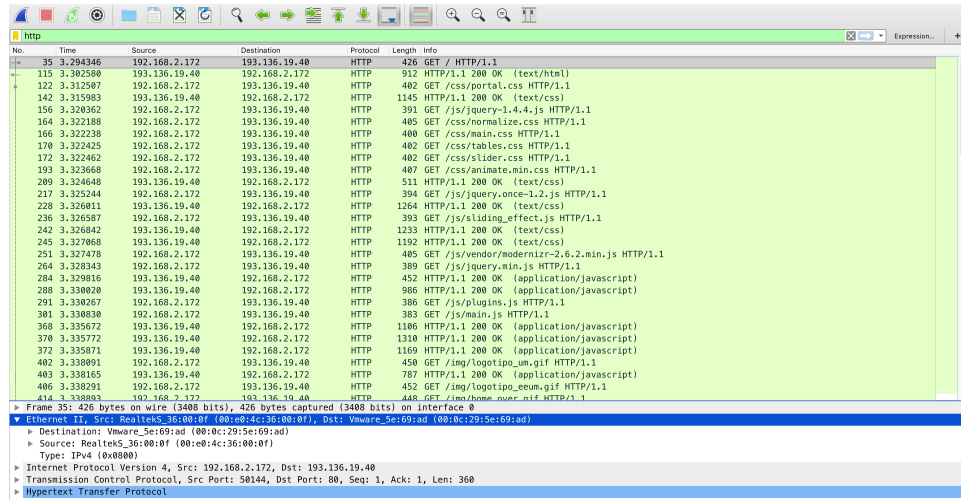


TP3
Camada de Ligação Lógica: Ethernet e Protocolo ARP

Rafael Silva, José Ramos, and Luís Ferreira

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {a74264,a73855,a76936}@alunos.uminho.pt

1 Captura e análise de Tramas Ethernet



No.	Time	Source	Destination	Protocol	Length	Info
35	3.294346	192.168.2.172	193.136.19.40	HTTP	426	GET / HTTP/1.1
115	3.302500	193.136.19.40	192.168.2.172	HTTP	912	HTTP/1.1 200 OK (text/html)
122	3.312507	192.168.2.172	193.136.19.40	HTTP	402	GET /css/portal.css HTTP/1.1
142	3.315983	193.136.19.40	192.168.2.172	HTTP	1145	HTTP/1.1 200 OK (text/css)
156	3.320362	192.168.2.172	193.136.19.40	HTTP	391	GET /js/jquery-1.4.4.js HTTP/1.1
164	3.322108	192.168.2.172	193.136.19.40	HTTP	405	GET /css/normalize.css HTTP/1.1
166	3.322238	192.168.2.172	193.136.19.40	HTTP	400	GET /css/main.css HTTP/1.1
170	3.322425	192.168.2.172	193.136.19.40	HTTP	402	GET /css/tables.css HTTP/1.1
172	3.322462	192.168.2.172	193.136.19.40	HTTP	402	GET /css/slider.css HTTP/1.1
193	3.323668	192.168.2.172	193.136.19.40	HTTP	407	GET /css/animate.min.css HTTP/1.1
209	3.324648	193.136.19.40	192.168.2.172	HTTP	511	HTTP/1.1 200 OK (text/css)
217	3.325244	192.168.2.172	193.136.19.40	HTTP	394	GET /js/jquery.once-1.2.js HTTP/1.1
228	3.326011	193.136.19.40	192.168.2.172	HTTP	1264	HTTP/1.1 200 OK (text/css)
236	3.326507	192.168.2.172	193.136.19.40	HTTP	393	GET /js/sliding.effect.js HTTP/1.1
242	3.326842	193.136.19.40	192.168.2.172	HTTP	1233	HTTP/1.1 200 OK (text/css)
245	3.327068	193.136.19.40	192.168.2.172	HTTP	1192	HTTP/1.1 200 OK (text/css)
251	3.327478	192.168.2.172	193.136.19.40	HTTP	485	GET /js/vendor/modernizr-2.6.2.min.js HTTP/1.1
264	3.328343	192.168.2.172	193.136.19.40	HTTP	389	GET /js/jquery.min.js HTTP/1.1
284	3.329816	193.136.19.40	192.168.2.172	HTTP	452	HTTP/1.1 200 OK (application/javascript)
288	3.330020	193.136.19.40	192.168.2.172	HTTP	906	HTTP/1.1 200 OK (application/javascript)
291	3.330267	192.168.2.172	193.136.19.40	HTTP	386	GET /js/plugins.js HTTP/1.1
301	3.330530	192.168.2.172	193.136.19.40	HTTP	383	GET /js/main.js HTTP/1.1
368	3.335672	193.136.19.40	192.168.2.172	HTTP	1106	HTTP/1.1 200 OK (application/javascript)
370	3.335772	193.136.19.40	192.168.2.172	HTTP	1310	HTTP/1.1 200 OK (application/javascript)
372	3.335871	193.136.19.40	192.168.2.172	HTTP	1169	HTTP/1.1 200 OK (application/javascript)
402	3.338091	192.168.2.172	193.136.19.40	HTTP	450	GET /img/looptipo_eem.gif HTTP/1.1
403	3.338165	193.136.19.40	192.168.2.172	HTTP	787	HTTP/1.1 200 OK (application/javascript)
406	3.338291	192.168.2.172	193.136.19.40	HTTP	452	GET /img/looptipo_eem.gif HTTP/1.1
414	3.339003	192.168.2.172	193.136.19.40	HTTP	448	GET /img/home_over_nid HTTP/1.1

