

TP4

Protocolo IPv4 (802.11)

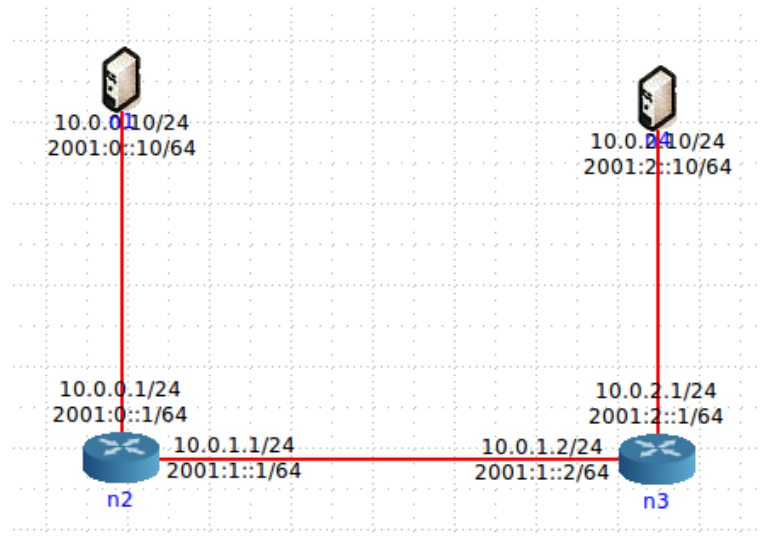
André Freitas A74619

Joel Morais A70841

Sofia Carvalho A76658

PARTE 1

1.



- a. Active o *wireshark* ou o *tcpdump* no *host* n4. Numa *shell* de n4, execute o comando *traceroute -I* para o endereço IP do *host* n1.

R:

```
root@n4: /tmp/pycore.46130/n4.conf
15:28:16.154206 IP (tos 0x0, ttl 6, id 56231, offset 0, flags [none], proto ICMP (1), length 60)
  10.0.2.10 > A9: ICMP echo request, id 314, seq 16, length 40
15:28:16.154233 IP (tos 0x0, ttl 62, id 24005, offset 0, flags [none], proto ICMP (1), length 60)
  A9 > 10.0.2.10: ICMP echo reply, id 314, seq 16, length 40
15:28:16.157467 IP (tos 0x0, ttl 6, id 48484, offset 0, flags [none], proto ICMP (1), length 60)
  10.0.2.10 > A9: ICMP echo request, id 314, seq 17, length 40
15:28:16.157523 IP (tos 0x0, ttl 62, id 35355, offset 0, flags [none], proto ICMP (1), length 60)
  A9 > 10.0.2.10: ICMP echo reply, id 314, seq 17, length 40
15:28:16.158341 IP (tos 0x0, ttl 6, id 2929, offset 0, flags [none], proto ICMP (1), length 60)
  10.0.2.10 > A9: ICMP echo request, id 314, seq 18, length 40
15:28:16.158392 IP (tos 0x0, ttl 62, id 56233, offset 0, flags [none], proto ICMP (1), length 60)
  A9 > 10.0.2.10: ICMP echo reply, id 314, seq 18, length 40
15:28:16.158953 IP (tos 0x0, ttl 7, id 46873, offset 0, flags [none], proto ICMP (1), length 60)
  10.0.2.10 > A9: ICMP echo request, id 314, seq 19, length 40
15:28:16.159003 IP (tos 0x0, ttl 62, id 56234, offset 0, flags [none], proto ICMP (1), length 60)
  A9 > 10.0.2.10: ICMP echo reply, id 314, seq 19, length 40
15:28:16.160343 IP (tos 0x0, ttl 7, id 57925, offset 0, flags [none], proto ICMP (1), length 60)
  10.0.2.10 > A9: ICMP echo request, id 314, seq 20, length 40
15:28:16.160394 IP (tos 0x0, ttl 62, id 35357, offset 0, flags [none], proto ICMP (1), length 60)
  A9 > 10.0.2.10: ICMP echo reply, id 314, seq 20, length 40
15:28:16.160951 IP (tos 0x0, ttl 7, id 24506, offset 0, flags [none], proto ICMP (1), length 60)
  10.0.2.10 > A9: ICMP echo request, id 314, seq 21, length 40
15:28:16.161000 IP (tos 0x0, ttl 7, id 24506, offset 0, flags [none], proto ICMP (1), length 60)
  A9 > 10.0.2.10: ICMP echo reply, id 314, seq 21, length 40

root@n4: /tmp/pycore.46130/n4.conf# traceroute -I 10.0.0.10
traceroute to 10.0.0.10 (10.0.0.10), 30 hops max, 60 byte packets
 1 10.0.2.1 (10.0.2.1) 0.078 ms 0.040 ms 0.019 ms
 2 10.0.1.1 (10.0.1.1) 0.056 ms 0.061 ms 0.031 ms
 3 A9 (10.0.0.10) 0.075 ms 0.046 ms 0.036 ms
root@n4: /tmp/pycore.46130/n4.conf#
```

b. Comente os resultados face ao comportamento esperado.

R: O tráfego ICMP enviado por n1 para 10.0.2.10 (n4) corresponde a 3 datagramas com o mesmo TTL de cada vez, pois não existe segurança na rede. Como por exemplo, pela análise da imagem acima, nos ICMP echo replies com TTL igual a 7.

c. Qual deve ser o valor inicial mínimo do campo TTL para alcançar o destino n1? Verifique na prática que a sua resposta está correta.

```
root@n1: /tmp/pycore.46130/n1.conf
root@n1: /tmp/pycore.46130/n1.conf# traceroute -I 10.0.2.10
traceroute to 10.0.2.10 (10.0.2.10), 30 hops max, 60 byte packets
 1  A0 (10.0.0.1) 0.066 ms 0.015 ms 0.012 ms
 2  10.0.1.2 (10.0.1.2) 0.045 ms 0.023 ms 0.022 ms
 3  10.0.2.10 (10.0.2.10) 0.040 ms 0.029 ms 0.028 ms
root@n1: /tmp/pycore.46130/n1.conf#
```

R: O tempo mínimo necessário para alcançar n4 é 3.

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

R: 1: $(0.066 + 0.015 + 0.012) / 3 = 0.067$ ms
2: $(0.045 + 0.023 + 0.022) / 3 = 0.03$ ms
3: $(0.040 + 0.029 + 0.028) / 3 = 0.032(3)$ ms

2.

