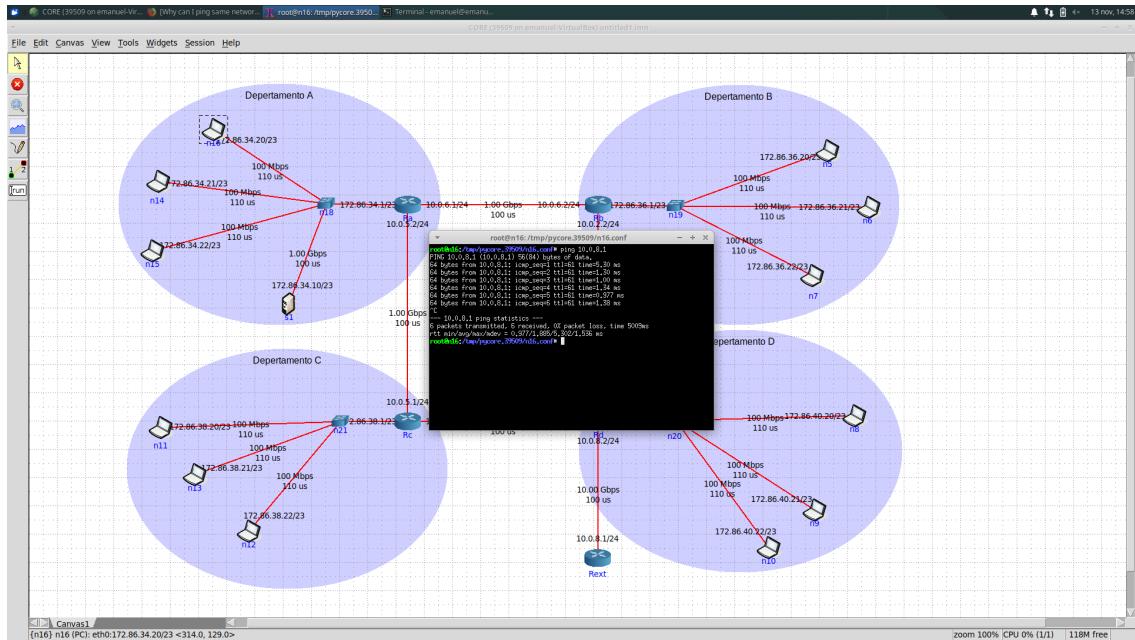


Figura 1.41: Conetividade entre laptop do Departamento A e router Rext

Capítulo 2

Conclusão

Com a realização deste trabalho prático adquirimos conhecimentos sobre a utilização de ferramentas como o Core e o Wireshark para análise de simulações de redes. Aprofundamos os conceitos abordados nas aulas teóricas, nomeadamente o mapeamento de rotas, fragmentação IP, endereçamento e encaminhamento IP e subnetting. Na primeira parte deste estudo é realizado o registo de datagramas IP enviados e recebidos através da execução do programa traceroute. São analisados os vários campos de um datagrama IP e detalhado o processo de fragmentação realizado pelo IP. Na segunda parte deste estudo abordou-se algumas das técnicas propostas para aumentar a escalabilidade do protocolo IP, mitigar a exaustão dos endereços IPv4 e reduzir os recursos de memória necessários nos routers para manter as tabelas de encaminhamento. As nossas maiores dificuldades foram a adaptação às ferramentas, particularmente o Core e o Wireshark, na primeira parte, e a parte da sub-rede, na segunda parte deste trabalho. Apesar das dificuldades acreditamos que alcançamos os objetivos propostos.