Frame 35: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface 0

Ethernet II, Src: RealtekS_36:00:0f (00:e0:4c:36:00:0f), Dst: Vmware_5e:69:ad (00:0c:29:5e:69:ad)

Destination: Vmware_5e:69:ad (00:0c:29:5e:69:ad)

Source: RealtekS_36:00:0f (00:e0:4c:36:00:0f)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.2.172, Dst: 193.136.19.40

Transmission Control Protocol, Src Port: 58144, Dst Port: 80, Seq: 1, Ack: 1, Len: 360

Hypertext Transfer Protocol

Figura 1. Captura do trafego no WireShark e aplicação do filtro HTTP

1.1 Questão 1

Questão: Anote os endereços MAC de origem e de destino da trama capturada.

Resposta:

- Destination: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
- Source: RealtekS_36:00:0f (00:e0:4c:36:00:0f)

Figura 2. Endereço MAC da origem e destino do frame 35

1.2 Questão 2

Questão: Identifique a que sistemas se referem. Justifique.

Resposta: Estes sistemas referem-se a um equipamento secundário e não ao endereço Ethernet do servidor, ou seja, são endereços que correspondem a equipamentos espalhados pelo Departamento que encaminham a rede até ao servidor.

1.3 Questão 3

Questão: Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?

Resposta: O valor do campo *Type* é **IPv4(0x0800)** e significa que o tipo de endereçamento a nível lógico é **IPv4**.

```
> Frame 35: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface 0
  > Ethernet II, Src: RealtekS_36:00:0f (00:e0:4c:36:00:0f), Dst: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
    > Destination: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
    > Source: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
    > Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.2.172, Dst: 193.136.19.40
  > Transmission Control Protocol, Src Port: 50144, Dst Port: 80, Seq: 1, Ack: 1, Len: 360
  > Hypertext Transfer Protocol
```

Figura 3. Valor hexadecimal do campo *Type*

1.4 Questão 4

Questão: 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.