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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.226	192.168.100.255	NBNS	92	Name query NB WPAD<00>
→	2 0.463121	192.168.100.201	192.168.100.174	ICMP	70	Echo (ping) request id=0x0001, seq=10891/35626, ttl=255 (reply in 3)
←	3 0.464177	192.168.100.174	192.168.100.201	ICMP	70	Echo (ping) reply id=0x0001, seq=10891/35626, ttl=64 (request in 2)

> Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Apple_f1:49:10 (10:dd:b1:f1:49:10)
v Internet Protocol Version 4, Src: 192.168.100.201, Dst: 192.168.100.174

0100 ... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 56
Identification: 0x3ddd (15837)
> Flags: 0x00
Fragment offset: 0
Time to live: 255
Protocol: ICMP (1)
Header checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.100.201

R: O endereço IP da interface ativa do computador é o endereço “Source”, que tem o valor 192.168.100.201

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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.226	192.168.100.255	NBNS	92	Name query NB WPAD<00>
2	0.463121	192.168.100.201	192.168.100.174	ICMP	70	Echo (ping) request id=0x0001, seq=10891/35626, ttl=255 (reply in 3)
3	0.464177	192.168.100.174	192.168.100.201	ICMP	70	Echo (nine) reply id=0x0001, seq=10891/35626, ttl=64 (request in 2)

> Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Apple_f1:49:10 (10:dd:b1:f1:49:10)
▼ Internet Protocol Version 4, Src: 192.168.100.201, Dst: 192.168.100.174
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 56
 Identification: 0x3ddd (15837)
 > Flags: 0x00
 Fragment offset: 0
 Time to live: 255
 Protocol: ICMP (1)
 Header checksum: 0x0000 [validation disabled]

R: O valor do campo protocolo é 1. Este campo identifica o tipo de protocolo e neste caso trata-se de 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*?

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.226	192.168.100.255	NBNS	92	Name query NB WPAD<00>
2	0.463121	192.168.100.201	192.168.100.174	ICMP	70	Echo (ping) request id=0x0001, seq=10891/35626, ttl=255 (reply in 3)
3	0.464177	192.168.100.174	192.168.100.201	ICMP	70	Echo (nine) reply id=0x0001, seq=10891/35626, ttl=64 (request in 2)

> Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Apple_f1:49:10 (10:dd:b1:f1:49:10)
▼ Internet Protocol Version 4, Src: 192.168.100.201, Dst: 192.168.100.174
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 56

R: O cabeçalho IP(v4) tem 20 *bytes*. O tamanho do campo de dados é 56 e o número de *bytes* deste é 56-20=36 *bytes*.

d. O datagrama IP foi fragmentado?

▼ Flags: 0x00
 0... = Reserved bit: Not set
 .0.. = Don't fragment: Not set
 ..0. = More fragments: Not set
 Fragment offset: 0

R: O datagrama IP não foi fragmentado, a partir da análise da informação das flags da imagem acima apresentada (“More Fragments: Not set”). Além disso, o offset toma o valor 0.

- 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 sua máquina. Para a sequência de mensagens ICMP enviadas pelo seu computador, indique que campos do cabeçalho IP variam de pacote para pacote.

No.	Source	Destination	Info	Length	Protocol	Time
1945	192.168.100.201	216.58.201.131	Payload (Encrypted), PKN: ...		258 QUIC	19
1948	192.168.100.201	216.58.201.131	Payload (Encrypted), PKN: ...		80 QUIC	19
1950	192.168.100.201	192.168.100.174	Echo (ping) request id=0x...		70 ICMP	19
1952	192.168.100.201	192.168.100.174	Echo (ping) request id=0x...		70 ICMP	19
1954	192.168.100.201	192.168.100.254	Refresh NB WORKGROUP<00>		110 NBNS	19
1955	192.168.100.201	192.168.100.254	Standard query 0x216f A www...		74 DNS	19
1957	192.168.100.201	216.58.201.132	57847→443 [SYN] Seq=0 Win=...		66 TCP	19
1958	192.168.100.201	216.58.201.132	Client Hello, PKN: 1, CID:...		1392 QUIC	19
1959	192.168.100.201	216.58.201.132	Payload (Encrypted), PKN: ...		430 QUIC	19

< >

> Frame 1950: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0

> Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Apple_f1:49:10 (10:dd:b1:f1:49:10)

> Internet Protocol Version 4, Src: 192.168.100.201, Dst: 192.168.100.174

> 0100 = Version: 4

> 0101 = Header Length: 20 bytes (5)

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

> Total Length: 56

> Identification: 0x3eee (16110)

> Flags: 0x00

> Fragment offset: 0

> Time to live: 255

> Protocol: ICMP (1)

> Header checksum: 0x0000 [validation disabled]

> [Header checksum status: Unverified]

> Source: 192.168.100.201

> Destination: 192.168.100.174

> [Source GeoIP: Unknown]

> [Destination GeoIP: Unknown]

No.	Source	Destination	Info	Length	Protocol	Time
1945	192.168.100.201	216.58.201.131	Payload (Encrypted), PKN: ...		258 QUIC	19
1948	192.168.100.201	216.58.201.131	Payload (Encrypted), PKN: ...		80 QUIC	19
1950	192.168.100.201	192.168.100.174	Echo (ping) request id=0x...		70 ICMP	19
1952	192.168.100.201	192.168.100.174	Echo (ping) request id=0x...		70 ICMP	19
1954	192.168.100.201	192.168.100.254	Refresh NB WORKGROUP<00>		110 NBNS	19
1955	192.168.100.201	192.168.100.254	Standard query 0x216f A www...		74 DNS	19
1957	192.168.100.201	216.58.201.132	57847→443 [SYN] Seq=0 Win=...		66 TCP	19
1958	192.168.100.201	216.58.201.132	Client Hello, PKN: 1, CID:...		1392 QUIC	19
1959	192.168.100.201	216.58.201.132	Payload (Encrypted), PKN: ...		430 QUIC	19

< >

> Frame 1952: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0

> Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Apple_f1:49:10 (10:dd:b1:f1:49:10)

> Internet Protocol Version 4, Src: 192.168.100.201, Dst: 192.168.100.174

> 0100 = Version: 4

> 0101 = Header Length: 20 bytes (5)

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

> Total Length: 56

> Identification: 0x3eef (16111)

> Flags: 0x00

> Fragment offset: 0

> Time to live: 1

> Protocol: ICMP (1)

> Header checksum: 0x0000 [validation disabled]

> [Header checksum status: Unverified]

> Source: 192.168.100.201

> Destination: 192.168.100.174

> [Source GeoIP: Unknown]

> [Destination GeoIP: Unknown]

> Internet Control Message Protocol

R: Os campos do cabeçalho IP que variam de pacote para pacote são o “Time to live” e o “Identification”.

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

R: Os campos de identificação do datagrama IP estão sequenciados (como se pode ver no campo *Identification* que consta nas imagens acima, sendo os seus valores 0x3eee e 0x3eef). O TTL também segue um padrão: verifica-se que os valores dos pedidos são sempre 255 e 1.

