

TP2

Camada de ligação Lógica: Ethernet e Protocolo ARP

André Freitas A74619
Joel Morais A70841
Sofia Carvalho A76658

PARTE 1

1. Qual é o endereço MAC da interface ativa do seu computador?

```
> Frame 53: 452 bytes on wire (3616 bits), 452 bytes captured (3616 bits) on interface 0
  Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    > Destination: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    > Source: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
```

R: O nosso endereço MAC é 2c:56:dc:b0:d0:89.

2. Qual é o endereço MAC destino da trama? A que sistema é destinada essa trama, será o endereço Ethernet do servidor http para cesium.di.uminho.pt? Justifique.

```
> Frame 53: 452 bytes on wire (3616 bits), 452 bytes captured (3616 bits) on interface 0
  Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    > Destination: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    > Source: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
```

R: O endereço MAC destino da trama é 00:0c:29:d2:19:f0. A trama é destinada ao router que vai estar ligado à rede do departamento, acedendo ao http server e enviando a resposta.

3. Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?

```
> Destination: Vmware_d2
> Source: AsustekC_b0:d0
  Type: IPv4 (0x0800)
```

R: O valor hexadecimal do campo Type da trama Ethernet é 0x0800, o que corresponde ao tipo de dados do IP que vai ser encapsulado.

4. Quantos bytes são usados desde o início da trama até ao caractere ASCII “G” do método HTTP GET? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar no envio do HTTP GET.

```

51 7.710344 192.168.100.201 193.136.19.148 TCP 54 65048+80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
52 7.710476 192.168.100.201 193.136.19.148 TCP 54 65049+80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
53 7.711258 192.168.100.201 193.136.19.148 HTTP 452 GET / HTTP/1.1

Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
    Host: cesium.di.uminho.pt\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2826.150 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate, sdch\r\n
    Accept-Language: pt-PT,pt;q=0.8,en-US;q=0.6,en;q=0.4\r\n

0030  01 00 fc 36 00 00 47 45 54 20 2f 20 48 54 54 50 ...6..GET / HTTP

```

R: Desde o início da trama até ao carácter ASCII “G” do método HTTP GET, são usados 53 bytes. O tamanho da trama é 452, logo o overhead será dado por $53/452 = 0,117 = 11,7 \%$

5. Em ligações com fios pouco susceptíveis a erros, nem sempre as NICs geram o código de detecção de erros. Verifique se o campo FCS está a ser utilizado. Aceda à opção Edit/Preferences/Protocols/Ethernet e indique que é assumido o uso do campo FCS. Verifique qual o valor hexadecimal desse campo na trama capturada. Que conclui? Reponha a configuração a configuração inicial.

Frame check sequence: 0x0d0a0d0a [incorrect, should be 0x932a78b4]

R: Não está a utilizar FCS, pois caso estivesse a utilizar a informação que corresponde a este iria aparecer a seguir à informação do tipo de nível 2. O valor que está na imagem corresponde aos 4 últimos bytes da trama. Concluimos que ao fazer a mudança vai dar como incorreto pois, como não foi utilizado FCS, não tem os bytes de verificação de erro como era suposto ter.

6. Qual é o endereço Ethernet da fonte? A que sistema de rede corresponde? Justifique.

```
> Frame 103: 630 bytes on wire (5040 bits), 630 bytes captured (5040 bits) on interface 0
▼ Ethernet II, Src: Vmware_d2:19:f0 (00:0c:29:d2:19:f0), Dst: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
  ▼ Destination: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    Address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    .... 0. .... = LG bit: Globally unique address (factory default)
    .... 0. .... = IG bit: Individual address (unicast)
  ▼ Source: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    Address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    .... 0. .... = LG bit: Globally unique address (factory default)
    .... 0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 193.136.19.148, Dst: 192.168.100.201
0000  2c 56 dc b0 d0 89 00 0c 29 d2 19 f0 08 00 45 00  ,V.....)....E.
0010  02 68 61 19 40 00 3f 06 dd e8 c1 88 13 94 c0 a8  .ha.@.?. ....
0020  64 c9 00 50 fe 19 c1 33 c1 22 cc 54 59 35 50 18  d..P...3 ".TY5P.
```

R: O endereço Ethernet da fonte é 00:0c:29:d2:19:f0. Este pertence à máquina virtual responsável pelo endereçamento da página, pois ao abrirmos a página cesium.uminho.pt, o pedido vai ter de ser enviado para a rede que suporta o site, e a resposta provém deste.

7. Qual é o endereço MAC do destino? A que sistema corresponde?

```
> Frame 103: 630 bytes on wire (5040 bits), 630 bytes captured (5040 bits) on interface 0
▼ Ethernet II, Src: Vmware_d2:19:f0 (00:0c:29:d2:19:f0), Dst: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
  ▼ Destination: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    Address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    .... 0. .... = LG bit: Globally unique address (factory default)
    .... 0. .... = IG bit: Individual address (unicast)
  ▼ Source: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    Address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    .... 0. .... = LG bit: Globally unique address (factory default)
    .... 0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 193.136.19.148, Dst: 192.168.100.201
0000  2c 56 dc b0 d0 89 00 0c 29 d2 19 f0 08 00 45 00  ,V.....)....E.
0010  02 68 61 19 40 00 3f 06 dd e8 c1 88 13 94 c0 a8  .ha.@.?. ....
0020  64 c9 00 50 fe 19 c1 33 c1 22 cc 54 59 35 50 18  d..P...3 ".TY5P.
```

R: O endereço MAC do destino é 2c:56:dc:b0:d0:89, que vai corresponder ao endereço MAC do computador que efetuou o pedido anteriormente.

