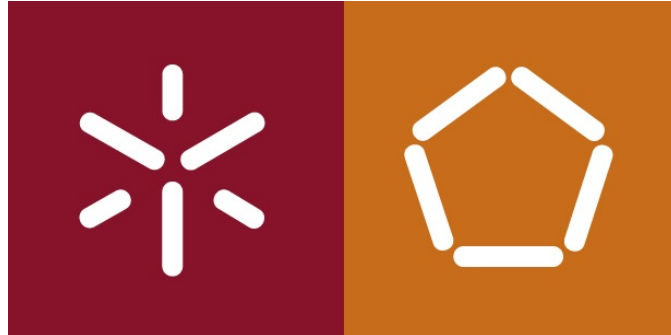


UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA



Redes de Computadores

RELATÓRIO DO TRABALHO PRÁTICO 2

PROTOCOLO IP

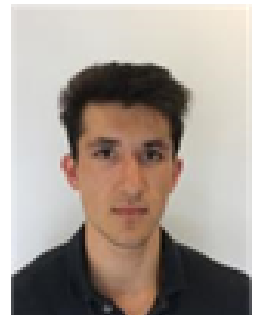
GRUPO 1



Adriana Meireles
A82582



Nuno Silva
A78156



Shahzod Yusupov
A82617

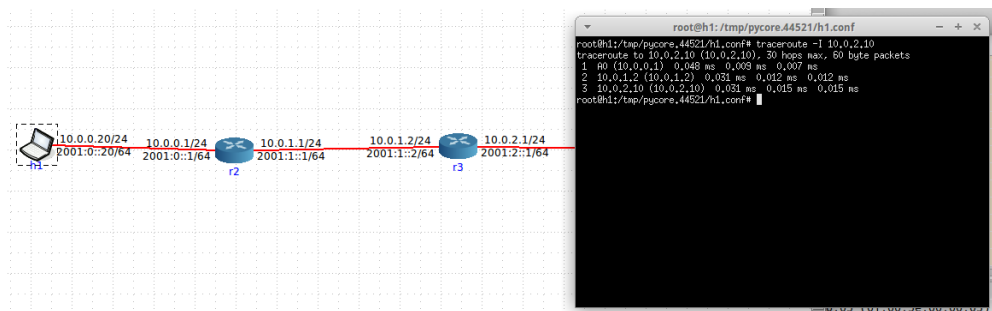
March 22, 2020

Parte I

Captura de Tráfego IP

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). Ajuste o nome dos equipamentos atribuídos por defeito para a topologia do enunciado.

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



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

Após fazer o comando “`traceroute -I 10.0.2.10`” a partir do host h1, são-nos dados 3 tempos (cada um correspondente a um pacote que tem TTL igual aos outros dois) de cada vez que são enviados 3 novos pacotes para o host h4. No início de cada linha temos o endereço de cada router sendo que o último endereço que aparece corresponde ao host h4. O host h1 tenta comunicar com o host servidor(h4) enviando pacotes com TTL=1, TTL=2 e TTL=3, sequencialmente. Os pacotes com TTL=1 e TTL=2 não chegam até ao h4 sendo descartados no router r2 e r3, respetivamente. O ICMP envia uma mensagem de erro(102 Time-To-Live Exceeded(Time to Live Exceeded in Transit)).

No.	Time	Source	Destination	Protocol	Length	Info
2.1.048818		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3779/49934, ttl=255 (reply in 3)
3.1.049820		marco.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	170	Echo (ping) reply id=0x0001, seq=3779/49934, ttl=62 (request in 2)
4.1.059559		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3780/50190, ttl=1 (no response found!)
5.1.099968		gv.sa.di.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
6.1.149039		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3781/50446, ttl=2 (no response found!)
7.1.150098		cisco.di.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
8.1.199063		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3782/50702, ttl=3 (reply in 9)
9.1.200103		marco.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	170	Echo (ping) reply id=0x0001, seq=3782/50702, ttl=62 (request in 8)
11.1.971488		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3783/50958, ttl=255 (reply in 12)

Ethernet II, Src: AsustekC_31:58:f7 (2c:4d:54:31:58:f7), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)	
Destination: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)	
Source: AsustekC_31:58:f7 (2c:4d:54:31:58:f7)	
Type: IPv4 (0x0800)	
Internet Protocol Version 4, Src: DESKTOP-VAS6812.local (192.168.100.215), Dst: marco.uminho.pt (193.136.9.240)	
0100 = Version: 4	
... 0101 = Header Length: 20 bytes (5)	
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	
Total Length: 156	
Identification: 0x2bd7 (11223)	
Flags: 0x0000	
Time to live: 255	
Protocol: ICMP (1)	
Header checksum: 0x0000 [validation disabled]	
[Header checksum status: Unverified]	
Source: DESKTOP-VAS6812.local (192.168.100.215)	
Destination: marco.uminho.pt (193.136.9.240)	
Internet Control Message Protocol	
Type: 8 (Echo (ping) request)	
Code: 0	
Checksum: 0xe133 [correct]	
[Checksum Status: Good]	
Identifier (BE): 1 (0x0001)	
Identifier (LE): 256 (0x0100)	
Sequence number (BE): 3779 (0x0ec3)	
Sequence number (LE): 49934 (0xc30e)	
[Response frame: 3]	
Data (128 bytes)	

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

O valor do protocolo é 1, que 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 reservados para o cabeçalho. O campo de dados terá tantos bytes quantos forem a diferença entre o número total de bytes do datagrama e o cabeçalho do datagrama. Portanto, para o payload estão reservados (156-20=) 136 bytes.

d.O datagrama IP foi fragmentado? Justifique.

No.	Time	Source	Destination	Protocol	Length	Info
2.1.048818		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3779/49934, ttl=255 (reply in 3)
3.1.049820		marco.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	170	Echo (ping) reply id=0x0001, seq=3779/49934, ttl=62 (request in 2)
4.1.059559		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3780/50190, ttl=1 (no response found!)
5.1.099968		gv.sa.di.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
6.1.149039		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3781/50446, ttl=2 (no response found!)
7.1.150098		cisco.di.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
8.1.199063		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3782/50702, ttl=3 (reply in 9)
9.1.200103		marco.uminho.pt	DESKTOP-VAS6812.loc...	ICMP	170	Echo (ping) reply id=0x0001, seq=3782/50702, ttl=62 (request in 8)
11.1.971488		DESKTOP-VAS6812.loc...	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3783/50958, ttl=255 (reply in 12)

Frame 2: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface 0	
Ethernet II, Src: AsustekC_31:58:f7 (2c:4d:54:31:58:f7), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)	
Destination: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)	
Source: AsustekC_31:58:f7 (2c:4d:54:31:58:f7)	
Type: IPv4 (0x0800)	
Internet Protocol Version 4, Src: DESKTOP-VAS6812.local (192.168.100.215), Dst: marco.uminho.pt (193.136.9.240)	
0100 = Version: 4	
... 0101 = Header Length: 20 bytes (5)	
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	
Total Length: 156	
Identification: 0x2bd7 (11223)	
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: 0x0000 [validation disabled]	
[Header checksum status: Unverified]	
Source: DESKTOP-VAS6812.local (192.168.100.215)	
Destination: marco.uminho.pt (193.136.9.240)	
Internet Control Message Protocol	
Type: 8 (Echo (ping) request)	
Code: 0	
Checksum: 0xe133 [correct]	
[Checksum Status: Good]	
Identifier (BE): 1 (0x0001)	
Identifier (LE): 256 (0x0100)	

O datagrama não foi fragmentado. Esta conclusão deve-se ao facto de no cabeçalho da primeira mensagem ICMP no indicador "flags" existir a flag "Fragment offset" que nos indica a posição de um fragmento relativamente aos outros que devidamente agrupados constituem um datagrama.