- 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.	Protocol	Source	Destination	Info	Length	Time
1965	TCP	216.58.201.132	192.168.100.201	443→57847 [ACK] Seq=1 Ack=196 Win=44032 Len=0	60	198.913380
1964	QUIC	216.58.201.132	192.168.100.201	Payload (Encrypted), PKN: 2	72	198.912419
1963	QUIC	216.58.201.132	192.168.100.201	Payload (Encrypted), PKN: 1	1392	198.912417
1960	TCP	216.58.201.132	192.168.100.201	443→57847 [SYN, ACK] Seq=0 Ack=1 Win=42900 Len=0 MSS=1430 SACK_PERM=1 WS=128	66	198.897962
1956	DNS	192.168.100.254	192.168.100.201	Standard query response 0x216f A www.google.com A 216.58.201.132 NS ns4.google.com NS ns1.go...	226	198.855553
1953	ICMP	192.168.100.174	192.168.100.201	Echo (ping) reply id=0x0001, seq=11050/10795, ttl=64 (request in 1952)	70	198.075391
← 1951	ICMP	192.168.100.174	192.168.100.201	Echo (ping) reply id=0x0001, seq=11049/10539, ttl=64 (request in 1950)	70	198.023965
1947	QUIC	216.58.201.131	192.168.100.201	Payload (Encrypted), PKN: 56	102	197.774175
1946	QUIC	216.58.201.131	192.168.100.201	Payload (Encrypted), PKN: 55	123	197.769415
1941	QUIC	216.58.201.131	192.168.100.201	Payload (Encrypted), PKN: 54	84	197.474614
> Frame 1951: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0 > Ethernet II, Src: Apple_f1:49:10 (10:dd:b1:f1:49:10), Dst: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89) > Internet Protocol Version 4, Src: 192.168.100.174, Dst: 192.168.100.201 0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 56 Identification: 0x7a1a (31258) Flags: 0x00 Fragment offset: 0 Time to live: 64 Protocol: ICMP (1) Header checksum: 0xb5e2 [validation disabled] [Header checksum status: Unverified] Source: 192.168.100.174 Destination: 192.168.100.201 [Source GeoIP: Unknown] [Destination GeoIP: Unknown] > Internet Control Message Protocol						
No.	Protocol	Source	Destination	Info	Length	Time
1965	TCP	216.58.201.132	192.168.100.201	443→57847 [ACK] Seq=1 Ack=196 Win=44032 Len=0	60	198.913380
1964	QUIC	216.58.201.132	192.168.100.201	Payload (Encrypted), PKN: 2	72	198.912419
1963	QUIC	216.58.201.132	192.168.100.201	Payload (Encrypted), PKN: 1	1392	198.912417
1960	TCP	216.58.201.132	192.168.100.201	443→57847 [SYN, ACK] Seq=0 Ack=1 Win=42900 Len=0 MSS=1430 SACK_PERM=1 WS=128	66	198.897962
1956	DNS	192.168.100.254	192.168.100.201	Standard query response 0x216f A www.google.com A 216.58.201.132 NS ns4.google.com NS ns1.go...	226	198.855553
← 1953	ICMP	192.168.100.174	192.168.100.201	Echo (ping) reply id=0x0001, seq=11050/10795, ttl=64 (request in 1952)	70	198.075391
1951	ICMP	192.168.100.174	192.168.100.201	Echo (ping) reply id=0x0001, seq=11049/10539, ttl=64 (request in 1950)	70	198.023965
1947	QUIC	216.58.201.131	192.168.100.201	Payload (Encrypted), PKN: 56	102	197.774175
1946	QUIC	216.58.201.131	192.168.100.201	Payload (Encrypted), PKN: 55	123	197.769415
1941	QUIC	216.58.201.131	192.168.100.201	Payload (Encrypted), PKN: 54	84	197.474614
> Frame 1953: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0 > Ethernet II, Src: Apple_f1:49:10 (10:dd:b1:f1:49:10), Dst: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89) > Internet Protocol Version 4, Src: 192.168.100.174, Dst: 192.168.100.201 0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 56 Identification: 0x5593 (21907) Flags: 0x00 Fragment offset: 0 Time to live: 64 Protocol: ICMP (1) Header checksum: 0xda69 [validation disabled] [Header checksum status: Unverified] Source: 192.168.100.174 Destination: 192.168.100.201 [Source GeoIP: Unknown] [Destination GeoIP: Unknown] > Internet Control Message Protocol						

R: O valor do campo TTL é 64. Permanece constante para todas as mensagens de resposta enviadas, pois trata-se de um *reply* e, por isso, o tempo de vida tem, regra geral, um valor alto. Este valor de tempo de vida alto serve para garantir que o *reply* chega ao destino independentemente dos saltos que possa vir a dar.

3.

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

```
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
▼ [3 IPv4 Fragments (4025 bytes): #5(1480), #6(1480), #7(1065)]
    [Frame: 5, payload: 0-1479 (1480 bytes)]
    [Frame: 6, payload: 1480-2959 (1480 bytes)]
    [Frame: 7, payload: 2960-4024 (1065 bytes)]
    [Fragment count: 3]
    [Reassembled IPv4 length: 4025]
    [Reassembled IPv4 data: 0800a1ea00013a18202020202020202020202020202020...]
> Internet Control Message Protocol
```

R: Houve necessidade de fragmentar o pacote inicial, pois este era demasiado grande, tendo de se dividir, neste caso, em 3 fragmentos.

- 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.	Protocol	Source	Destination	Info	Length	Time
1	SSDP	192.168.100.186	239.255.255.250	M-SEARCH * HTTP/1.1	179	0.000000
2	STP	CiscoInc_7e:6a:4b	Spanning-tree...	Conf. Root = 4096/720/00:0a:8a:97:74:80 Cost = 6008 Port = 0x800b	60	0.863019
3	MDNS	192.168.100.174	224.0.0.251	Standard query 0x0000 PTR _privet._tcp.local, "QM" question PTR _uscan._tcp.local, "QM" ques...	369	1.462150
4	MDNS	fe80::18cd:896f:2643:cc05	ff02::fb	Standard query 0x0000 PTR _privet._tcp.local, "QM" question PTR _uscan._tcp.local, "QM" ques...	389	1.462179
5	IPv4	192.168.100.201	192.168.100.174	Fragmented IP protocol (proto=ICMP 1, off=0, ID=50cb) [Reassembled in #7]	1514	1.884312
6	IPv4	192.168.100.201	192.168.100.174	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=50cb) [Reassembled in #7]	1514	1.884351
7	ICMP	192.168.100.201	192.168.100.174	Echo (ping) request id=0x0001 seq=14872/6702 ttl=255 (reply in 10)	1065	1.884370

> Frame 5: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

> Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Apple_f1:49:10 (10:dd:b1:f1:49:10)

> Internet Protocol Version 4, Src: 192.168.100.201, Dst: 192.168.100.174

0100 = Version: 4

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

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

Total Length: 1500

Identification: 0x50cb (20683)

> Flags: 0x01 (More Fragments)

0... = Reserved bit: Not set

..0... = Don't fragment: Not set

..1. = More fragments: Set

Fragment offset: 0

Time to live: 255

Protocol: ICMP (1)

Header checksum: 0x0000 [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.100.201

Destination: 192.168.100.174

[Source GeoIP: Unknown]

[Destination GeoIP: Unknown]

Reassembled IPv4 in frame: 7

> Data (1480 bytes)

Data: 0800a1ea00013a18202020202020202020202020202020...