8. Qual é o valor hexadecimal do campo tipo (Type)?

```
> Frame 103: 630 bytes on wire (5040 bits), 630 bytes captured (5040 bits) on interface 0
▼ Ethernet II, Src: Vmware_d2:19:f0 (00:0c:29:d2:19:f0), Dst: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
  ▼ Destination: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    Address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
      ....0. .... = LG bit: Globally unique address (factory default)
      ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
    Address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
      ....0. .... = LG bit: Globally unique address (factory default)
      ....0. .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 193.136.19.148, Dst: 192.168.100.201
0000  2c 56 dc b0 d0 89 00 0c 29 d2 19 f0 08 00 45 00  ,V.....)....E.
0010  02 68 61 19 40 00 3f 06 dd e8 c1 88 13 94 c0 a8  .ha.@.?.....
0020  64 c9 00 50 fe 19 c1 33 c1 22 cc 54 59 35 50 18  d..P...3..".TY5P.
```

R: O valor hexadecimal do campo Type é 0X0800.

9. Que tipo de resposta foi enviada pelo servidor?

No.	Time	Source	Destination	Protocol	Length	Info
53	7.711258	192.168.100.201	193.136.19.148	HTTP	452	GET / HTTP/1.1
59	7.734841	193.136.19.148	192.168.100.201	HTTP	74	HTTP/1.1 200 OK (text/html)
61	7.755894	192.168.100.201	193.136.19.148	HTTP	799	GET /assets/welcome-360743b13d4ca
71	7.758797	192.168.100.201	193.136.19.148	HTTP	806	GET /assets/logo-130aa2b71d4ca
103	7.762997	193.136.19.148	192.168.100.201	HTTP	630	HTTP/1.1 200 OK (PNG)
288	7.783523	192.168.100.201	216.58.211.202	HTTP	457	GET /maps/api/js HTTP/1.1
449	7.799725	193.136.19.148	192.168.100.201	HTTP	657	HTTP/1.1 200 OK (text/css)
452	7.809028	192.168.100.201	193.136.19.148	HTTP	783	GET /assets/welcome-1fa929694t

```
> Frame 59: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: Vmware_d2:19:f0 (00:0c:29:d2:19:f0), Dst: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
> Internet Protocol Version 4, Src: 193.136.19.148, Dst: 192.168.100.201
> Transmission Control Protocol, Src Port: 80, Dst Port: 65048, Seq: 4479, Ack: 399, Len: 20
> [5 Reassembled TCP Segments (4498 bytes): #55(1460), #56(751), #57(1460), #58(807), #59(20)]
▼ Hypertext Transfer Protocol
  ▼ HTTP/1.1 200 OK\r\n
    ▼ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
```

R: O tipo de resposta que foi enviado pelo servidor foi um "OK".

Protocolo ARP

10. Observe o conteúdo da tabela ARP. Diga o que significa cada uma das colunas?

```
C:\Users\André>arp -a

Interface: 192.168.56.1 --- 0x7
Internet Address      Physical Address      Type
192.168.56.255        ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.100.201 --- 0xa
Internet Address      Physical Address      Type
192.168.100.254        00-0c-29-d2-19-f0     dynamic
192.168.100.255        ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.250       01-00-5e-7f-ff-fa     static
255.255.255.255        ff-ff-ff-ff-ff-ff     static

C:\Users\André>
```

R: A primeira coluna (Internet Address) corresponde ao endereço IP, a segunda coluna (Physical Address) corresponde ao MAC Address e a última coluna (Type) refere-se ao tipo, que pode ser estático ou dinâmico.

11. Qual é o valor hexadecimal dos endereços origem e destino na trama Ethernet que contém a mensagem com o pedido ARP (ARP Request)? Como interpreta e justifica o endereço destino usado?

```
✓ Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ✓ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    .... 1. .... = LG bit: Locally administered address (this is NOT the f
    .... 1. .... = IG bit: Group address (multicast/broadcast)
  ✓ Source: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    Address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    .... 0. .... = LG bit: Globally unique address (factory default)
    .... 0. .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
  > Address Resolution Protocol (request)
```

0000	ff ff ff ff ff 2c 56 dc b0 d0 89 08 06 00 01V.....
0010	08 00 06 04 00 01 2c 56 dc b0 d0 89 c0 a8 64 c9,V.....d.
0020	00 00 00 00 00 00 c0 a8 64 fed.

R: O valor hexadecimal do endereço de origem é 2c:56:dc:b0:d0:89 e o de destino é ff:ff:ff:ff:ff:ff. Isto indica que estamos a fazer um broadcast para a camada 2.

12. Qual o valor hexadecimal do campo tipo da trama Ethernet? O que indica?

```
▼ Ethernet II, Src: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    .... ..1. .... = LG bit: Locally administered address (1
    .... ..1 .... = IG bit: Group address (multicast/broad
  ▼ Source: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    Address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
    .... ..0. .... = LG bit: Globally unique address (factor
    .... ..0 .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
  > Address Resolution Protocol (request)
```

0000	ff ff ff ff ff ff 2c 56 dc b0 d0 89 08 06 00 01,Vd.
0010	08 00 06 04 00 01 2c 56 dc b0 d0 89 c0 a8 64 c9,Vd.
0020	00 00 00 00 00 00 c0 a8 64 fe d.

R: O valor hexadecimal do campo do tipo Ethernet é 0x0806, que significa que o ARP do tipo de Ethernet está associado a esse valor.

13. Qual o valor do campo ARP opcode? O que especifica? Se necessário, consulte a RFC do protocolo ARP (<http://tools.ietf.org/html/rfc826.html>).

```
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
  Sender IP address: 192.168.100.201
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.254
```

0000	ff ff ff ff ff ff 2c 56 dc b0 d0 89 08 06 00 01,Vd.
0010	08 00 06 04 00 01 2c 56 dc b0 d0 89 c0 a8 64 c9,Vd.
0020	00 00 00 00 00 00 c0 a8 64 fe d.