Resposta: O tamanho total é : **426 bytes**. O tamanho até ao caracter ”G” é : **66 bytes**. Portanto o overhead introduzido pela pilha protocolar é: **66 / 426 = 0.15492957746**, que em termos de percentagem significa cerca de **16%** do tamanho total.

```
> Frame 35: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface 0
  > Ethernet II, Src: RealtekS_36:00:0f (00:e0:4c:36:00:0f), Dst: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
  > Internet Protocol Version 4, Src: 192.168.2.172, Dst: 193.136.19.40
  > Transmission Control Protocol, Src Port: 50144, Dst Port: 80, Seq: 1, Ack: 1, Len: 360
  > Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
      [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      [GET / HTTP/1.1\r\n]
      [Severity Level: Chat]
      [Group: Sequence]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
      Host: miel.dl.uninho.pt\r\n
      Upgrade-Insecure-Requests: 1\r\n
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
      User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_1) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0.1 Safari/605.1.15\r\n
      Accept-Encoding: gzip, deflate\r\n
      Content-Type: text/html
0000  00 0c 29 5e 69 ad 00 e0 4c 36 00 0f 08 00 45 02  ...L6...E
0010  01 9c 00 00 40 00 40 06 a1 55 c0 a8 02 ac c1 88  ...@...U...
0020  13 28 c3 e0 00 50 93 55 32 ea bb 50 69 82 80 18  ...PU2...P...
0030  00 0a c4 f5 00 00 01 01 00 0a 09 2a 65 92 84 8c  ...e...
0040  cf f4 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 2e  GET / HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20 6d 69 85 69 2e 64 69 2e  Host: miel.dl
0060  75 6d 69 6e 68 6f 2e 70 74 2e 6d 6a 55 70 67 72  uninho.p...Upgr
0070  61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 65 71  ade-Inse cure-Req
0080  75 65 73 74 73 3a 20 31 0d 0a 41 63 63 65 70 74  uests: 1--Accept
0090  3a 20 74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c  : text/h tml,appl
00a0  69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d  ication/ xhtml+xml
00b0  6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 6d  /,applic ation/xml
00c0  6c 3b 71 3d 30 2e 39 2c 2a 2f 2a 3b 71 3d 30 2e  l;q=0.9, /*;q=0.
00d0  38 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 8-User- Agent: M
00e0  6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 4d 61 63 69  ozilla/5.0 (Maci
00f0  6e 74 6f 73 68 3b 20 49 6e 74 65 6c 20 4d 61 63  ntosh; I ntel Mac
0100  20 4f 53 20 58 20 31 30 5f 31 34 5f 31 29 20 41  OS X 10 _14.1) A
0110  70 70 6c 65 57 65 62 4b 69 74 2f 36 30 35 2e 31  plineWebK it/605.1
0120  2e 31 35 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65  .15 (KHT ML, like
0130  20 47 65 63 6b 6f 29 20 56 65 72 73 69 6f 6e 2f  Gecko) Version/
0140  31 32 2e 30 2e 31 28 53 61 66 61 72 69 2f 36 30  12.0.1 S aafari/60
0150  35 2e 31 2e 31 35 0d 0a 41 63 63 65 70 74 2d 4c  5.1.15-- Accept-L
0160  61 6e 67 73 6f 65 3a 20 70 74 2d 70 74 8d 0a  anguage: pt-pt--
0170  41 63 63 65 70 74 2d 4c 6e 63 6f 64 69 6e 67 3a  Accept-E ncoding:
0180  20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d 0a  gzip, d eflate--
0190  43 6f 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70  -connecti on: keep
01a0  2d 61 6c 69 76 65 0d 0a 0d 0a  -alive:--
```

Figura 4. Calculo dos bytes usados desde o início da trama até ao caractere ASCII G

1.5 Questão 5

Questão: Através de visualização direta de uma trama capturada, verifique que, possivelmente, o campo FCS (Frame Check Sequence) usado para detecção de erros não está a ser usado. Em sua opinião, porque será?

Resposta: O método **FCS** nao foi usado, pois o Wireshark assume que foi usado, no entanto nao foi o caso, portanto é por isso que da o erro de checksum e a trama teria de ser rejeitada.

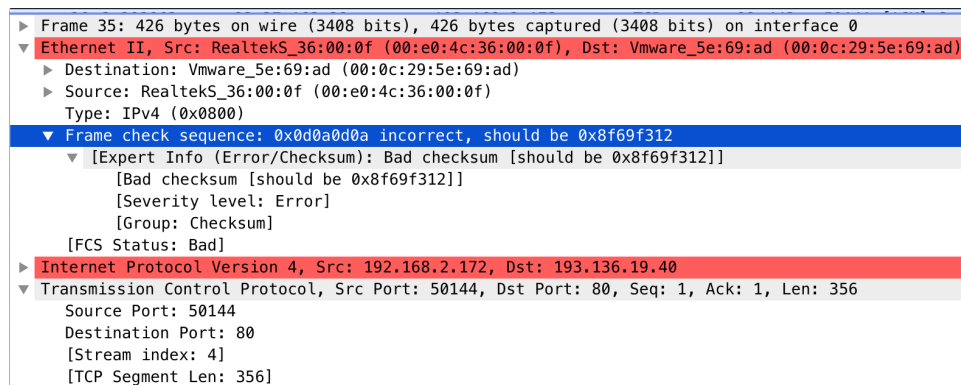


Figura 5. Verificação do campo FCS

1.6 Questão 6

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

Resposta: O endereço Ethernet da fonte é **00:0c:29:5e:69:ad**. Este pertence à máquina virtual responsável pelo endereçamento da página, pois ao abrirmos a página <http://miei.di.uminho.pt>, o pedido vai ter de ser enviado para a rede que suporta o site, e a resposta provém deste.