[Length: 1480]

R: “Reassembled IPv4 in frame: 7” indica-nos que é um fragmento que vai ser reconstruído na trama 7. “More Fragments: Set” (toma valor 1) indica que existem mais fragmentos para além deste. Trata-se do primeiro

fragmento pois o offset é igual a 0. O segundo fragmento vai ter o offset igual a 1480. Em relação ao tamanho, temos 1480 de dados e 20 de header, logo, no total, 1500.

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

[illegible]

R: O que indica que não se trata do 1º fragmento é o facto de o offset ter o valor 1480, ou seja, diferente de 0. Há mais fragmentos pois conseguimos ver na flag que “More Fragments : Set” (toma valor 1).

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

No.	Protocol	Source	Destination	Info
*	5 IPv4	192.168.100.201	192.168.100.174	Fragmented IP protocol (proto=ICMP 1, off=0, ID=50cb) [Reassembled in #7]
*	6 IPv4	192.168.100.201	192.168.100.174	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=50cb) [Reassembled in #7]
→	7 ICMP	192.168.100.201	192.168.100.174	Echo (ping) request id=0x0001, seq=14872/6202, ttl=255 (reply in 10)

```

> Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Apple_f1:49:10 (10:dd:b1:f1:49:10)
v Internet Protocol Version 4, Src: 192.168.100.201, Dst: 192.168.100.174
  0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1085
    Identification: 0x50cb (20683)
  v Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  Fragment offset: 2960
  Time to live: 255
  Protocol: ICMP (1)
  Header checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.100.201
  Destination: 192.168.100.174
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
  v [3 IPv4 Fragments (4025 bytes): #5(1480), #6(1480), #7(1065)]
    [Frame: 5, payload: 0-1479 (1480 bytes)]
    [Frame: 6, payload: 1480-2959 (1480 bytes)]
    [Frame: 7, payload: 2960-4024 (1065 bytes)]
    [Fragment count: 3]
    [Reassembled IPv4 length: 4025]
    [Reassembled IPv4 data: 0800a1ea00013a18202020202020202020202020202020...]

```

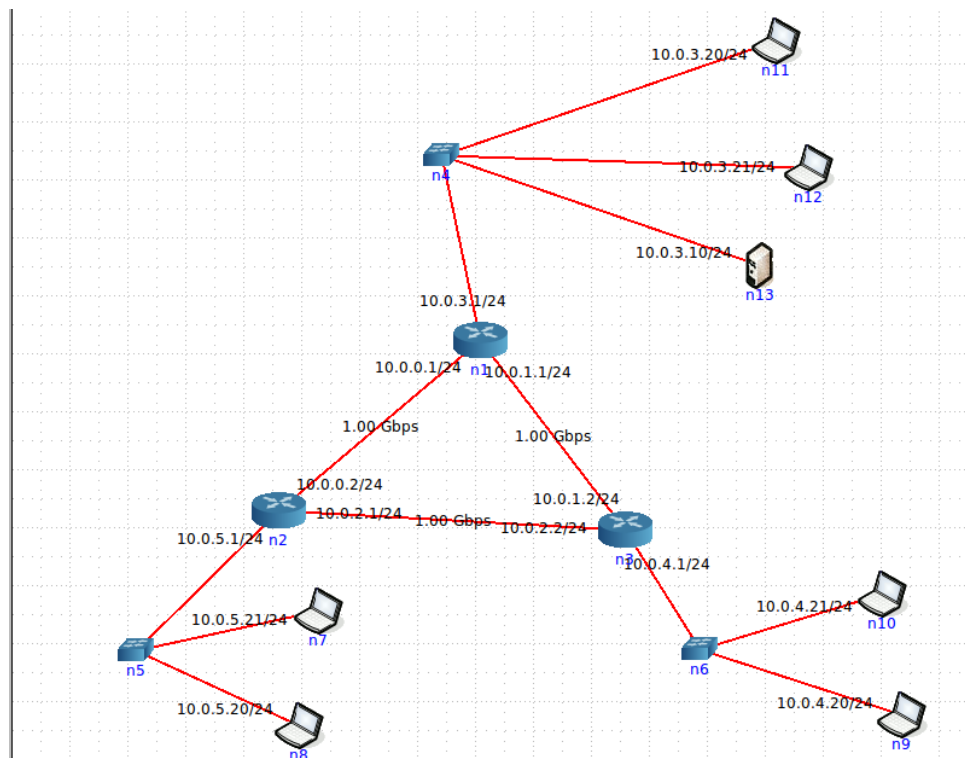
R: Foram criados 3 fragmentos a partir do datagrama original. Como estes 3 fragmentos vão ser reconstruídos na trama 7, a trama 7 corresponderá ao último fragmento do datagrama original. Através da flag “More Fragments: Not Set” (toma valor 0) sabemos que não há mais fragmentos, e através do offset igual a 2960 verifica-se que não se trata do primeiro fragmento, pois é diferente de 0, logo o fragmento em questão só pode ser o último.

- 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.

R: Os campos que mudam no cabeçalho IP entre os diferentes fragmentos são o *offset* dos fragmentos, assim como a identificação de cada um destes. Através do *offset*, é possível reconstruir o datagrama, pois este fornece a posição de cada fragmento.

PARTE 2

- 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. Se preferir, pode incluir uma imagem que ilustre de forma clara a topologia e o endereçamento.



R: Atribuímos a seguinte máscara de rede pois temos /24 no endereço IP, o que significa que temos 24 bits identificadores da rede, sendo que isso corresponde ao endereço 255.255.255.0, visto que a máscara vai conter o limite máximo de valores decimais quando temos os primeiros 24 bits do endereço IP todos a 1.

N1 to N4: ip=10.0.3.1	Mascara de rede= 255.255.255.0
N12: ip = 10.0.3.21	Mascara de rede= 255.255.255.0
N1 to N2: ip=10.0.0.1	Mascara de rede= 255.255.255.0
N1 to N3: ip=10.0.1.2	Mascara de rede= 255.255.255.0
N11: ip=10.0.3.20	Mascara de rede= 255.255.255.0
N13: ip=10.0.3.10	Mascara de rede= 255.255.255.0
N3 to N1: ip= 10.0.1.2	Mascara de rede= 255.255.255.0

N3 to N2: ip=10.0.2.2	Mascara de rede= 255.255.255.0
N2 to N3: 10.0.2.1	Mascara de rede= 255.255.255.0
N2 to N1: 10.0.0.2	Mascara de rede= 255.255.255.0
N7: ip= 10.0.5.21	Mascara de rede= 255.255.255.0
N8: ip=10.0.5.20	Mascara de rede= 255.255.255.0
N9: ip=10.0.4.20	Mascara de rede= 255.255.255.0
N10: 10.0.4.21	Mascara de rede= 255.255.255.0

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

R: Estes endereçamentos vão ser privados, visto que eles não têm acesso direto à internet.