R: O valor do campo ARP opcode é 00 01, que nos diz que isto é relativo a um request ou a uma reply (ou seja, vai ser uma flag). Neste caso, vai ser um request como visto em cima [request (1)].

14. A mensagem ARP contém o endereço IP de origem? Que tipo de pergunta é feita?

No.	Time	Source	Destination	Protocol	Length	Info
13	5.117977	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.254? Tell 192.168.100.201
12	10.121	AsustekC_b0:d0:89	Vmware_d2:19:f0	ARP	42	192.168.100.201 is at 2c:56:dc:b0:d0:89
16	20.916	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.173? Tell 192.168.100.201
17	25.931	AsustekC_b0:d0:89	SonyCorp_1f:0b:1c	ARP	42	192.168.100.201 is at 2c:56:dc:b0:d0:89
12	11.567	FujitsuT_b6:b3:38	Broadcast	ARP	60	Who has 192.168.100.254? Tell 192.168.100.193
16	24.732	Pegatron_8f:e7:0f	Broadcast	ARP	60	Who has 192.168.100.173? Tell 192.168.100.222
16	20.917	SonyCorp_1f:0b:1c	AsustekC_b0:d0:89	ARP	60	192.168.100.173 is at 00:1d:ba:1f:0b:1c
17	25.931	SonyCorp_1f:0b:1c	AsustekC_b0:d0:89	ARP	60	Who has 192.168.100.201? Tell 192.168.100.173
14	5.118754	Vmware_d2:19:f0	AsustekC_b0:d0:89	ARP	60	192.168.100.254 is at 00:0c:29:d2:19:f0
12	10.121	Vmware_d2:19:f0	AsustekC_b0:d0:89	ARP	60	Who has 192.168.100.201? Tell 192.168.100.254

R: O nosso endereço IP está de facto contido na mensagem ARP, como podemos observar na imagem acima, estando este endereço a seguir a “Tell” (endereço IP da nossa máquina). Nós conectámos a nossa máquina a uma máquina dos nossos colegas. Para fazermos isso, enviamos um request (broadcast) a todas as máquinas que estão ligadas à nossa rede Ethernet a perguntar quem tem o IP 192.168.100.254 (“Who has 192.168.100.254?”), obtendo assim o seu endereço MAC para o conectarmos ao nosso.

15. Localize a mensagem ARP que é a resposta ao pedido ARP efetuado.

No.	Time	Source	Destination	Protocol	Length	Info
13	5.117977	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.254? Tell 192.168.100.201
12	10.121	AsustekC_b0:d0:89	Vmware_d2:19:f0	ARP	42	192.168.100.201 is at 2c:56:dc:b0:d0:89
16	20.916	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.173? Tell 192.168.100.201
17	25.931	AsustekC_b0:d0:89	SonyCorp_1f:0b:1c	ARP	42	192.168.100.201 is at 2c:56:dc:b0:d0:89
12	11.567	FujitsuT_b6:b3:38	Broadcast	ARP	60	Who has 192.168.100.254? Tell 192.168.100.193
16	24.732	Pegatron_8f:e7:0f	Broadcast	ARP	60	Who has 192.168.100.173? Tell 192.168.100.222
16	20.917	SonyCorp_1f:0b:1c	AsustekC_b0:d0:89	ARP	60	192.168.100.173 is at 00:1d:ba:1f:0b:1c
17	25.931	SonyCorp_1f:0b:1c	AsustekC_b0:d0:89	ARP	60	Who has 192.168.100.201? Tell 192.168.100.173
14	5.118754	Vmware_d2:19:f0	AsustekC_b0:d0:89	ARP	60	192.168.100.254 is at 00:0c:29:d2:19:f0
12	10.121	Vmware_d2:19:f0	AsustekC_b0:d0:89	ARP	60	Who has 192.168.100.201? Tell 192.168.100.254

a. Qual o valor do campo ARP opcode? O que especifica?

▼ Address Resolution Protocol (reply)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2)

Sender MAC address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)

Sender IP address: 192.168.100.254

Target MAC address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)

Target IP address: 192.168.100.201

```
0000 2c 56 dc b0 d0 89 00 0c 29 d2 19 f0 08 06 00 01 ,V.....).....
0010 08 00 06 04 00 02 00 0c 29 d2 19 f0 c0 a8 64 fe ....d.....d.
```


R: A resposta que nós procuramos é o endereço MAC relativo ao endereço IP procurado ("00:0c:29:d2:19:f0"), como é possível observar na primeira imagem.

Relativamente ao valor do campo ARP opcode, vai ser 00 02. Ao estarmos a especificar o reply (2), estamos a ir buscar o valor do broadcast que fizemos.

b. Em que posição da mensagem ARP está a resposta ao pedido ARP?

▼ Address Resolution Protocol (reply)

Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
Sender IP address: 192.168.100.254
Target MAC address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
Target IP address: 192.168.100.201

0000	2c 56 dc b0 d0 89 00 0c 29 d2 19 f0 08 06 00 01	,V.....).....
0010	08 00 06 04 00 02 00 0c 29 d2 19 f0 c0 a8 64 fe).....d.
0020	2c 56 dc b0 d0 89 c0 a8 64 c9 00 00 00 00 00 00	,V..... d.....

R: A resposta ao pedido ARP vai estar no "Target MAC address" (2c:56:dc:b0:d0:89).

16. Quais são os valores hexadecimais para os endereços origem e destino da trama que contém a resposta ARP? Que conclui?

▼ Address Resolution Protocol (reply)

Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
Sender IP address: 192.168.100.254
Target MAC address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
Target IP address: 192.168.100.201

0000	2c 56 dc b0 d0 89 00 0c 29 d2 19 f0 08 06 00 01	,V.....).....
0010	08 00 06 04 00 02 00 0c 29 d2 19 f0 c0 a8 64 fe).....d.