► Source: Vmware_5e:69:ad (00:0c:29:5e:69:ad)

Figura 6. Endereço *Ethernet* da fonte

1.7 Questão 7

Questão: Qual é o endereço MAC do destino? A que sistema corresponde?

Resposta: O endereço MAC do destino é **00:e0:4c:36:00:0f**, que vai corresponder ao endereço MAC do computador que efetuou o pedido anteriormente.

```
► Destination: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
```

Figura 7. Endereço *Ethernet* do destino

1.8 Questão 8

Questão: Atendendo ao conceito de desencapsulamento protocolar, identifique os vários protocolos contidos na trama recebida.

Resposta: Os protocolos que estão contidos na trama recebida são: o protocolo **IPv4**, o protocolo **TCP** e o protocolo **HTTP**.

```
► Frame 35: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface 0
▼ Ethernet II, Src: RealtekS_36:00:0f (00:e0:4c:36:00:0f), Dst: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
  ► Destination: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
  ► Source: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
  Type: IPv4 (0x0800)
► Internet Protocol Version 4, Src: 192.168.2.172, Dst: 193.136.19.40
► Transmission Control Protocol, Src Port: 50144, Dst Port: 80, Seq: 1, Ack: 1, Len: 360
▼ Hypertext Transfer Protocol
  ► GET / HTTP/1.1\r\n
    Host: miei.di.uminho.pt.\r\n
    Upgrade-Insecure-Requests: 1\r\n
```

Figura 8. Diferentes protocolos do frame 35

2 Protocolo ARP

2.1 Questão 9

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

```
? (172.26.254.254) at 0:d0:3:ff:94:0 on en0 ifscope [ethernet]
? (172.26.255.255) at ff:ff:ff:ff:ff:ff on en0 ifscope [ethernet]
? (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
```

Figura 9.

Resposta: A primeira coluna diz respeito aos endereços IP, a segunda coluna diz respeito aos endereços MAC e a terceira coluna diz respeito ao tipo de atribuição do endereço IP que pode ser dinâmico ou estático. Como estamos no sistema operativo MacOS o tipo ifscope é dinâmico e o ifscope permanent é estático.

2.2 Questão 10

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?

No.	Time	Source	Destination	Protocol	Length	Info
147	11.900382	Vmware_5e:69:ad	RealtekS_36:00:0f	ARP	60	Who has 192.168.2.172? Tell 192.168.2.1
148	11.900438	RealtekS_36:00:0f	Vmware_5e:69:ad	ARP	42	192.168.2.172 is at 00:e0:4c:36:00:0f

▶ Frame 147: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

▼ Ethernet II, Src: Vmware_5e:69:ad (00:0c:29:5e:69:ad), Dst: RealtekS_36:00:0f (00:e0:4c:36:00:0f)

▶ Destination: RealtekS_36:00:0f (00:e0:4c:36:00:0f)

▶ Source: Vmware_5e:69:ad (00:0c:29:5e:69:ad)

Type: ARP (0x0806)

Padding: 00000000000000000000000000000000

▼ Address Resolution Protocol (request)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

Sender MAC address: Vmware_5e:69:ad (00:0c:29:5e:69:ad)

Sender IP address: 192.168.2.1

Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)

Target IP address: 192.168.2.172

Figura 10. Endereços origem e destino da trama Ethernet com o pedido ARP

Resposta: O endereço da origem é VmWare_5 e:69:ad(00:0c:29:5e:69:ad) e o endereço de destino é Realtek_36:00:0f(00:e0:4c:36:00:0f) o que indica que o pedido foi executado com uma transmissão Broadcast, tal que todos os dispositivos conectados a esta rede recebem este pedido. Em suma, para descobrir o endereço IP de cada interface faz todo o sentido fazer uma transmissão Broadcast pois assim todos os dispositivos recebem a mensagem.