Neste caso o "Fragment offset" está a 0, pelo que se conclui que se existirem mais fragmentos, este é o primeiro. Para além disso, temos a flag "More Fragments" que nos indica se há ou não mais fragmentos para além do atual. Tendo a flag valor 1, existem mais fragmentos, tendo-a 0, não existem. Neste caso o valor é 0(não existem), e sendo o primeiro fragmento e não havendo mais, o fragmento é, na verdade, o datagrama original.

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
2	1.048818	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3779/49934, ttl=255 (reply in 3)
4	1.059859	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3780/50190, ttl=1 (no response found!)
6	1.149039	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3781/50446, ttl=2 (no response found!)
8	1.199063	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3782/50702, ttl=3 (reply in 9)
11	1.971488	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3783/50958, ttl=255 (reply in 12)
13	2.022539	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3784/51214, ttl=1 (no response found!)
15	2.072671	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3785/51470, ttl=2 (no response found!)
17	2.123685	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3786/51726, ttl=3 (reply in 18)
21	3.549889	DESKTOP-VAS6B12.local	marco.uminho.pt	ICMP	170	Echo (ping) request id=0x0001, seq=3787/51982, ttl=255 (reply in 22)

Destination: Vmware_d2:19:f0 (08:0c:29:d2:19:f0)
 Source: AsustekC_31:58:f7 (2c:4d:54:31:58:f7)
 Type: IPv4 (0x0800)
 Internet Protocol Version 4, Src: DESKTOP-VAS6B12.local (192.168.100.215), Dst: marco.uminho.pt (193.136.9.240)
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 156
 Identification: 0x2bd7 (11223)
 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: 0x0000 [validation disabled]
 [Header checksum status: Unverified]
 Source: DESKTOP-VAS6B12.local (192.168.100.215)
 Destination: marco.uminho.pt (193.136.9.240)
 Internet Control Message Protocol
 Type: 8 (Echo (ping) request)
 Code: 0
 Checksum: 0xe133 [correct]
 [Checksum Status: Good]
 Identifier (BE): 1 (0x0001)
 Identifier (LE): 256 (0x0100)
 Sequence number (BE): 3779 (0x0ec3)

Os campos que variam de pacote para pacote são o campo de identificação do datagrama e o TTL.

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

Relativamente ao campo de identificação é incrementado 1, o TTL tem o seguinte padrão: 1, 2, 3, 255.

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?

O valor do campo TTL é constante dependendo da source:

- Se a source for *gw.sa.di.uminho.pt*, o valor do TTL é 64;
- Se a source for *cisco.di.uminho.pt*, o valor do TTL é 254;

No.	Time	Source	Destination	Protocol	Length	Info
3	1.049820	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3779/49934, ttl=62 (request in 2)
5	1.099968	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
7	1.150098	cisco.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
9	1.200103	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3782/50702, ttl=62 (request in 8)
12	1.972573	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3783/50958, ttl=62 (request in 11)
14	2.023550	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
16	2.073729	cisco.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18	2.124761	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3786/51726, ttl=62 (request in 17)
22	3.550903	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3787/51982, ttl=62 (request in 21)
25	3.601761	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
27	3.652417	cisco.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
29	3.702561	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3790/52750, ttl=62 (request in 28)
31	4.474430	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3791/53006, ttl=62 (request in 30)
33	4.525366	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)

Destination: AsustekC_31:58:f7 (2c:d4:54:31:58:f7)
Source: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
Type: IPv4 (0x0000)
Internet Protocol Version 4, Src: gw.sa.di.uminho.pt (192.168.100.254), Dst: DESKTOP-VAS6812.local (192.168.100.215)
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
Total Length: 184
Identification: 0xf759 (63321)
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: 64
Protocol: ICMP (1)
Header checksum: 0x3705 [validation disabled]
[Header checksum status: Unverified]
Source: gw.sa.di.uminho.pt (192.168.100.254)
Destination: DESKTOP-VAS6812.local (192.168.100.215)

No.	Time	Source	Destination	Protocol	Length	Info
3	1.049820	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3779/49934, ttl=62 (request in 2)
5	1.099968	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
7	1.150098	cisco.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
9	1.200103	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3782/50702, ttl=62 (request in 8)
12	1.972573	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3783/50958, ttl=62 (request in 11)
14	2.023550	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
16	2.073729	cisco.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18	2.124761	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3786/51726, ttl=62 (request in 17)
22	3.550903	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3787/51982, ttl=62 (request in 21)
25	3.601761	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)
27	3.652417	cisco.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
29	3.702561	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3790/52750, ttl=62 (request in 28)
31	4.474430	marco.uminho.pt	DESKTOP-VAS6812.local	ICMP	170	Echo (ping) reply id=0x0001, seq=3791/53006, ttl=62 (request in 30)
33	4.525366	gw.sa.di.uminho.pt	DESKTOP-VAS6812.local	ICMP	198	Time-to-live exceeded (Time to live exceeded in transit)

Destination: AsustekC_31:58:f7 (2c:d4:54:31:58:f7)
Source: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
Type: IPv4 (0x0000)
Internet Protocol Version 4, Src: cisco.di.uminho.pt (193.136.19.254), Dst: DESKTOP-VAS6812.local (192.168.100.215)
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
Total Length: 56
Identification: 0xb582 (46466)
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: 254
Protocol: ICMP (1)
Header checksum: 0x0b7c [validation disabled]
[Header checksum status: Unverified]
Source: cisco.di.uminho.pt (193.136.19.254)
Destination: DESKTOP-VAS6812.local (192.168.100.215)

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 3531 bytes.