R: Os valores para os endereços origem que contém a resposta ARP são c0 a8 64 fe.

```
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
  Sender IP address: 192.168.100.254
  Target MAC address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
  Target IP address: 192.168.100.201
```

0000	2c 56 dc b0 d0 89 00 0c 29 d2 19 f0 08 06 00 01	,V.....).....
0010	08 00 06 04 00 02 00 0c 29 d2 19 f0 c0 a8 64 fe).....d.
0020	2c 56 dc b0 d0 89 c0 a8 64 c9 00 00 00 00 00 00	,V.... .. d.....

R: E os valores para o endereço destino da trama são c0 a8 64 c9.

Ou seja, é possível ligar diretamente a nossa máquina através do endereço IP.

17. Com auxílio do comando `ifconfig` obtenha os endereços Ethernet das interfaces dos diversos routers.

```
n1.eth0.210 Link encap:Ethernet HWaddr 0a:1b:8b:23:6b:1e
  inet6 addr: fe80::81b:8bff:fe23:6b1e/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:14 errors:0 dropped:0 overruns:0 frame:0
  TX packets:45 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:1244 (1.2 KB) TX bytes:8370 (8.3 KB)

n2.eth0.210 Link encap:Ethernet HWaddr 5e:fe:90:6d:92:8d
  inet6 addr: fe80::5cfe:90ff:fe6d:928d/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:14 errors:0 dropped:0 overruns:0 frame:0
  TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:1240 (1.2 KB) TX bytes:8264 (8.2 KB)

n2.eth1.210 Link encap:Ethernet HWaddr 86:0a:5e:53:23:62
  inet6 addr: fe80::840a:5eff:fe53:2362/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:15 errors:0 dropped:0 overruns:0 frame:0
  TX packets:46 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:1310 (1.3 KB) TX bytes:8444 (8.4 KB)

n3.eth0.210 Link encap:Ethernet HWaddr fa:88:0f:20:f1:a9
  inet6 addr: fe80::f888:fff:fe20:f1a9/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:15 errors:0 dropped:0 overruns:0 frame:0
  TX packets:45 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:1314 (1.3 KB) TX bytes:8330 (8.3 KB)

core@XubunCORE:~$
```

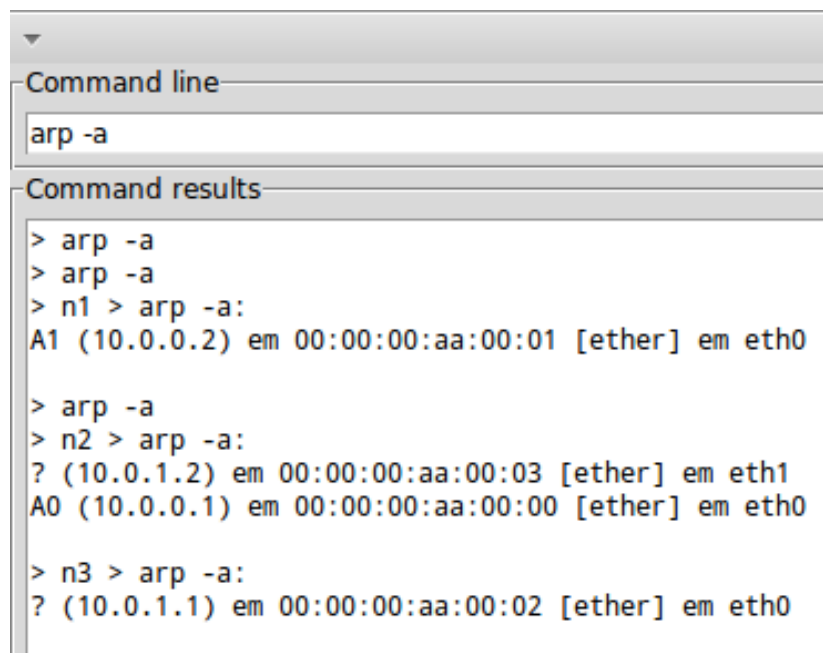
R: Os endereços Ethernet das interfaces dos diversos routers são:

n1: ee:aa:d8:c3:ce:b1

n2: 96:e8:b0:fb:27:e4 e a2:ba:cf:80:b3:2f

n3: ae:70:35:ce:0a:0d

18. Usando o comando arp obtenha as caches arp dos diversos sistemas.



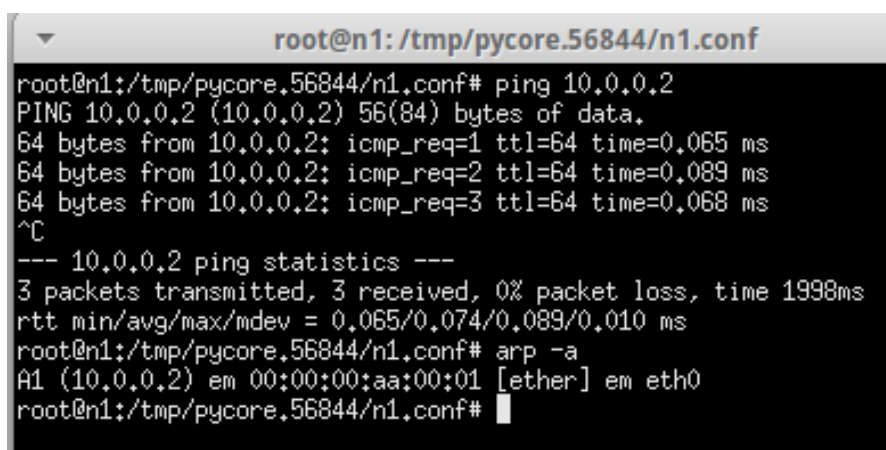
```
Command line
arp -a

Command results
> arp -a
> arp -a
> n1 > arp -a:
A1 (10.0.0.2) em 00:00:00:aa:00:01 [ether] em eth0

> arp -a
> n2 > arp -a:
? (10.0.1.2) em 00:00:00:aa:00:03 [ether] em eth1
A0 (10.0.0.1) em 00:00:00:aa:00:00 [ether] em eth0

> n3 > arp -a:
? (10.0.1.1) em 00:00:00:aa:00:02 [ether] em eth0
```