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

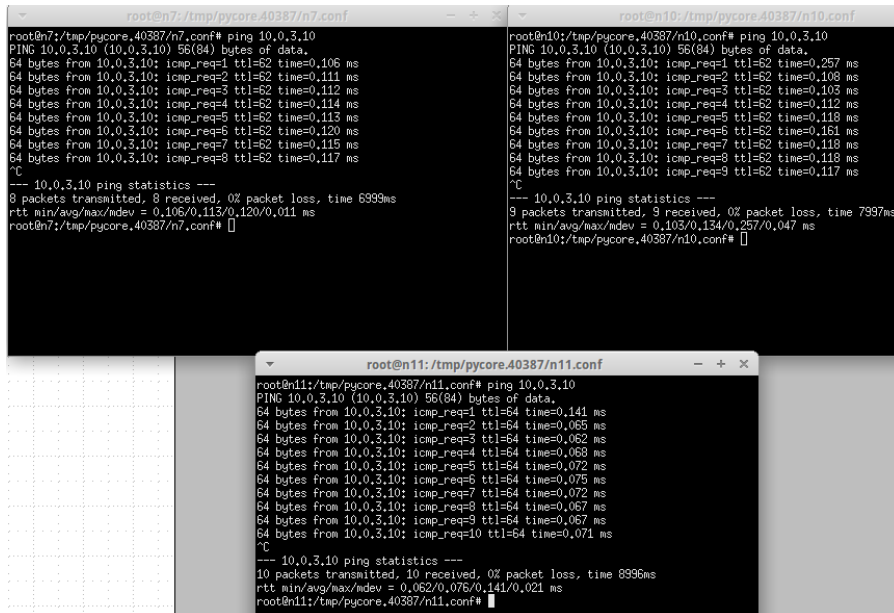
```

root@n13: /tmp/pycore.40387/n13.conf
root@n13:/tmp/pycore.40387/n13.conf# route delete
Usage: inet_route [-vF] del {-host|-net} Target[/prefix] [gw Gw] [metric M] [[dev
v] If]
       inet_route [-vF] add {-host|-net} Target[/prefix] [gw Gw] [metric M]
       [netmask N] [mss Mss] [window W] [irtt I]
       [mod] [dyn] [reinststate] [[dev] If]
       inet_route [-vF] add {-host|-net} Target[/prefix] [metric M] reject
       inet_route [-FC] flush      NOT supported
root@n13:/tmp/pycore.40387/n13.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@n13:/tmp/pycore.40387/n13.conf# route delete 0.0.0.0
SIOCDELRT: Processo inexistente
root@n13:/tmp/pycore.40387/n13.conf# route delete default
root@n13:/tmp/pycore.40387/n13.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@n13:/tmp/pycore.40387/n13.conf# █

```

R: Não é atribuído um endereço IP aos *switches* porque estes são usados em ligações de nível 2, e o endereço IP só irá ser atribuído em ligações de nível 3, nível de rede.

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



```
root@n7:/tmp/pycore.40387/n7.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data:
64 bytes from 10.0.3.10: icmp_req=1 ttl=62 time=0.106 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=62 time=0.111 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=62 time=0.112 ms
64 bytes from 10.0.3.10: icmp_req=4 ttl=62 time=0.114 ms
64 bytes from 10.0.3.10: icmp_req=5 ttl=62 time=0.113 ms
64 bytes from 10.0.3.10: icmp_req=6 ttl=62 time=0.120 ms
64 bytes from 10.0.3.10: icmp_req=7 ttl=62 time=0.115 ms
64 bytes from 10.0.3.10: icmp_req=8 ttl=62 time=0.117 ms
^C
--- 10.0.3.10 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 699ms
rtt min/avg/max/mdev = 0.106/0.113/0.120/0.011 ms
root@n7:/tmp/pycore.40387/n7.conf#

root@n10:/tmp/pycore.40387/n10.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data:
64 bytes from 10.0.3.10: icmp_req=1 ttl=62 time=0.257 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=62 time=0.108 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=62 time=0.103 ms
64 bytes from 10.0.3.10: icmp_req=4 ttl=62 time=0.112 ms
64 bytes from 10.0.3.10: icmp_req=5 ttl=62 time=0.118 ms
64 bytes from 10.0.3.10: icmp_req=6 ttl=62 time=0.161 ms
64 bytes from 10.0.3.10: icmp_req=7 ttl=62 time=0.118 ms
64 bytes from 10.0.3.10: icmp_req=8 ttl=62 time=0.118 ms
64 bytes from 10.0.3.10: icmp_req=9 ttl=62 time=0.117 ms
^C
--- 10.0.3.10 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 799ms
rtt min/avg/max/mdev = 0.103/0.134/0.257/0.047 ms
root@n10:/tmp/pycore.40387/n10.conf#

root@n11:/tmp/pycore.40387/n11.conf# ping 10.0.3.10
PING 10.0.3.10 (10.0.3.10) 56(84) bytes of data:
64 bytes from 10.0.3.10: icmp_req=1 ttl=64 time=0.141 ms
64 bytes from 10.0.3.10: icmp_req=2 ttl=64 time=0.065 ms
64 bytes from 10.0.3.10: icmp_req=3 ttl=64 time=0.062 ms
64 bytes from 10.0.3.10: icmp_req=4 ttl=64 time=0.068 ms
64 bytes from 10.0.3.10: icmp_req=5 ttl=64 time=0.072 ms
64 bytes from 10.0.3.10: icmp_req=6 ttl=64 time=0.075 ms
64 bytes from 10.0.3.10: icmp_req=7 ttl=64 time=0.072 ms
64 bytes from 10.0.3.10: icmp_req=8 ttl=64 time=0.067 ms
64 bytes from 10.0.3.10: icmp_req=9 ttl=64 time=0.067 ms
64 bytes from 10.0.3.10: icmp_req=10 ttl=64 time=0.071 ms
^C
--- 10.0.3.10 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 899ms
rtt min/avg/max/mdev = 0.062/0.075/0.141/0.021 ms
root@n11:/tmp/pycore.40387/n11.conf#
```

R: Como se pode ver pela figura acima, existe conectividade entre um laptop de cada departamento e o servidor do departamento A, sendo n7 do departamento B, n10 do departamento C, n11 do departamento A e 10.0.3.10 o endereço IP do servidor do departamento A.

- 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 respectivo (man netstat).