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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.215	gw.sa.di.uminho.pt	NBNS	110	Refresh NB WORKGROUP<0>
2	0.102932	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4367) [Reassembled in #4]
3	0.102941	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4367) [Reassembled in #4]
4	0.102948	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9811/21286, ttl=255 (reply in 7)
5	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3e5e) [Reassembled in #7]
6	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3e5e) [Reassembled in #7]
7	0.105807	marco.uminho.pt	192.168.100.215	ICMP	585	Echo (ping) reply id=0x0001, seq=9811/21286, ttl=62 (request in 4)
8	0.153082	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4368) [Reassembled in #10]
9	0.153091	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4368) [Reassembled in #10]
10	0.153117	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9812/21542, ttl=1 (no response found)

Ethernet II, Src: AsustekC_31:58:f7 (2c:d4:54:31:58:f7), Dst: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0)
Destination: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0)
Source: AsustekC_31:58:f7 (2c:d4:54:31:58:f7)
Type: IPv4 (0x0000)
Internet Protocol Version 4, Src: 192.168.100.215 (192.168.100.215), Dst: marco.uminho.pt (193.136.9.240)
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 571
Identification: 0x4308 (17256)
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: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.100.215 (192.168.100.215)
Destination: marco.uminho.pt (193.136.9.240)
[3 IPv4 Fragments (3511 bytes): #8(1480), #9(1480), #10(551)]
Internet Control Message Protocol
Type: 0 (Echo (ping) request)

Houve uma necessidade de fragmentação do pacote porque era demasiado grande para circular na rede em questão (o tamanho do pacote foi alterado para 3531 bytes). O tamanho máximo do pacote permitido nesta rede é de 1500 bytes.

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?

No.	Time	Source	Destination	Protocol	Length	Info
2	0.102932	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4367) [Reassembled in #4]
3	0.102941	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4367) [Reassembled in #4]
4	0.102948	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9811/21286, ttl=255 (reply in 7)
5	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3e5e) [Reassembled in #7]
6	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3e5e) [Reassembled in #7]
7	0.105807	marco.uminho.pt	192.168.100.215	ICMP	585	Echo (ping) reply id=0x0001, seq=9811/21286, ttl=62 (request in 4)
8	0.153082	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4368) [Reassembled in #10]
9	0.153091	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4368) [Reassembled in #10]
10	0.153117	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9812/21542, ttl=1 (no response found)
11	0.154385	gw.sa.di.uminho.pt	192.168.100.215	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
12	0.203558	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4369) [Reassembled in #16]

Ethernet II, Src: AsustekC_31:58:f7 (2c:4d:54:31:58:f7), Dst: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0)

> Destination: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0)

> Source: AsustekC_31:58:f7 (2c:4d:54:31:58:f7)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.100.215 (192.168.100.215), Dst: marco.uminho.pt (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: 0x4367 (17255)

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: 255

Protocol: ICMP (1)

Header checksum: 0x0000 [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.100.215 (192.168.100.215)

Destination: marco.uminho.pt (193.136.9.240)

Reassembled IPv4 in frame: 4

> Data (1480 bytes)

A flag *"More Fragments"* indica-nos se existem mais fragmentos ou não. Esta flag tem o valor de 1, logo existem mais fragmentos. Para além disso temos a flag *"Fragment Offset"* que nos indica a que parte do pacote corresponde este fragmento, isto é, quando o pacote for reagrupado, a disposição dos fragmentos irá ficar ordenada pela ordem crescente da flag *"Fragment offset"*. A mesma está a 0, logo é o primeiro pacote. O indicador *"Total Length"* mostra-nos o tamanho total do pacote, sendo que temos que lhe tirar o comprimento do *"Header"* que é de 20 bytes. Logo, o tamanho deste datagrama é de 1480 bytes.

c. Imprima o segundo fragmento do datagrama IP original. Que informação do cabeçalho IP indica que não se trata do 1º fragmento? Há mais fragmentos? O que nos permite afirmar isso?

Como foi mencionado anteriormente, é possível observar que um fragmento é o primeiro quando o offset tem o valor 0 e o more fragments a 1. Neste caso, o valor da flag *"Fragment Offset"* é 185 (e, portanto, diferente de 0) quer dizer que este fragmento não é o primeiro. Podemos afirmar também que existem mais fragmentos devido ao *"More Fragments"* encontrar-se a 1.

No.	Time	Source	Destination	Protocol	Length	Info
2	0.102932	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4367) [Reassembled in #4]
3	0.102941	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4367) [Reassembled in #4]
4	0.102948	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9811/21286, ttl=255 (reply in 7)
5	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3e5e) [Reassembled in #7]
6	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3e5e) [Reassembled in #7]
7	0.105807	marco.uminho.pt	192.168.100.215	ICMP	585	Echo (ping) reply id=0x0001, seq=9811/21286, ttl=62 (request in 4)
8	0.153082	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4368) [Reassembled in #10]
9	0.153091	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4368) [Reassembled in #10]
10	0.153117	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9812/21542, ttl=1 (no response found!)
11	0.154385	gw.sa.di.uminho.pt	192.168.100.215	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
12	0.203558	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4369) [Reassembled in #14]

Ethernet II, Src: AsustekC_31:58:f7 (2c:d4:54:31:58:f7), Dst: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0) > Destination: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0) > Source: AsustekC_31:58:f7 (2c:d4:54:31:58:f7) Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.100.215 (192.168.100.215), Dst: marco.uminho.pt (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: 0x4367 (17255) > Flags: 0x20b9, 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: 255 Protocol: ICMP (1) Header checksum: 0x0000 [validation disabled] [Header checksum status: Unverified] Source: 192.168.100.215 (192.168.100.215) Destination: marco.uminho.pt (193.136.9.240) Reassembled IPv4 in frame: 4 > Data (1480 bytes)

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

No.	Time	Source	Destination	Protocol	Length	Info
2	0.102932	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4367) [Reassembled in #4]
3	0.102941	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4367) [Reassembled in #4]
4	0.102948	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9811/21286, ttl=255 (reply in 7)
5	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3e5e) [Reassembled in #7]
6	0.105804	marco.uminho.pt	192.168.100.215	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3e5e) [Reassembled in #7]
7	0.105807	marco.uminho.pt	192.168.100.215	ICMP	585	Echo (ping) reply id=0x0001, seq=9811/21286, ttl=62 (request in 4)
8	0.153082	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4368) [Reassembled in #10]
9	0.153091	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=4368) [Reassembled in #10]
10	0.153117	192.168.100.215	marco.uminho.pt	ICMP	585	Echo (ping) request id=0x0001, seq=9812/21542, ttl=1 (no response found!)
11	0.154385	gw.sa.di.uminho.pt	192.168.100.215	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
12	0.203558	192.168.100.215	marco.uminho.pt	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=4369) [Reassembled in #14]

Ethernet II, Src: AsustekC_31:58:f7 (2c:d4:54:31:58:f7), Dst: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0) > Destination: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0) > Source: AsustekC_31:58:f7 (2c:d4:54:31:58:f7) Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.100.215 (192.168.100.215), Dst: marco.uminho.pt (193.136.9.240) 0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 571 Identification: 0x4367 (17255) > 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: 255 Protocol: ICMP (1) Header checksum: 0x0000 [validation disabled] [Header checksum status: Unverified] Source: 192.168.100.215 (192.168.100.215) Destination: marco.uminho.pt (193.136.9.240) > [3 IPv4 Fragments (3511 bytes): #2(1480), #3(1480), #4(551)] Internet Control Message Protocol

Foram criados 3 fragmentos, contando os anteriormente analisados, mais o fragmento que se deteta em forma de mensagem ICMP. Note-se que todos estes fragmentos têm um número de identificação igual(0x4367) que nos mostra que pertencem todos ao mesmo datagrama. É de esperar que a flag "*Fragment Offset*" deste fragmento seja(185+185=) 370, como se pode verificar na imagem que se encontra em cima. Resta-nos olhar para a flag "*More Fragments*" que está a zero. Daqui se conclui que não existem mais fragmentos do datagrama original.