19. Faça ping de n1 para n2. Que modificações observa nas caches ARP desses sistemas? Faça ping de n1 para n3. Consulte as caches ARP. Que conclui?



```
root@n1: /tmp/pycore.56844/n1.conf
root@n1:/tmp/pycore.56844/n1.conf# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_req=1 ttl=64 time=0.065 ms
64 bytes from 10.0.0.2: icmp_req=2 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_req=3 ttl=64 time=0.068 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.065/0.074/0.089/0.010 ms
root@n1:/tmp/pycore.56844/n1.conf# arp -a
A1 (10.0.0.2) em 00:00:00:aa:00:01 [ether] em eth0
root@n1:/tmp/pycore.56844/n1.conf#
```

R: n1 para n2

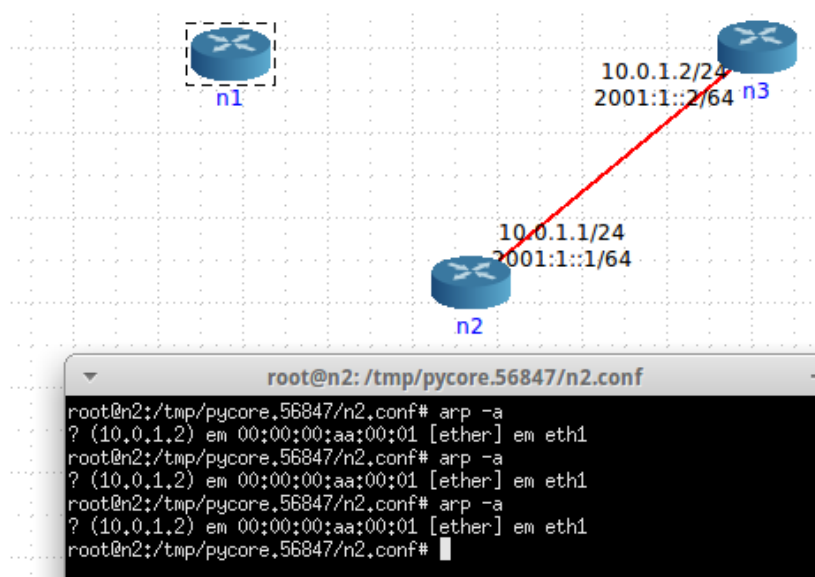
Pela análise da cache de n1, é possível reparar que o endereço 10.0.0.2 está ligado a 00:00:00:aa:00:01.

```
root@n1: /tmp/pycore.56844/n1.conf
root@n1:/tmp/pycore.56844/n1.conf# ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_req=1 ttl=63 time=0.091 ms
64 bytes from 10.0.1.2: icmp_req=2 ttl=63 time=0.086 ms
64 bytes from 10.0.1.2: icmp_req=3 ttl=63 time=0.088 ms
64 bytes from 10.0.1.2: icmp_req=4 ttl=63 time=0.086 ms
64 bytes from 10.0.1.2: icmp_req=5 ttl=63 time=0.087 ms
^C
--- 10.0.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.086/0.087/0.091/0.010 ms
root@n1:/tmp/pycore.56844/n1.conf# arp -a
A1 (10.0.0.2) em 00:00:00:aa:00:01 [ether] em eth0
root@n1:/tmp/pycore.56844/n1.conf#
```

R: n1 para n3

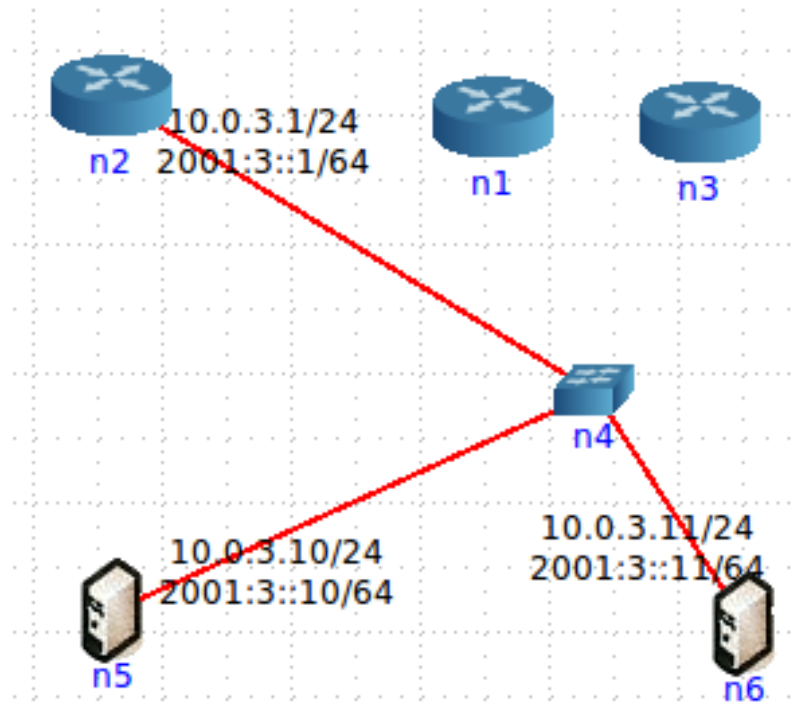
Pela análise da cache de n1, é possível reparar que o endereço 10.0.1.2 está ligado a 00:00:00:aa:00:01.