2.3 Questão 11

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

Type: ARP (0x0806)

Figura 11.

Resposta: Type: 0x0806, indica-nos que é do tipo ARP.

2.4 Questão 12

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>.

No.	Time	Source	Destination	Protocol	Length	Info
147	11.900382	Vmware_5e:69:ad	RealtekS_36:00:0f	ARP	60	Who has 192.168.2.172? Tell 192.168.2.1
148	11.900438	RealtekS_36:00:0f	Vmware_5e:69:ad	ARP	42	192.168.2.172 is at 00:e0:4c:36:00:0f

▶ Frame 147: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▼ Ethernet II, Src: Vmware_5e:69:ad (00:0c:29:5e:69:ad), Dst: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
▶ Destination: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
▶ Source: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
▼ Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
Sender IP address: 192.168.2.1
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.2.172

Figura 12.

Resposta: O valor do campo ARP opcode é igual a 1, o que significa que esta mensagem é uma request.

2.5 Questão 13

Identifique que tipo de endereços estão contidos na mensagem ARP? Que conclui?

Resposta: Os endereços que podemos encontrar numa mensagem ARP referem a máquina que realiza o pedido e para a que queremos saber, isto é os endereços IP destas duas máquinas.

2.6 Questão 14

Explicita o tipo de pedido ou pergunta que é feita pelo host de origem?

Who has 192.168.2.172? Tell 192.168.2.1

Figura 13.

Resposta: A pergunta realizada aos dispositivos em rede é: Quem é o 192.168.112.253, a máquina com este IP passa a responder a 192.168.112.184 que tem esse endereço IP.

2.7 Questão 15

Localize a mensagem ARP que é a resposta ao pedido ARP efectuado.

- Qual o valor do campo ARP opcode? O que especifica?
- Em que posição da mensagem ARP está a resposta ao pedido ARP ?

No.	Time	Source	Destination	Protocol	Length	Info
147	11.900382	Vmware_5e:69:ad	RealtekS_36:00:0f	ARP	60	Who has 192.168.2.172? Tell 192.168.2.1
148	11.900438	RealtekS_36:00:0f	Vmware_5e:69:ad	ARP	42	192.168.2.172 is at 00:e0:4c:36:00:0f

▶ Frame 148: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0

▼ Ethernet II, Src: RealtekS_36:00:0f (00:e0:4c:36:00:0f), Dst: Vmware_5e:69:ad (00:0c:29:5e:69:ad)

- ▶ Destination: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
- ▶ Source: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
- Type: ARP (0x0806)

▼ Address Resolution Protocol (reply)

- Hardware type: Ethernet (1)
- Protocol type: IPv4 (0x0800)
- Hardware size: 6
- Protocol size: 4
- Opcode: reply (2)
- Sender MAC address: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
- Sender IP address: 192.168.2.172
- Target MAC address: Vmware_5e:69:ad (00:0c:29:5e:69:ad)
- Target IP address: 192.168.2.1

Figura 14.

Resposta: a) O campo opcode apresenta como valor: '2' logo é uma reply.

Resposta: b). A resposta ao pedido ARP encontra-se no **Sender MAC Address**. Por cima do Sender IP Address do IP que queríamos saber.

3 ARP Gratuito

3.1 Questão 16

Identifique um pacote de pedido ARP gratuito originado pelo seu sistema.

No.	Time	Source	Destination	Protocol	Length	Info
14	7.794802	Vmware_5e:69:ad	RealtekS_36:00:0f	ARP	60	192.168.2.1 is at 00:0c:29:5e:69:ad
15	7.795474	RealtekS_36:00:0f	Broadcast	ARP	42	Gratuitous ARP for 192.168.2.172 (Request)
16	7.798159	RealtekS_36:00:0f	Broadcast	ARP	42	Who has 169.254.255.255? Tell 192.168.2.172
17	7.828597	RealtekS_36:00:0f	Broadcast	ARP	42	Who has 192.168.2.1? Tell 192.168.2.172
18	7.828883	Vmware_5e:69:ad	RealtekS_36:00:0f	ARP	60	192.168.2.1 is at 00:0c:29:5e:69:ad
26	8.120983	RealtekS_36:00:0f	Broadcast	ARP	42	Who has 169.254.255.255? Tell 192.168.2.172

Figura 15.