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.

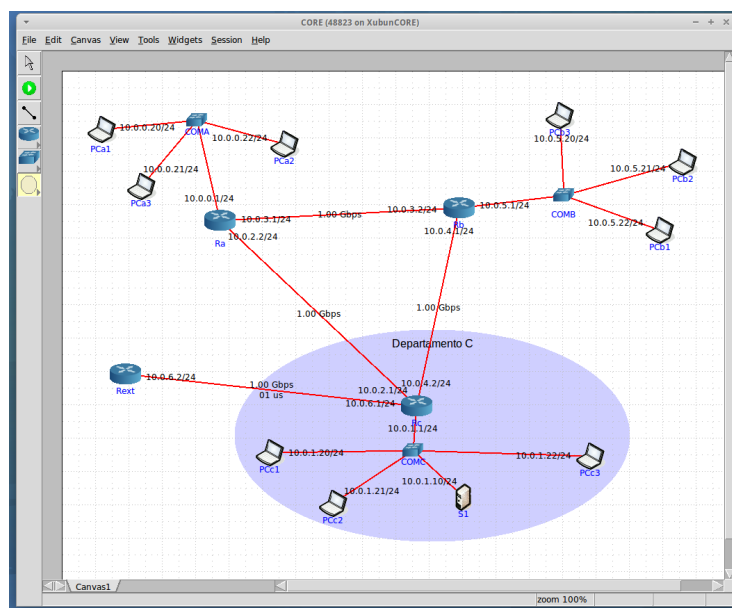
Os únicos campos que se alteram no cabeçalho IP, entre os diferentes fragmentos são as flags "*Fragment Offset*" e "*More Fragments*". A "*Fragment Offset*" permite-nos identificar a posição do fragmento no datagrama original e a "*More Fragments*" permite-nos concluir se há ou não mais fragmentos a veicular na rede. Desta forma, os equipamentos conseguem reconstituir o pacote original. Neste caso, irá agrupar os pacotes pela ordem crescente do valor da flag "*Fragment Offset*", isto é, 0 -> 185 -> 370.

Parte II

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



A foto acima mostra os vários endereços IP e a respectiva máscara (255.255.255.0, em notação CIDR, /24) atribuídos pelo CORE a cada equipamento.

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

Tratam-se de endereços privados porque não são vistos pela rede global e também porque pertencem à classe A que está definida entre os endereços 10.0.0.0 e 10.255.255.255

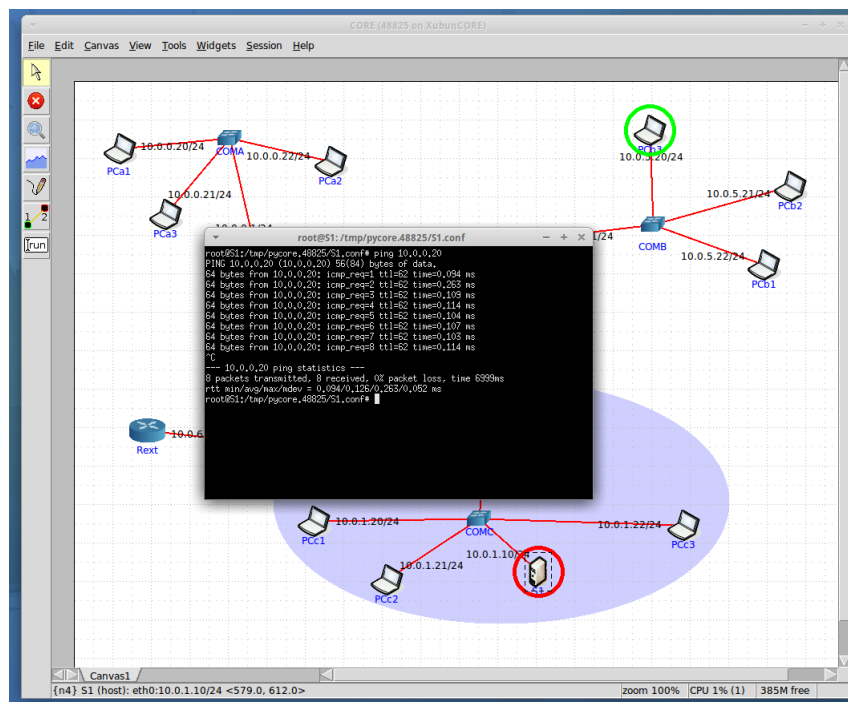
c. Porque razão não é atribuído um endereço IP aos switches?

Um switch (ou comutador) é um equipamento activo que funciona normalmente na camada 2 do modelo OSI e tem como principal funcionalidade a interligação de equipamentos de uma rede.

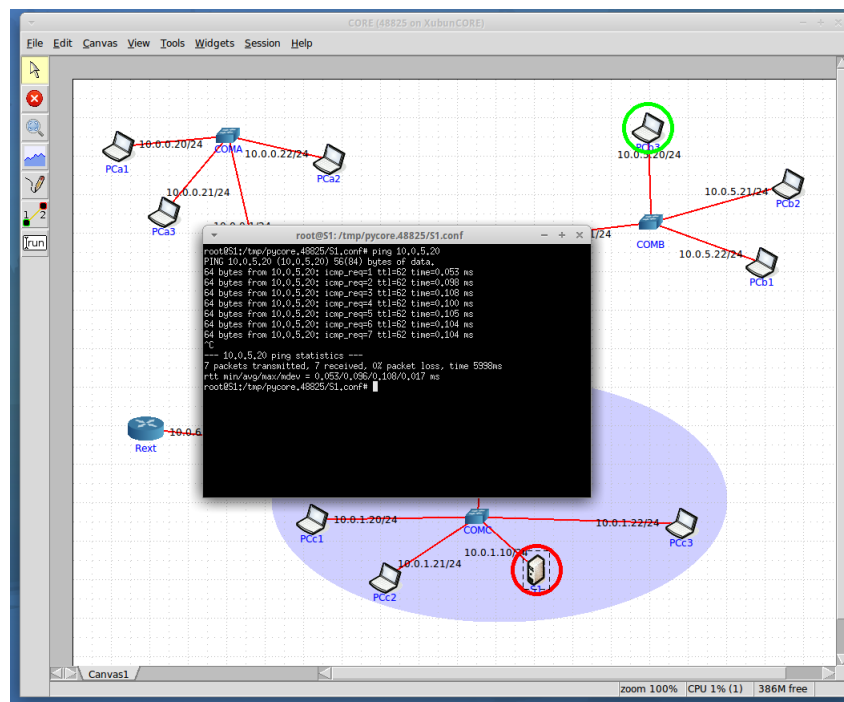
Numa primeira fase (antes do switch saber quem tem ligado a ele), quando um switch recebe informação numa determinada porta, transmite essa mesma informação por todas as outras portas, exceto por aquela que recebeu essa informação. Os switches registam o endereço MAC dos dispositivos que estão ligados a cada porta do equipamento. Sempre que um equipamento envia uma frame (trama), o switch analisa o endereço MAC de destino e comuta a frame para a porta onde se encontra a máquina de destino. Desta forma, 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).