20. Em n1 remova a entrada correspondente a n2. Coloque uma nova entrada para n2 com endereço Ethernet inexistente. O que acontece?



R: Pela imagem, podemos ver que apenas a interface n3 aparece, sendo o endereço que colocamos (ff:ff:ff:ff:ff) ignorado.

21. Faça ping de n5 para n6. Sem consultar a tabela ARP anote a entrada que, em sua opinião, é criada na tabela ARP de n5. Verifique,



justificando, se a sua interpretação sobre a operação da rede Ethernet e protocolo ARP estava correto.

```
root@n5:/tmp/pycore.56848/n5.conf# ping 10.0.3.11
PING 10.0.3.11 (10.0.3.11) 56(84) bytes of data.
64 bytes from 10.0.3.11: icmp_req=1 ttl=64 time=0.129 ms
64 bytes from 10.0.3.11: icmp_req=2 ttl=64 time=0.079 ms
64 bytes from 10.0.3.11: icmp_req=3 ttl=64 time=0.074 ms
64 bytes from 10.0.3.11: icmp_req=4 ttl=64 time=0.069 ms
64 bytes from 10.0.3.11: icmp_req=5 ttl=64 time=0.085 ms
^C
--- 10.0.3.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.069/0.087/0.129/0.022 ms
root@n5:/tmp/pycore.56848/n5.conf#
```

R: No ARP da bash de n5 deverá aparecer o IP de n6 e à frente o endereço MAC de n6.

```
root@n5: /tmp/pycore.56848/n5.conf
root@n5:/tmp/pycore.56848/n5.conf# arp -a
? (10.0.3.11) em 00:00:00:aa:00:00 [ether] em eth0
root@n5:/tmp/pycore.56848/n5.conf#
```

```
root@n6: /tmp/pycore.56848/n6.conf
root@n6:/tmp/pycore.56848/n6.conf# arp -a
? (10.0.3.10) em 00:00:00:aa:00:01 [ether] em eth0
root@n6:/tmp/pycore.56848/n6.conf#
```

R: Pela imagem apresentada acima, é possível verificar que a nossa interpretação feita estava correta. Em n6, aparece também uma nova entrada com o IP de n5.

PARTE 2

ARP Gratuito

1. Identifique um pacote de pedido ARP gratuito originado pelo seu sistema. Verifique quantos pacotes ARP gratuito foram enviados e com que intervalo temporal?

No.	Time	Source	Destination	Protocol	Length	Info
3	0.166813	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.153? Tell 192.168.100.254
8	1.167849	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.153? Tell 192.168.100.254
10	2.220759	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.153? Tell 192.168.100.254
12	3.220771	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.153? Tell 192.168.100.254
15	4.220673	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.153? Tell 192.168.100.254
17	5.273646	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.153? Tell 192.168.100.254
20	6.273710	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.153? Tell 192.168.100.254
24	15.100...	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.254? Tell 192.168.100.201
34	15.544...	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.201? Tell 0.0.0.0
38	15.548...	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.192? Tell 192.168.100.254
42	16.044...	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.254? Tell 192.168.100.201
44	16.045...	Vmware_d2:19:f0	AsustekC_b0:d...	ARP	60	192.168.100.254 is at 00:0c:29:d2:19:f0
58	16.544...	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.201? Tell 0.0.0.0
96	17.544...	AsustekC_b0:d0:89	Broadcast	ARP	42	Who has 192.168.100.201? Tell 0.0.0.0
119	18.544...	AsustekC_b0:d0:89	Broadcast	ARP	42	Gratuitous ARP for 192.168.100.201 (Request)
135	19.340...	HewlettP_72:5d:ce	Broadcast	ARP	60	Who has 192.168.100.254? Tell 192.168.100.171

R: Foi enviado um pacote ARP gratuito que demorou 18.5s.

2. Analise o conteúdo de um pedido ARP gratuito e identifique em que se distingue dos restantes pedidos ARP. Registe a trama Ethernet correspondente. Qual o resultado esperado face ao pedido ARP gratuito enviado?

ARP Gratuito

Sender MAC address: AsustekC_b0:d0:89 (2c:56:dc:b0:d0:89)
Sender IP address: 192.168.100.201
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.100.201

ARP Normal

Sender MAC address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
Sender IP address: 192.168.100.254
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.100.192

R: Num ARP Normal, é procurado o MAC correspondente a um certo IP. No Gratuito, a máquina questiona-se a si própria sobre qual o MAC que corresponde ao próprio IP, para descobrir se tem mais alguma máquina a usar o nosso IP.

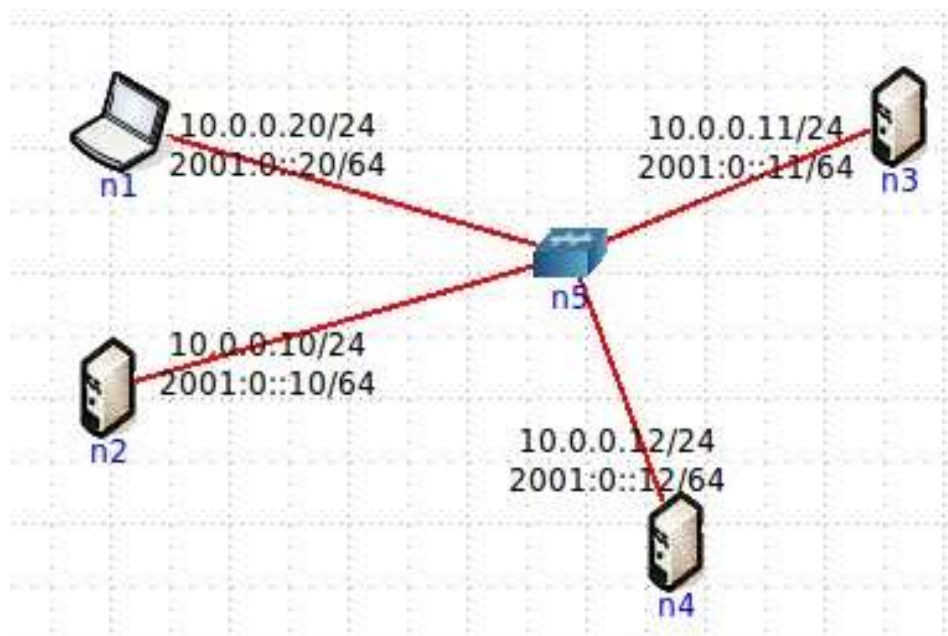
A trama Ethernet que corresponde ao ARP Gratuito é a seguinte:

0000	ff ff ff ff ff ff 2c 56 dc b0 d0 89 08 06 00 01,V
0010	08 00 06 04 00 01 2c 56 dc b0 d0 89 c0 a8 64 c9,Vd.
0020	00 00 00 00 00 00 c0 a8 64 c9 d.

O resultado esperado face ao pedido ARP Gratuito enviado será “verdadeiro” se a nossa máquina for a única a utilizar este endereço IP.

Domínios de Colisão

1. Faça ping de n1 para n2. Verifique com a opção tcpdump como flui o tráfego nas diversas interfaces dos vários dispositivos. Que conclui?




```
root@n1: /tmp/pycore.49992/n1.conf
64 bytes from 10.0.0.10: icmp_req=455 ttl=64 time=0.092 ms
64 bytes from 10.0.0.10: icmp_req=456 ttl=64 time=0.091 ms
64 bytes from 10.0.0.10: icmp_req=457 ttl=64 time=0.089 ms
64 bytes from 10.0.0.10: icmp_req=458 ttl=64 time=0.100 ms
64 bytes from 10.0.0.10: icmp_req=459 ttl=64 time=0.095 ms
64 bytes from 10.0.0.10: icmp_req=460 ttl=64 time=0.098 ms
64 bytes from 10.0.0.10: icmp_req=461 ttl=64 time=0.092 ms
64 bytes from 10.0.0.10: icmp_req=462 ttl=64 time=0.000 ms
64 bytes from 10.0.0.10: icmp_req=463 ttl=64 time=0.106 ms
64 bytes from 10.0.0.10: icmp_req=464 ttl=64 time=0.102 ms
64 bytes from 10.0.0.10: icmp_req=465 ttl=64 time=0.223 ms
64 bytes from 10.0.0.10: icmp_req=466 ttl=64 time=0.562 ms
64 bytes from 10.0.0.10: icmp_req=467 ttl=64 time=0.697 ms
64 bytes from 10.0.0.10: icmp_req=468 ttl=64 time=0.249 ms
64 bytes from 10.0.0.10: icmp_req=469 ttl=64 time=0.103 ms
64 bytes from 10.0.0.10: icmp_req=470 ttl=64 time=0.103 ms
64 bytes from 10.0.0.10: icmp_req=471 ttl=64 time=0.260 ms
64 bytes from 10.0.0.10: icmp_req=472 ttl=64 time=0.099 ms
64 bytes from 10.0.0.10: icmp_req=473 ttl=64 time=0.087 ms
64 bytes from 10.0.0.10: icmp_req=474 ttl=64 time=0.105 ms
64 bytes from 10.0.0.10: icmp_req=475 ttl=64 time=0.095 ms
64 bytes from 10.0.0.10: icmp_req=476 ttl=64 time=0.096 ms
64 bytes from 10.0.0.10: icmp_req=477 ttl=64 time=0.102 ms

vcmd
gth 64
19:40:04.398889 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 472, len
h 64
19:40:05.399557 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 473, len
gth 64
19:40:05.399580 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 473, len
h 64
19:40:06.400004 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 474, len
gth 64
19:40:06.400035 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 474, len
h 64
19:40:07.400790 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 475, len
gth 64
19:40:07.400816 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 475, len
h 64
19:40:08.402687 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 476, len
gth 64
19:40:08.402713 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 476, len
h 64
19:40:09.402789 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 477, len
gth 64
19:40:09.402819 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 477, len
h 64

vcmd
gth 64
19:40:05.399590 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 473, len
h 64
19:40:06.400000 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 474, len
gth 64
19:40:06.400045 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 474, len
h 64
19:40:07.400787 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 475, len
gth 64
19:40:07.400825 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 475, len
h 64
19:40:08.402683 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 476, len
gth 64
19:40:08.402722 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 476, len
h 64
19:40:09.402785 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 477, len
gth 64
19:40:09.402828 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 477, len
h 64
19:40:10.403881 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 41, seq 478, len
gth 64
19:40:10.404067 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 41, seq 478, len
h 64
```

R: No terminal do canto superior esquerdo, está o ping do computador n1 para o host n2, no superior direito está o tcpdump de n2 e em baixo encontra-se o tcpdump de n3. Como temos um hub, há partilha de meio e o pedido vai fluir por todas as máquinas conectadas ao hub. O ping não foi para o servidor n3 mas este vai receber na mesma a trama enviada no meio. O pedido vai, por isso, fluir por todas as máquinas conectadas ao hub.