Router:

```
root@n1: /tmp/pycore.40387/n1.conf
root@n1:/tmp/pycore.40387/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         0.0.0.0         255.255.255.0   U        0 0        0 eth1
10.0.2.0         10.0.0.2        255.255.255.0   UG       0 0        0 eth0
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         10.0.0.2        255.255.255.0   UG       0 0        0 eth0
root@n1:/tmp/pycore.40387/n1.conf#
```

Laptop:

```
root@n11: /tmp/pycore.40387/n11.conf
root@n11:/tmp/pycore.40387/n11.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@n11:/tmp/pycore.40387/n11.conf#
```

R: Relativamente ao router N1: A primeira entrada da tabela tem como rede destino a rede entre os routers N1 e N2 (10.0.0.0) e o próximo salto vai corresponder ao próprio router N1 (10.0.0.1, sendo na tabela representado por 0.0.0.0), pois já estamos dentro dessa rede.

Para a segunda entrada, a rede destino é entre os routers N1 e N3 (10.0.1.0) e o próximo salto vai corresponder ao próprio router N1 (10.0.1.1, sendo na tabela representado por 0.0.0.0), pois já estamos dentro dessa rede.

Para a terceira entrada, o destino é a rede dos routers N2 e N3 (10.0.2.0) e o próximo salto vai corresponder ao router N2 (10.0.0.2) visto

ser o caminho mais próximo de chegarmos a essa rede através do router N1.

Para a quarta entrada, o destino é a rede do departamento A (10.0.3.0) e o próximo salto vai corresponder ao router N1 (10.0.0.1, sendo na tabela representado por 0.0.0.0), pois já estamos dentro dessa rede.

Para a quinta entrada, o destino é a rede do departamento C (10.0.4.0) e o próximo salto vai corresponder ao *router* N3 (10.0.1.2) visto que é o caminho mais próximo de chegarmos a essa rede através do router N1.

Para a sexta entrada, o destino é a rede do departamento B (10.0.5.0) e o próximo salto vai corresponder ao *router* N2 (10.0.0.2) pois é o caminho mais próximo para chegarmos a essa rede através do router N1.

Relativamente ao laptop: Para a primeira entrada da tabela de encaminhamento, a rede destino é a rota por defeito (0.0.0.0) e o próximo salto vai corresponder ao *router* N1 (10.0.3.1).

Para a segunda entrada, a rede destino é a rede do departamento A (10.0.3.1) e o próximo salto vai corresponder ao próprio *host* (10.0.3.10, sendo que na tabela corresponde a 0.0.0.0).

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

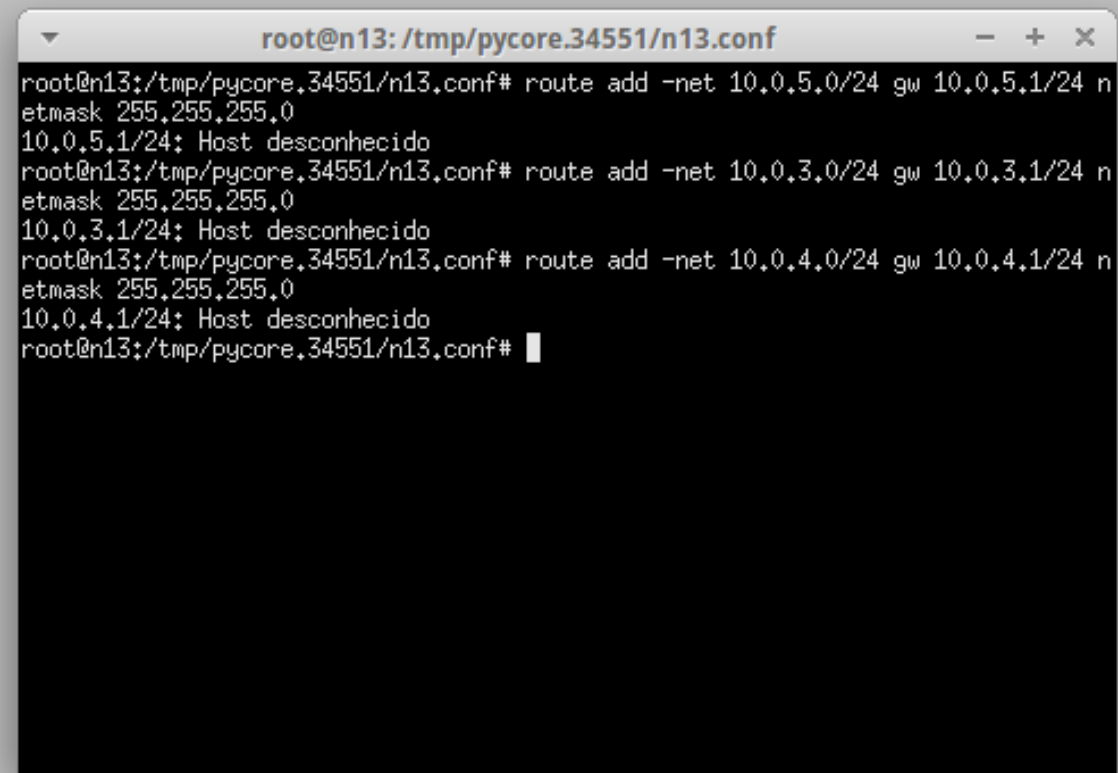
R: Está a ser usado encaminhamento estático nos três departamentos porque, além de ser uma rede de pequena dimensão, o endereçamento é baseado em rotas pré-definidas e estas rotas permanecem fixas e entre os routers dos três departamentos o encaminhamento é dinâmico.

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

```
root@n13: /tmp/pycore.40387/n13.conf
root@n13:/tmp/pycore.40387/n13.conf# route delete
Usage: inet_route [-vF] del {-host|-net} Target[/prefix] [gw Gw] [metric M] [[dev
v] If]
      inet_route [-vF] add {-host|-net} Target[/prefix] [gw Gw] [metric M]
      [netmask N] [mss Mss] [window W] [irtt I]
      [mod] [dyn] [reinstat] [[dev] If]
      inet_route [-vF] add {-host|-net} Target[/prefix] [metric M] reject
      inet_route [-FC] flush      NOT supported
root@n13:/tmp/pycore.40387/n13.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@n13:/tmp/pycore.40387/n13.conf# route delete 0.0.0.0
SIOCDELRT: Processo inexistente
root@n13:/tmp/pycore.40387/n13.conf# route delete default
root@n13:/tmp/pycore.40387/n13.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@n13:/tmp/pycore.40387/n13.conf#
```

R: A implementação desta medida não tem nenhuma implicação para os utilizadores que acedem ao servidor pois continua a existir a rota para a rede 10.0.3.0 a partir do servidor.