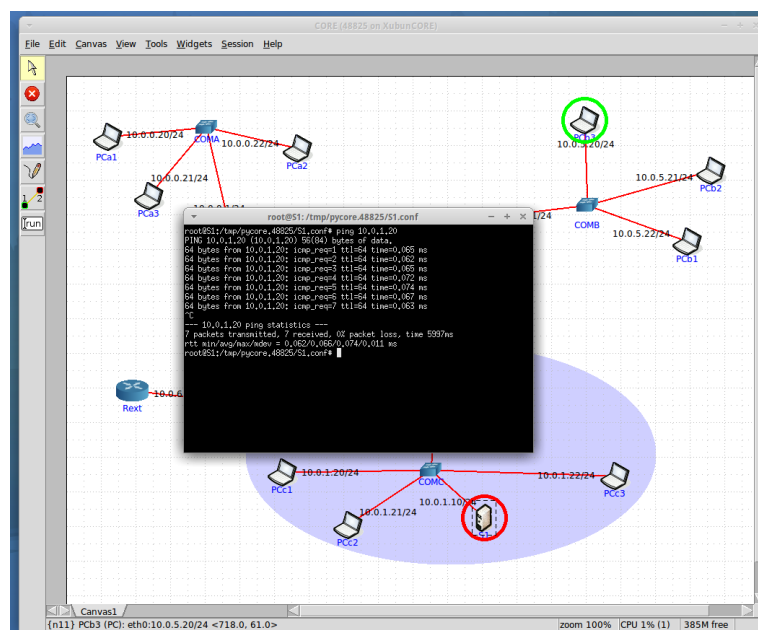
Do servidor S1 para o PCa1(Departamento A)



Do servidor S1 para o PCb3(Departamento B)



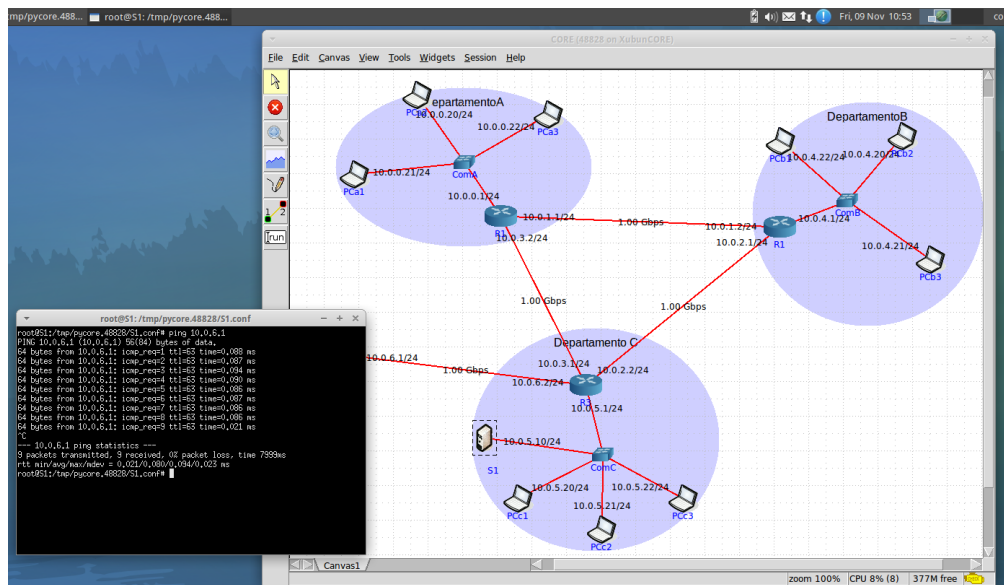
Do servidor S1 para o PCc1(Departamento C)



Como podemos observar, existe sempre conetividade entre os PC's de cada departamento e o Servidor S1. Verificámos isso com o envio de pacotes por cada ping feito para o servidor que os recebe e envia uma mensagem de volta para o respetivo PC, por exemplo, do número do pacote recebido, do tempo de ida e volta, do TTL..Se não houvesse conetividade IP entre os equipamentos ser-nos-ia devolvida uma mensagem do género "Reply from <endereço do servidor>:Destination host unreachable." por cada pacote.

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

(Por equívoco apagamos a primeira topologia CORE da rede local da empresa, por isso, tivemos de reconstruir. Portanto, alguns dados são diferentes)



Como podemos observar, também existe conectividade entre o router de acesso, Rext, e o servidor S1 do departamento C, pelas razões que foram mencionadas na alínea anterior.

Nota: Para responder às questões a partir daqui foi utilizado o Modelo Core da alínea anterior 1E

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@PCa1:/tmp/pycore.53971/PCa1.conf# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 10.0.0.1 0.0.0.0 UG 0 0 0 eth0
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
root@PCa1:/tmp/pycore.53971/PCa1.conf#

```

Figure 1: Tabela de encaminhamento laptop A

```

root@R1: /tmp/pycore.53971/R1.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         0.0.0.0         255.255.255.0   U        0  0        0     eth1
10.0.2.0         10.0.1.2        255.255.255.0   UG       0  0        0     eth1
10.0.3.0         0.0.0.0         255.255.255.0   U        0  0        0     eth2
10.0.4.0         10.0.1.2        255.255.255.0   UG       0  0        0     eth1
10.0.5.0         10.0.3.1        255.255.255.0   UG       0  0        0     eth2
10.0.6.0         10.0.3.1        255.255.255.0   UG       0  0        0     eth2
root@R1: /tmp/pycore.53971/R1.conf#

```

Figure 2: Tabela de encaminhamento router A

Nas tabelas unicast (IPv4) obtidas, retiramos alguma informação relativa à rota que o pacote terá de fazer. A coluna "*Destination*" indica a sub-rede de destino, a "*Gateway*" indica por que equipamento terá de passar o pacote e a "*Genmask*" o tipo de máscara usada.

Como se pode observar na tabela do laptop, vemos que tem duas entradas. A linha 0.0.0.0 é o endereço default que se usa quando não é conhecido o destino de um pacote. A outra linha que tem a destination 10.0.0.0 é usado quando o laptop pretende mandar um pacote para a própria rede.

Relativamente à tabela de endereçamento do router possui as redes que são possíveis atingir e os respetivos gateways. No caso da gateway ser 0.0.0.0, se o endereço destino estiver incluído na rede 10.0.0.0 e se não existissem mais indicações de destinos, então o pacote iria seguir um caminho aleatório.

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

No router, o encaminhamento que está a ser usado é dinâmico pois permite ao pacote seguir caminhos diferentes quando não é lhe é possível fazer a rota prevista. Podemos fundamentar isto na coluna "*CMD*" que nos diz que o router inclui o protocolo *ospfd*. Para o caso do laptop, concluímos que o encaminhamento é estático pois não é utilizado nenhum protocolo.