2. Na topologia de rede substitua o hub por um switch. Repita os procedimentos que realizou na pergunta anterior. Comente os resultados obtidos quanto à utilização de hubs e switches no contexto de controlar ou dividir domínios de colisão. Documente as suas observações e conclusões com base no tráfego observado/capturado.

```

root@n1:/tmp/pycore.49893/n1.conf
64 bytes from 10.0.0.12: icmp_req=55 ttl=64 time=0.075 ms
64 bytes from 10.0.0.12: icmp_req=56 ttl=64 time=0.078 ms
64 bytes from 10.0.0.12: icmp_req=57 ttl=64 time=0.076 ms
64 bytes from 10.0.0.12: icmp_req=58 ttl=64 time=0.081 ms
64 bytes from 10.0.0.12: icmp_req=59 ttl=64 time=0.073 ms
64 bytes from 10.0.0.12: icmp_req=60 ttl=64 time=0.072 ms
64 bytes from 10.0.0.12: icmp_req=61 ttl=64 time=0.070 ms
64 bytes from 10.0.0.12: icmp_req=62 ttl=64 time=0.081 ms
64 bytes from 10.0.0.12: icmp_req=63 ttl=64 time=0.077 ms
64 bytes from 10.0.0.12: icmp_req=64 ttl=64 time=0.081 ms
64 bytes from 10.0.0.12: icmp_req=65 ttl=64 time=0.117 ms
64 bytes from 10.0.0.12: icmp_req=66 ttl=64 time=0.116 ms
64 bytes from 10.0.0.12: icmp_req=67 ttl=64 time=0.076 ms
64 bytes from 10.0.0.12: icmp_req=68 ttl=64 time=0.077 ms
64 bytes from 10.0.0.12: icmp_req=69 ttl=64 time=0.080 ms
64 bytes from 10.0.0.12: icmp_req=70 ttl=64 time=0.090 ms
64 bytes from 10.0.0.12: icmp_req=71 ttl=64 time=0.122 ms
64 bytes from 10.0.0.12: icmp_req=72 ttl=64 time=0.076 ms
64 bytes from 10.0.0.12: icmp_req=73 ttl=64 time=0.080 ms
64 bytes from 10.0.0.12: icmp_req=74 ttl=64 time=0.087 ms
64 bytes from 10.0.0.12: icmp_req=75 ttl=64 time=0.082 ms
64 bytes from 10.0.0.12: icmp_req=76 ttl=64 time=0.082 ms
64 bytes from 10.0.0.12: icmp_req=77 ttl=64 time=0.077 ms

vcmd
th 64
19:46:26.343414 IP 10.0.0.12 > 10.0.0.20: ICMP echo reply, id 28, seq 72, length 64
19:46:27.342425 IP 10.0.0.20 > 10.0.0.12: ICMP echo request, id 28, seq 73, length 64
19:46:27.342449 IP 10.0.0.12 > 10.0.0.20: ICMP echo reply, id 28, seq 73, length 64
19:46:28.343366 IP 10.0.0.20 > 10.0.0.12: ICMP echo request, id 28, seq 74, length 64
19:46:28.343388 IP 10.0.0.12 > 10.0.0.20: ICMP echo reply, id 28, seq 74, length 64
19:46:29.342519 IP 10.0.0.20 > 10.0.0.12: ICMP echo request, id 28, seq 75, length 64
19:46:29.342542 IP 10.0.0.12 > 10.0.0.20: ICMP echo reply, id 28, seq 75, length 64
19:46:30.342479 IP 10.0.0.20 > 10.0.0.12: ICMP echo request, id 28, seq 76, length 64
19:46:30.342503 IP 10.0.0.12 > 10.0.0.20: ICMP echo reply, id 28, seq 76, length 64
19:46:31.342481 IP 10.0.0.20 > 10.0.0.12: ICMP echo request, id 28, seq 77, length 64
19:46:31.342503 IP 10.0.0.12 > 10.0.0.20: ICMP echo reply, id 28, seq 77, length 64

vcmd
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
19:46:09.359941 IP6 fe80::e400:45ff:fed1:f463.5353 > ff02::fb.5353: 0 [6q] PTR (QM)? _afpovertcp._tcp.local. PTR (QM)? _ftp._tcp.local. PTR (QM)? _webdav._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _smb._tcp.local. (107)
19:46:09.807728 IP6 fe80::ac4c:8ff:fee5:7941.5353 > ff02::fb.5353: 0 [6q] PTR (QM)? _afpovertcp._tcp.local. PTR (QM)? _ftp._tcp.local. PTR (QM)? _webdav._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _smb._tcp.local. (107)

```

R: No terminal do canto superior esquerdo, está o ping do computador n1 para o host n2, no superior direito está o tcpdump de n2 e em baixo encontra-se o tcpdump de n3. Como temos um switch, não há partilha de meio e o pedido vai apenas passar pelas máquinas envolvidas no ping. O ping não foi para o host n3, por isso, este não vai receber a trama. O pedido flui apenas pelas máquinas diretamente envolvidas nele.

Os switches eliminam as colisões, conectando cada dispositivo a uma porta do comutador, enquanto que os hubs permitem que estas colisões possam existir. Ao utilizar um hub para comunicar entre um computador e um host, também passava tráfego pelas outras máquinas do esquema, no entanto, ao mudarmos para o switch, observamos que apenas passava tráfego pelo computador e pelo host ao efetuar o ping, eliminando assim, o tráfego pelas outras máquinas e evitando colisões.

Conclusões

Neste nosso trabalho de Redes de Computadores foram abordados vários temas, incluindo a camada de ligação lógica, direcionada para a tecnologia Ethernet, e do protocolo ARP, aprendendo como funcionam os endereços MAC e outros subjects essenciais abordados no nosso relatório.

Mais especificamente em relação à ligação lógica falámos de transferência de dados, deteção e correção de erros, protocolos de acesso de controlo de ligação, endereços MAC (como dito anteriormente), Address Resolution Protocol, Ethernet e interligação de redes locais.

Aprendemos também como funcionam as tramas Ethernet, respondendo às várias questões propostas pelo docente.

Por último, na parte 2 do protocolo ARP abordamos o ARP gratuito e domínios de colisão. E com o CORE fomos funcionando com os vários tipos de ligação, as diferenças entre HUB's e Switch's.