- d) Adicione as rotas estáticas necessárias para restaurar a conectividade para o servidor, por forma a contornar a restrição imposta em c). Utilize para o efeito o comando `route add` e registre os comandos que usou.



```
root@n13: /tmp/pycore.34551/n13.conf
root@n13:/tmp/pycore.34551/n13.conf# route add -net 10.0.5.0/24 gw 10.0.5.1/24 n
etmask 255.255.255.0
10.0.5.1/24: Host desconhecido
root@n13:/tmp/pycore.34551/n13.conf# route add -net 10.0.3.0/24 gw 10.0.3.1/24 n
etmask 255.255.255.0
10.0.3.1/24: Host desconhecido
root@n13:/tmp/pycore.34551/n13.conf# route add -net 10.0.4.0/24 gw 10.0.4.1/24 n
etmask 255.255.255.0
10.0.4.1/24: Host desconhecido
root@n13:/tmp/pycore.34551/n13.conf#
```

R: Como se pode ver pela figura acima tentamos restaurar a conectividade como pretendido, no entanto, encontramos um erro que não conseguimos ultrapassar.

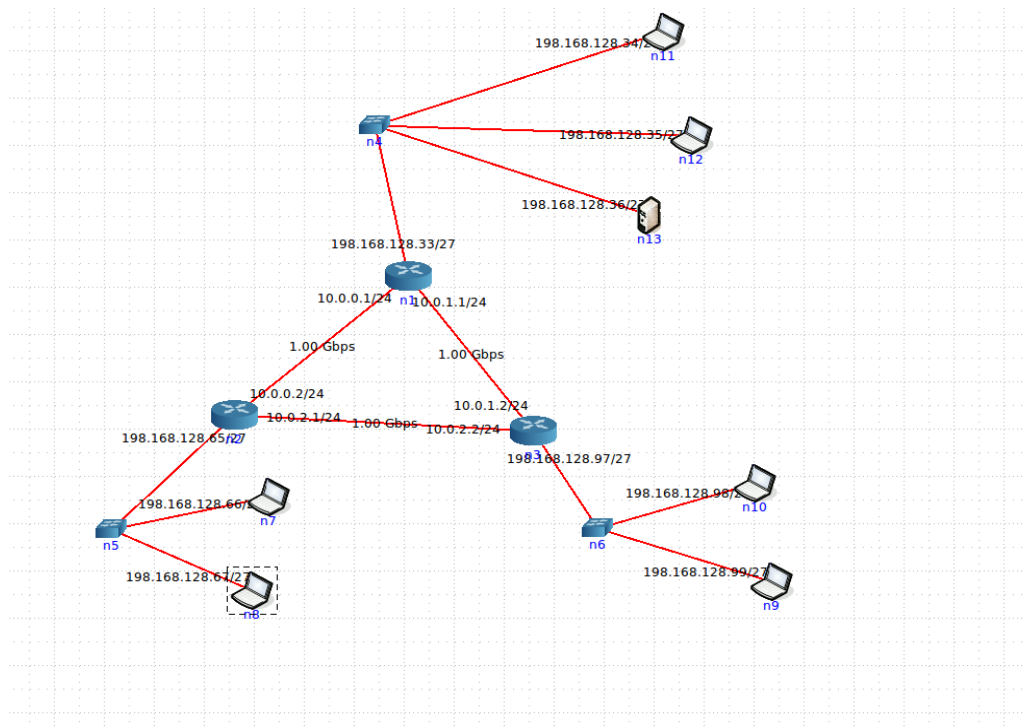
- e) Teste a nova política de encaminhamento garantindo que o servidor está novamente acessível, utilizando para o efeito o comando `ping`. Registe a nova tabela de encaminhamento do servidor.

R: Face à situação descrita anteriormente na questão d), o servidor não poderá estar acessível e a tabela de encaminhamento será aquela que obtivemos no final da questão c).

3)

- 1) Assumindo que dispõe apenas de um único endereço de rede IP classe C 192.168.128.0/24, defina um novo esquema de endereçamento para as redes dos departamentos (mantendo a rede de core inalterada) e atribua endereços às interfaces dos vários sistemas envolvidos. Deve justificar as opções usadas.

R:



192.168.128.0/24

3 sub-redes (uma por departamento)

8 bits para gerir ($32-24=8$)

Usando 3 bits, é possível definir $2^3-2=6$ sub-redes. Usando menos de 3 bits, apenas poderíamos definir 2 sub-redes ou menos ($2^2-2=2$). Para o *host*, usamos 5 bits pois permite obter $2^5-2=30$ *hosts* diferentes. Estes 5

bits são suficientes para os dispositivos que temos ligados e permite um bom crescimento desta sub-rede a nível do número de *hosts* ligados.

SR1: 198.168.128.32/27 -> 198.168.33 até .62

SR2: 198.168.128.64/27 -> 198.168.128.65 até .94

SR3: 198.168.128.96/27 -> 198.168.128.97 até .126

SR1 – departamento A

n1: 198.168.128.33/27

n11: 198.168.128.34/27

n12: 198.168.128.35/27

n13: 198.168.128.36/27

SR2 – departamento B

n2: 198.168.128.65/27

n7: 198.168.128.66/27

n8: 198.168.128.67/27

SR3 – departamento C

n3: 198.168.128.97/27

n9: 198.168.128.98/27

n10: 198.168.128.99/27

2) Qual a máscara de rede que usou (em formato decimal)?
Justifique.

Identificam a rede

R: 11111111.11111111.11111111.11100000₂



3 bits da sub-rede

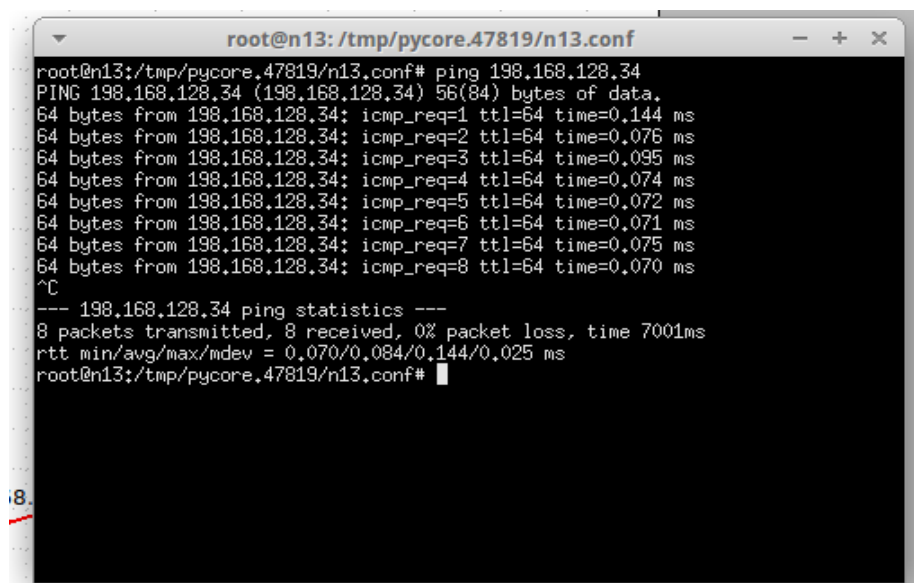
255.255.255.224₁₀. Existem 27 bits para identificar a rede, sendo 3 destes usados para especificar a sub-rede. Por isso, e como para criar a máscara de rede é necessário colocar todos os bits identificadores da rede a 1, o número obtido em decimal foi 255.255.255.224₁₀.