```
root@PCa1: /tmp/pycore.53971/PCa1.conf
root@PCa1:/tmp/pycore.53971/PCa1.conf# netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags    MSS Window  irtt Iface
0.0.0.0          10.0.0.1        0.0.0.0         UG        0 0        0 eth0
10.0.0.0          0.0.0.0         255.255.255.0   U         0 0        0 eth0
root@PCa1:/tmp/pycore.53971/PCa1.conf# ps -A
PID TTY          TIME CMD
  1 ?            00:00:00 vncd
 72 pts/9        00:00:00 bash
127 pts/10        00:00:00 bash
181 pts/11        00:00:00 bash
236 pts/11        00:00:00 ps
root@PCa1:/tmp/pycore.53971/PCa1.conf#
```

Figure 3: Processos do laptop

```
root@R1: /tmp/pycore.53971/R1.conf
root@R1:/tmp/pycore.53971/R1.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          0.0.0.0         255.255.255.0   U         0 0        0 eth1
10.0.2.0          10.0.1.2        255.255.255.0   UG        0 0        0 eth1
10.0.3.0          0.0.0.0         255.255.255.0   U         0 0        0 eth2
10.0.4.0          10.0.1.2        255.255.255.0   UG        0 0        0 eth1
10.0.5.0          10.0.3.1        255.255.255.0   UG        0 0        0 eth2
10.0.6.0          10.0.3.1        255.255.255.0   UG        0 0        0 eth2
root@R1:/tmp/pycore.53971/R1.conf# ps -A
PID TTY          TIME CMD
  1 ?            00:00:00 vncd
 55 ?            00:00:00 zebra
 62 ?            00:00:00 ospf6d
 72 ?            00:00:03 ospf6d
226 pts/8        00:00:00 bash
296 pts/8        00:00:00 ps
root@R1:/tmp/pycore.53971/R1.conf#
```

Figure 4: Processos do router

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.

Ao eliminarmos o endereço de destino 0.0.0.0, o default, estamos a cortar a possibilidade do servidor S1 se conectar com outros equipamentos fora da rede do departamento C, podendo apenas conectar-se aos laptops do departamento em questão. Neste cenário, os utilizadores ao usarem o comando ping para ver se existe conexão, por exemplo, com PCb2, a mensagem que aparece é *Network unreachable*, isto é, "*Impossível atingir a rede*".

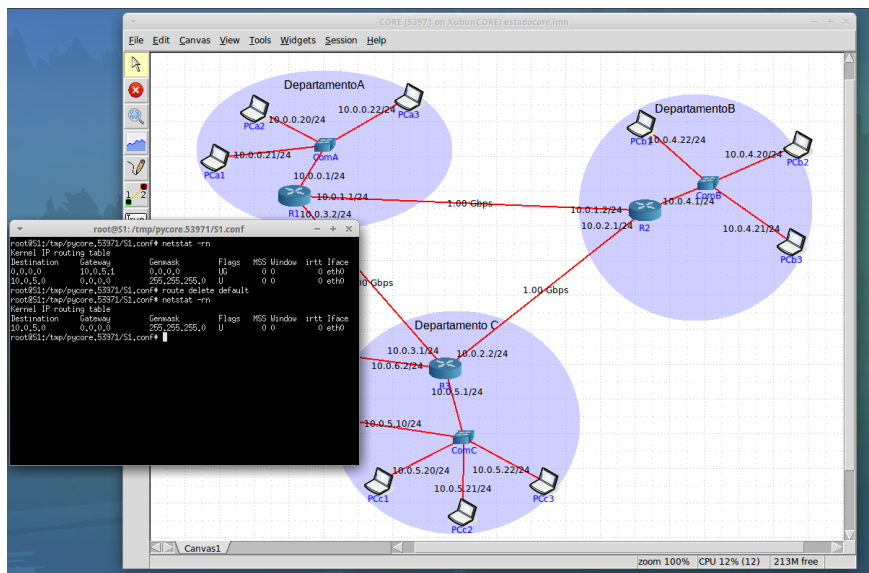


Figure 5: Tabela de Encaminhamento do Servidor S1

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 registre os comandos que usou.

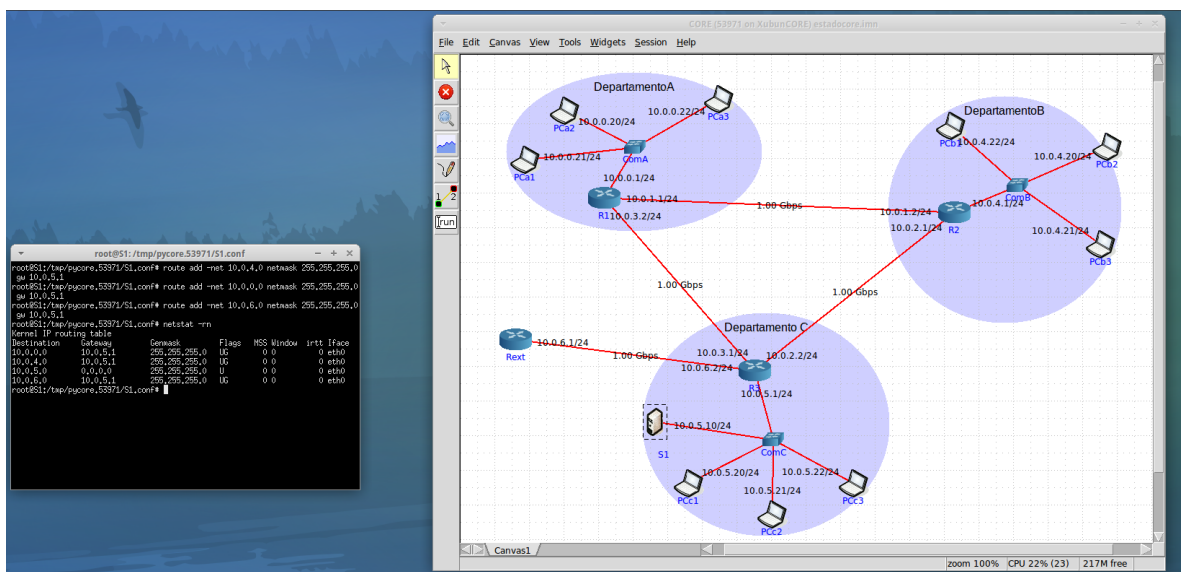


Figure 6: Adicionar Rota - Departamentos B,A e Router de Acesso(por esta ordem)

O comando usado para retomar a conexão foi: `route add -net 10.0.X.0 netmask 255.255.255.0 gw 10.0.5.1`. Devido a este comando são criadas rotas entre o servidor e os diversos departamentos e o servidor 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.

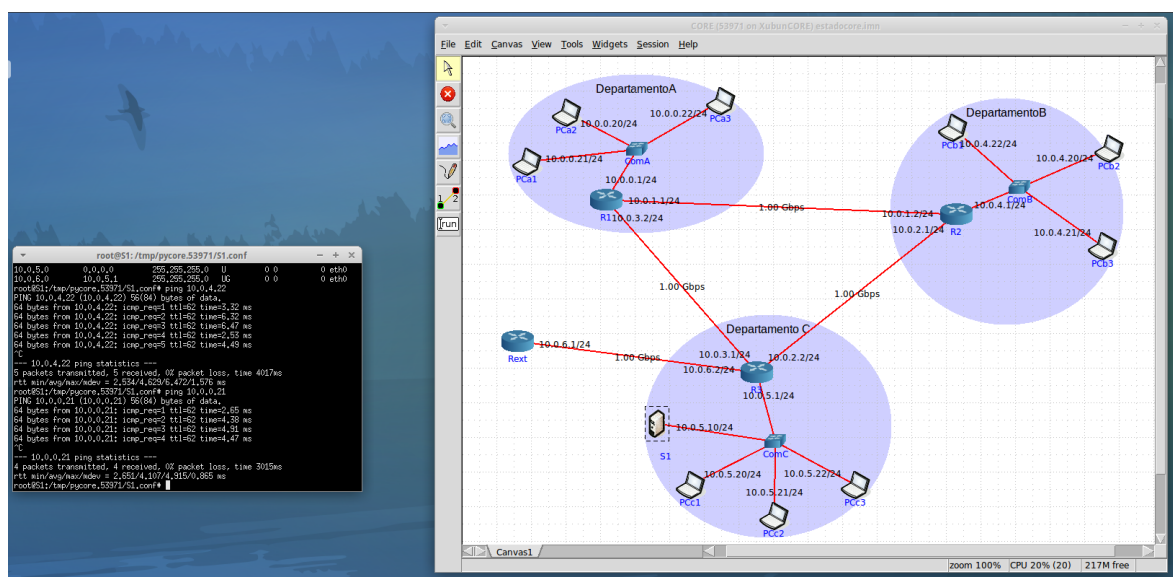


Figure 7: Departamento A e B

Como podemos observar na figura 7, a conectividade foi reestabelecida, pois aplicando o comando ping do servidor S1 para laptops do departamento A e B, verificamos que existe envio de packets do S1.