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?

```
▼ Address Resolution Protocol (request/gratuitous ARP)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  [Is gratuitous: True]
  Sender MAC address: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
  Sender IP address: 192.168.2.172
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.2.172
```

Figura 16. ARP gratuito

```
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: RealtekS_36:00:0f (00:e0:4c:36:00:0f)
  Sender IP address: 192.168.2.172
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.2.1
```

Figura 17. ARP normal

Resposta: Um ARP normal procede na procura do MAC, é procurado o MAC correspondente a um dado IP. Um ARP “Gratuito”, procede a integrorração da propria maquina

sobre o seu MAC que corresponde ao próprio IP, visando descobrir se tem mais alguma máquina a usar o mesmo IP. A trama Ethernet que corresponde ao ARP Gratuito é a seguinte:

0000	ff ff ff ff ff ff 00 e0 4c 36 00 0f 08 06 00 01 L6.....
0010	08 00 06 04 00 01 00 e0 4c 36 00 0f c0 a8 02 ac L6.....
0020	00 00 00 00 00 00 c0 a8 02 ac

Figura 18. Trama de Ethernet

4 Domínios de colisão

4.1 Questão 17

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

Resposta: Neste tipo de topologia, os pacotes provenientes de n1 vão para o hub, que por sua vez os direciona em broadcast, possibilitando todos os dispositivos da rede capturarem os pacotes, mesmo que não sejam a si endereçados. Desta forma, o hub cria um único domínio de colisão.

```
root@n2:/tmp/pycore.36062/n2.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C09:56:15.677006 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 31, length 64
09:56:15.677015 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 31, length 64
09:56:16.677010 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 32, length 64
09:56:16.677019 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 32, length 64
09:56:17.676974 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 33, length 64
09:56:17.676983 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 33, length 64
09:56:18.677025 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 34, length 64
09:56:18.677034 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 34, length 64
09:56:19.676971 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 35, length 64
09:56:19.676980 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 35, length 64

10 packets captured
10 packets received by filter
0 packets dropped by kernel
root@n2:/tmp/pycore.36062/n2.conf#
```

Figura 19. Comando 'tcpdump' em n2

```
root@n3:/tmp/pycore.36062/n3.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C09:56:45.677065 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 61, length 64
09:56:45.678058 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 62, length 64
09:56:46.678074 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 62, length 64
09:56:47.678043 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 63, length 64
09:56:47.678059 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 63, length 64
09:56:48.678124 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 64, length 64
09:56:48.678140 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 64, length 64
09:56:49.677124 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 65, length 64
09:56:49.677139 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 65, length 64
09:56:50.676999 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 66, length 64
09:56:50.677015 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 66, length 64
09:56:51.677004 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 67, length 64
09:56:51.677020 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 67, length 64

14 packets captured
14 packets received by filter
0 packets dropped by kernel
root@n3:/tmp/pycore.36062/n3.conf#
```

Figura 20. Comando 'tcpdump' em n3

```

root@n4:/tmp/pycore.36062/n4.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C09:56:31.677050 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 47, length 64
09:56:31.677066 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 47, length 64
09:56:32.677040 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 48, length 64
09:56:32.677055 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 48, length 64
09:56:33.677875 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 49, length 64
09:56:33.677898 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 49, length 64
09:56:34.677041 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 50, length 64
09:56:34.677057 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 50, length 64
09:56:35.677049 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 51, length 64
09:56:35.677064 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 51, length 64
09:56:36.678588 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 52, length 64
09:56:36.678603 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 52, length 64
09:56:37.678091 IP 10.0.0.20 > A9: ICMP echo request, id 63, seq 53, length 64
09:56:37.678106 IP A9 > 10.0.0.20: ICMP echo reply, id 63, seq 53, length 64

14 packets captured
14 packets received by filter
0 packets dropped by kernel
root@n4:/tmp/pycore.36062/n4.conf# █

```

Figura 21. Comando 'tcpdump' em n4

4.2 Questão 18

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.