- 3) Quantos *hosts* IP pode interligar em cada departamento? Justifique.

R: 5 bits para *hosts* -> $2^5 = 32$. Há dois endereços reservados, logo há $32 - 2 = 30$ endereços possíveis para *hosts*. Assim, é possível interligar 30 *hosts* em cada departamento, assumindo que apenas vamos ligar *hosts*.

- 4) Garanta que conectividade IP entre as várias redes locais da empresa MIEInet é mantida.

R: Como se pode ver nas figuras abaixo, há conectividade entre todas as redes locais da empresa.



```
root@n13: /tmp/pycore.47819/n13.conf
root@n13: /tmp/pycore.47819/n13.conf# ping 198.168.128.34
PING 198.168.128.34 (198.168.128.34) 56(84) bytes of data:
64 bytes from 198.168.128.34: icmp_req=1 ttl=64 time=0.144 ms
64 bytes from 198.168.128.34: icmp_req=2 ttl=64 time=0.076 ms
64 bytes from 198.168.128.34: icmp_req=3 ttl=64 time=0.095 ms
64 bytes from 198.168.128.34: icmp_req=4 ttl=64 time=0.074 ms
64 bytes from 198.168.128.34: icmp_req=5 ttl=64 time=0.072 ms
64 bytes from 198.168.128.34: icmp_req=6 ttl=64 time=0.071 ms
64 bytes from 198.168.128.34: icmp_req=7 ttl=64 time=0.075 ms
64 bytes from 198.168.128.34: icmp_req=8 ttl=64 time=0.070 ms
^C
--- 198.168.128.34 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7001ms
rtt min/avg/max/mdev = 0.070/0.084/0.144/0.025 ms
root@n13: /tmp/pycore.47819/n13.conf#
```

Ping feito dentro da rede do departamento A

```
root@n9: /tmp/pycore.47819/n9.conf
root@n9: /tmp/pycore.47819/n9.conf# ping 198.168.128.99
PING 198.168.128.99 (198.168.128.99) 56(84) bytes of data.
64 bytes from 198.168.128.99: icmp_req=1 ttl=64 time=0.047 ms
64 bytes from 198.168.128.99: icmp_req=2 ttl=64 time=0.050 ms
64 bytes from 198.168.128.99: icmp_req=3 ttl=64 time=0.096 ms
64 bytes from 198.168.128.99: icmp_req=4 ttl=64 time=0.046 ms
64 bytes from 198.168.128.99: icmp_req=5 ttl=64 time=0.110 ms
64 bytes from 198.168.128.99: icmp_req=6 ttl=64 time=0.028 ms
64 bytes from 198.168.128.99: icmp_req=7 ttl=64 time=0.047 ms
64 bytes from 198.168.128.99: icmp_req=8 ttl=64 time=0.048 ms
64 bytes from 198.168.128.99: icmp_req=9 ttl=64 time=0.047 ms
64 bytes from 198.168.128.99: icmp_req=10 ttl=64 time=0.048 ms
64 bytes from 198.168.128.99: icmp_req=11 ttl=64 time=0.044 ms
64 bytes from 198.168.128.99: icmp_req=12 ttl=64 time=0.042 ms
64 bytes from 198.168.128.99: icmp_req=13 ttl=64 time=0.054 ms
^C
--- 198.168.128.99 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12000ms
rtt min/avg/max/mdev = 0.028/0.054/0.110/0.022 ms
root@n9: /tmp/pycore.47819/n9.conf#
```

Ping feito dentro da rede do departamento A para uma máquina do departamento C

```
root@n8: /tmp/pycore.47819/n8.conf
root@n8: /tmp/pycore.47819/n8.conf# ping 198.168.128.98
PING 198.168.128.98 (198.168.128.98) 56(84) bytes of data.
64 bytes from 198.168.128.98: icmp_req=1 ttl=62 time=0.115 ms
64 bytes from 198.168.128.98: icmp_req=2 ttl=62 time=0.105 ms
64 bytes from 198.168.128.98: icmp_req=3 ttl=62 time=0.105 ms
64 bytes from 198.168.128.98: icmp_req=4 ttl=62 time=0.114 ms
64 bytes from 198.168.128.98: icmp_req=5 ttl=62 time=0.154 ms
64 bytes from 198.168.128.98: icmp_req=6 ttl=62 time=0.222 ms
64 bytes from 198.168.128.98: icmp_req=7 ttl=62 time=0.346 ms
64 bytes from 198.168.128.98: icmp_req=8 ttl=62 time=0.119 ms
64 bytes from 198.168.128.98: icmp_req=9 ttl=62 time=0.123 ms
^C
--- 198.168.128.98 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 7998ms
rtt min/avg/max/mdev = 0.105/0.155/0.346/0.077 ms
root@n8: /tmp/pycore.47819/n8.conf#
```

Ping feito dentro da rede do departamento B para uma máquina do departamento C


```
root@n7: /tmp/pycore.47819/n7.conf
root@n7:/tmp/pycore.47819/n7.conf# ping 198.168.128.66
PING 198.168.128.66 (198.168.128.66) 56(84) bytes of data.
64 bytes from 198.168.128.66: icmp_req=1 ttl=64 time=0.044 ms
64 bytes from 198.168.128.66: icmp_req=2 ttl=64 time=0.041 ms
64 bytes from 198.168.128.66: icmp_req=3 ttl=64 time=0.043 ms
64 bytes from 198.168.128.66: icmp_req=4 ttl=64 time=0.045 ms
64 bytes from 198.168.128.66: icmp_req=5 ttl=64 time=0.044 ms
64 bytes from 198.168.128.66: icmp_req=6 ttl=64 time=0.046 ms
64 bytes from 198.168.128.66: icmp_req=7 ttl=64 time=0.050 ms
64 bytes from 198.168.128.66: icmp_req=8 ttl=64 time=0.047 ms
64 bytes from 198.168.128.66: icmp_req=9 ttl=64 time=0.048 ms
64 bytes from 198.168.128.66: icmp_req=10 ttl=64 time=0.048 ms
64 bytes from 198.168.128.66: icmp_req=11 ttl=64 time=0.046 ms
64 bytes from 198.168.128.66: icmp_req=12 ttl=64 time=0.061 ms
64 bytes from 198.168.128.66: icmp_req=13 ttl=64 time=0.056 ms
^C
--- 198.168.128.66 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 11997ms
rtt min/avg/max/mdev = 0.041/0.047/0.061/0.009 ms
root@n7:/tmp/pycore.47819/n7.conf#
```

Ping feito dentro da rede do departamento A para uma máquina do departamento B