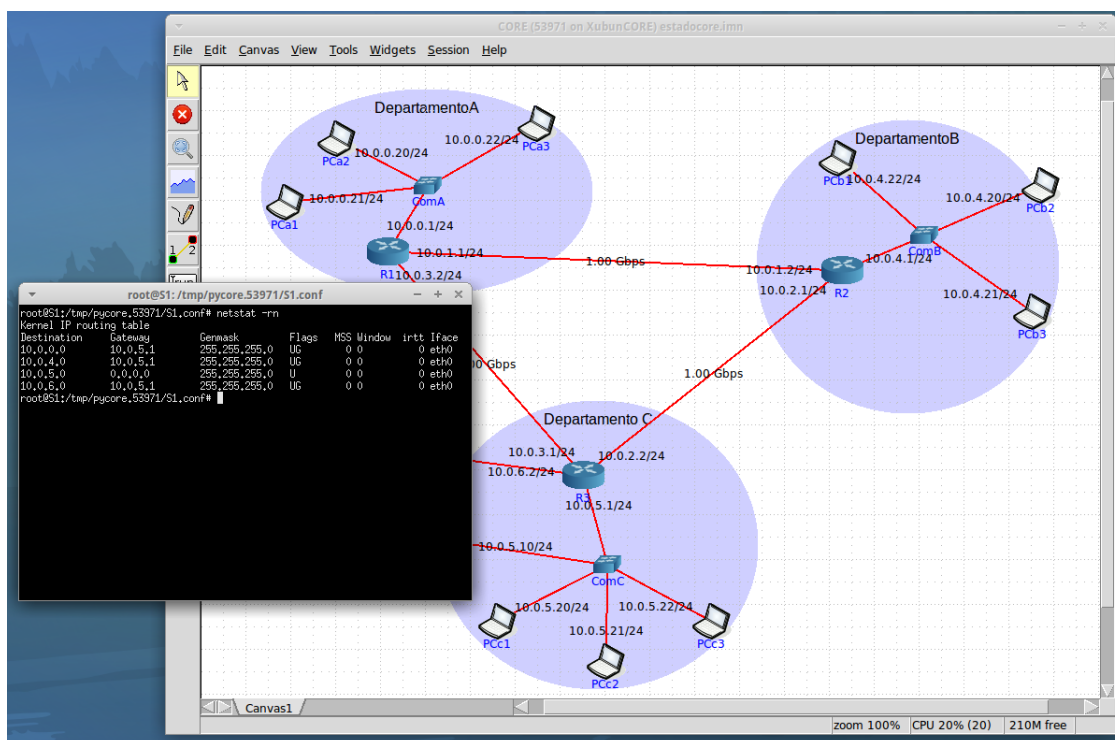


Figure 8: Tabela de Encaminhamento

Parte III

Definição de Sub-redes

1) Considere que dispõe 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 usadas.

O nosso IP da rede é 172.31.48.0/20 e dado que a máscara é 20 quer dizer que os endereços possíveis estão entre 172.31.48.0 e 172.31.63.255. Como existem 3 departamentos, para representar as 3 redes são necessários 3 bits, $2^3 - 2 = 6 > 3$, (é subtraído 2, devido a 000 e 111 estarem reservados). Pela razão referida anteriormente, usar 2 bits foi excluído pois $2^2 - 2 = 2 < 3$.

172.31.0011|XXX|0.0

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

Como se pode observar na tabela em cima, como são usados 3 bits, existem 8 opções possíveis para endereços para as sub-redes. No entanto, como as possibilidades 000 e 111 ficam reservadas, ficamos cingidos a 6 endereços possíveis. Deste modo foi atribuída, de forma aleatória, uma das possibilidades a cada departamento.

IP host por departamento			
Departamento	IP	IP-Inicio	IP-Fim
A	172.31.52.0/23	172.31.52.1	172.31.53.254
B	172.31.56.0/23	172.31.56.1	172.31.57.254
C	172.31.58.0/23	172.31.58.1	172.31.58.254

Com o auxílio da tabela dos IP's de host para cada departamento, foram atribuídos IP's (sem ser com tudo zeros ou tudo uns) que estão dentro dos intervalos possíveis de cada departamento.

Endereços atribuídos a cada dispositivo					
Dep. A	IP atribuido	Dep. B	IP atribuido	Dep. C	IP atribuido
Pca1	172.31.52.2/23	Pcb1	172.31.56.20/23	Pcc1	172.31.58.25/23
Pca2	172.31.52.3/23	Pcb2	172.31.56.31/23	Pcc2	172.31.58.40/23
Pca3	172.31.52.4/23	Pcb3	172.31.56.47/23	Pcc3	172.31.58.111/23
R1	172.31.52.1/23	R2	172.31.56.1/23	R3	172.31.58.1/23
				S1	172.31.58.5/23

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

Como reservamos 3 bits para fazer sub-netting, a nossa máscara passa de /20 para /23 ficando o seu valor decimal 255.255.254.0, sobrando 9 bits para podermos alterar. O número de host é dado por $2^9 - 2$ (1 reservado para broadcasting e outro para comunicar com todos os dispositivos), ficando com 510 hosts IP .

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

Na tabela está representado os novos endereços atribuídos a cada interface. Para garantir a conectividade usamos o comando ping de um laptop do departamento A para um laptop dos departamentos B e C. De seguida, usamos o mesmo comando para um laptop do departamento B para um laptop do departamento C e por fim usamos o comando ping do router exterior para o servidor S1. Tudo o que foi mencionado, pode ser comprovado nas imagens que se encontram em baixo.

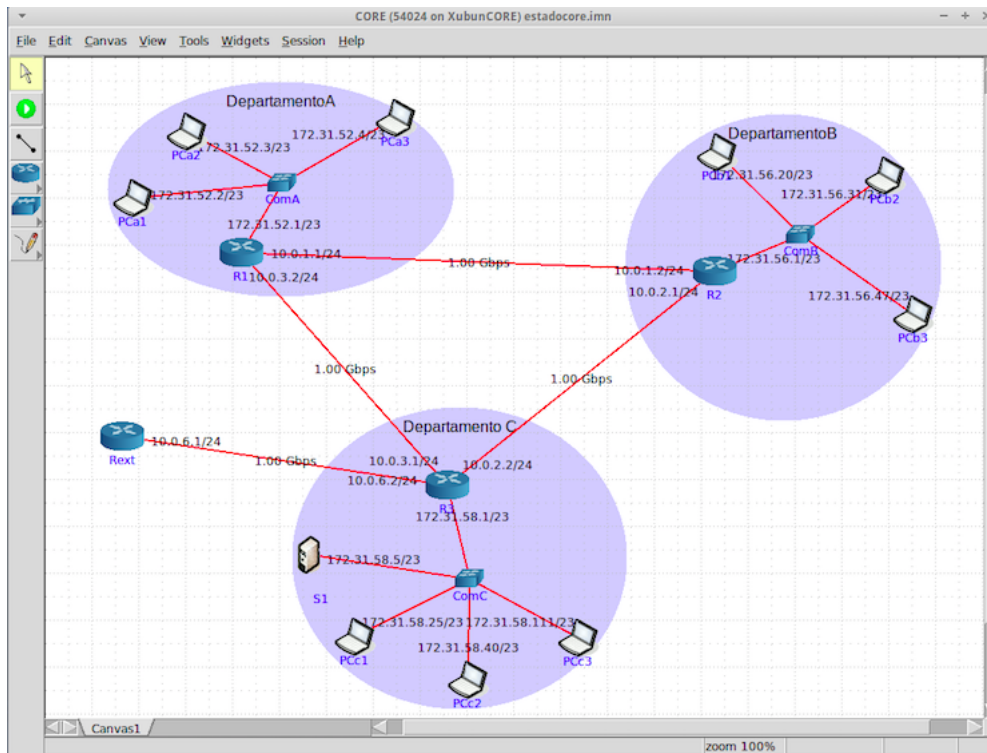


Figure 9: Equipamentos e Departamentos com novos IP's