Resposta: Nesta topologia os pacotes vão da fonte para o switch, que por sua vez os encaminha apenas para a máquina destino, e não em broadcast. Desta forma, obtemos uma organização mais estruturada das comunicações e temos uma separação dos domínios de colisão em função das conexões que cada dispositivo da rede estabelece com o switch.

```

root@n1:/tmp/pycore.35690/n1.conf# ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data:
64 bytes from 10.0.0.10: icmp_req=1 ttl=64 time=0.033 ms
64 bytes from 10.0.0.10: icmp_req=2 ttl=64 time=0.035 ms
64 bytes from 10.0.0.10: icmp_req=3 ttl=64 time=0.033 ms
64 bytes from 10.0.0.10: icmp_req=4 ttl=64 time=0.032 ms
64 bytes from 10.0.0.10: icmp_req=5 ttl=64 time=0.034 ms
64 bytes from 10.0.0.10: icmp_req=6 ttl=64 time=0.058 ms
64 bytes from 10.0.0.10: icmp_req=7 ttl=64 time=0.043 ms
64 bytes from 10.0.0.10: icmp_req=8 ttl=64 time=0.032 ms
64 bytes from 10.0.0.10: icmp_req=9 ttl=64 time=0.034 ms
64 bytes from 10.0.0.10: icmp_req=10 ttl=64 time=0.053 ms
64 bytes from 10.0.0.10: icmp_req=11 ttl=64 time=0.033 ms
^C
--- 10.0.0.10 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.032/0.038/0.058/0.009 ms
root@n1:/tmp/pycore.35690/n1.conf# █

```

Figura 22. Comando 'ping' em n1

```

root@n2:/tmp/pycore.35690/n2.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C19:46:16.095262 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 1, length 64
19:46:16.095271 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 1, length 64
19:46:17.094917 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 2, length 64
19:46:17.094927 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 2, length 64
19:46:18.093917 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 3, length 64
19:46:18.093927 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 3, length 64
19:46:19.094256 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 4, length 64
19:46:19.094265 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 4, length 64
19:46:20.094298 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 5, length 64
19:46:20.094308 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 5, length 64
19:46:21.094475 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 6, length 64
19:46:21.094494 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 6, length 64
19:46:21.105549 ARP, Request who-has 10.0.0.20 tell A9, length 28
19:46:21.105549 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:00 (oui Ethernet), length 28
19:46:22.093466 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 7, length 64
19:46:22.093479 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 7, length 64
19:46:23.094308 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 8, length 64
19:46:23.094317 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 8, length 64
19:46:24.093412 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 9, length 64
19:46:24.093421 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 9, length 64
19:46:25.093442 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 10, length 64
19:46:25.093451 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 10, length 64
19:46:26.093411 IP 10.0.0.20 > A9: ICMP echo request, id 30, seq 11, length 64
19:46:26.093420 IP A9 > 10.0.0.20: ICMP echo reply, id 30, seq 11, length 64

24 packets captured
24 packets received by filter
0 packets dropped by kernel
root@n2:/tmp/pycore.35690/n2.conf# █

```

Figura 23. Comando 'tcpdump' em n2

```

root@n3:/tmp/pycore.35690/n3.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@n3:/tmp/pycore.35690/n3.conf# █

```

Figura 24. Comando 'tcpdump' em n3

```

root@n4:/tmp/pycore.35690/n4.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@n4:/tmp/pycore.35690/n4.conf# █

```

Figura 25. Comando 'tcpdump' em n4

5 Conclusão

A realização deste trabalho prático permitiu-nos expandir os nossos conhecimentos sobre o endereçamento Ethernet e protocolo ARP. Dada uma captura de tráfego do WireShark, somos agora capazes de indentificar os endereços MAC dos dispositivos envolvidos, assim como os seus endereços IP, a informação que passa nas tramas e o respetivo propósito desta. Conseguimos também consultar e modificar as caches ARP, permitindo-nos entender melhor como são identificados diferentes dispositivos dentro da rede e como é que as respetivas conexões são gerenciadas. Por último, através do simulador de rede CORE, conseguimos identificar as diferenças concretas entre um switch e um hub e a forma como gerem o encaminhamento de pacotes dentro da rede, permitindo-nos ter uma noção de domínios de colisão.