```
root@PCa1: /tmp/pycore.54024/PCa1.conf
root@PCa1:/tmp/pycore.54024/PCa1.conf# ping 172.31.56.31
PING 172.31.56.31 (172.31.56.31) 56(84) bytes of data:
64 bytes from 172.31.56.31: icmp_req=1 ttl=62 time=1.79 ms
64 bytes from 172.31.56.31: icmp_req=2 ttl=62 time=1.02 ms
^C
--- 172.31.56.31 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 1.022/1.406/1.791/0.386 ms
root@PCa1:/tmp/pycore.54024/PCa1.conf# ping 172.31.58.40
PING 172.31.58.40 (172.31.58.40) 56(84) bytes of data:
64 bytes from 172.31.58.40: icmp_req=1 ttl=62 time=1.21 ms
64 bytes from 172.31.58.40: icmp_req=2 ttl=62 time=0.626 ms
64 bytes from 172.31.58.40: icmp_req=3 ttl=62 time=1.06 ms
^C
--- 172.31.58.40 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.626/0.967/1.210/0.249 ms
root@PCa1:/tmp/pycore.54024/PCa1.conf#
```

Figure 10: Ping de um laptop do DepA para um laptop do Depb e DepC

```
root@PCb1: /tmp/pycore.54024/PCb1.conf
root@PCb1:/tmp/pycore.54024/PCb1.conf# ping 172.31.58.25
PING 172.31.58.25 (172.31.58.25) 56(84) bytes of data.
64 bytes from 172.31.58.25: icmp_req=1 ttl=62 time=1.50 ms
64 bytes from 172.31.58.25: icmp_req=2 ttl=62 time=1.06 ms
64 bytes from 172.31.58.25: icmp_req=3 ttl=62 time=2.10 ms
^C
--- 172.31.58.25 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.065/1.557/2.101/0.426 ms
root@PCb1:/tmp/pycore.54024/PCb1.conf#
```

Figure 11: Ping de um laptop do DepB para um laptop do DepC

```
root@Rext: /tmp/pycore.54024/Rext.conf
root@Rext:/tmp/pycore.54024/Rext.conf# ping 172.31.58.5
PING 172.31.58.5 (172.31.58.5) 56(84) bytes of data.
64 bytes from 172.31.58.5: icmp_req=1 ttl=63 time=1.62 ms
64 bytes from 172.31.58.5: icmp_req=2 ttl=63 time=0.632 ms
64 bytes from 172.31.58.5: icmp_req=3 ttl=63 time=0.897 ms
64 bytes from 172.31.58.5: icmp_req=4 ttl=63 time=1.43 ms
^C
--- 172.31.58.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.632/1.146/1.621/0.399 ms
root@Rext:/tmp/pycore.54024/Rext.conf#
```

Figure 12: Ping do router exterior para o servidor S1

Conclusão

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

Por outro lado, na segunda parte do trabalho, houve um foco no funcionamento do processo de endereçamento e encaminhamento IP. Após a construção da topologia com os vários departamentos e respetivos equipamentos, permitiu nos analisar a conectividade entre os equipamentos de cada departamento (com as rotas, tipo de encaminhamento: estático ou dinâmico). Em suma, vimos o funcionamento do encaminhamento entre redes diferentes de 3 departamentos, adquirindo conhecimentos essenciais em torno da divisão de sub-netting. Estes casos teóricos, podem facilmente ser aplicados na prática e em infra-estruturas a que nos ligamos diariamente.