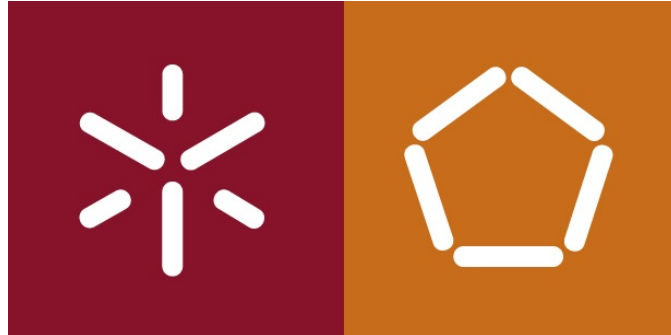


UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA



Redes de Computadores

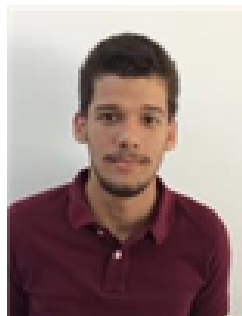
RELATÓRIO DO TRABALHO PRÁTICO 3

CAMADA DE LIGAÇÃO LÓGICA: ETHERNET E PROTOCOLO ARP

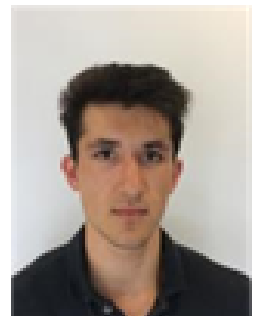
GRUPO 1



Adriana Meireles
A82582



Nuno Silva
A78156



Shahzod Yusupov
A82617

November 30, 2018

Questões e Respostas

3. Captura e análise de tramas

No.	Time	Source	Destination	Protocol	Length	Info
27	2.342514	192.168.100.185	a1089.dscd.akamai.n...	HTTP	373	GET /success.txt HTTP/1.1
30	2.349901	a1089.dscd.akamai.n...	192.168.100.185	HTTP	438	HTTP/1.1 200 OK (text/plain)
95	3.197808	192.168.100.185	93.184.220.29	OCSF	516	Request
108	3.249647	93.184.220.29	192.168.100.185	OCSF	841	Response
110	3.262521	192.168.100.185	93.184.220.29	OCSF	516	Request
113	3.313658	93.184.220.29	192.168.100.185	OCSF	842	Response
155	5.362433	192.168.100.185	www6.di.uminho.pt	HTTP	476	GET / HTTP/1.1
210	5.371975	www6.di.uminho.pt	192.168.100.185	HTTP	910	HTTP/1.1 200 OK (text/html)
213	5.424286	192.168.100.185	www6.di.uminho.pt	HTTP	451	GET /css/portal.css HTTP/1.1
229	5.428287	www6.di.uminho.pt	192.168.100.185	HTTP	977	HTTP/1.1 200 OK (text/css)
245	5.432086	192.168.100.185	www6.di.uminho.pt	HTTP	449	GET /css/main.css HTTP/1.1
246	5.432130	192.168.100.185	www6.di.uminho.pt	HTTP	451	GET /css/tables.css HTTP/1.1
247	5.432133	192.168.100.185	www6.di.uminho.pt	HTTP	454	GET /css/normalize.css HTTP/1.1
248	5.432753	192.168.100.185	www6.di.uminho.pt	HTTP	440	GET /js/jquery-1.4.4.js HTTP/1.1
249	5.432780	192.168.100.185	www6.di.uminho.pt	HTTP	451	GET /css/slider.css HTTP/1.1
250	5.432849	192.168.100.185	www6.di.uminho.pt	HTTP	456	GET /css/animate.min.css HTTP/1.1
261	5.435728	www6.di.uminho.pt	192.168.100.185	HTTP	1192	HTTP/1.1 200 OK (text/css)
264	5.436624	www6.di.uminho.pt	192.168.100.185	HTTP	1168	HTTP/1.1 200 OK (text/css)
293	5.439484	www6.di.uminho.pt	192.168.100.185	HTTP	1149	HTTP/1.1 200 OK (text/css)
304	5.440597	www6.di.uminho.pt	192.168.100.185	HTTP	487	HTTP/1.1 200 OK (text/css)
323	5.441921	192.168.100.185	www6.di.uminho.pt	HTTP	443	GET /js/jquery.once-1.2.js HTTP/1.1
324	5.442206	192.168.100.185	www6.di.uminho.pt	HTTP	442	GET /js/sliding_effect.js HTTP/1.1
354	5.445246	192.168.100.185	www6.di.uminho.pt	HTTP	454	GET /js/vendor/modernizr-2.6.2.min.js HTTP/1.1
364	5.446129	192.168.100.185	www6.di.uminho.pt	HTTP	438	GET /js/jquery.min.js HTTP/1.1
376	5.447126	www6.di.uminho.pt	192.168.100.185	HTTP	962	HTTP/1.1 200 OK (application/javascript)
382	5.447141	www6.di.uminho.pt	192.168.100.185	HTTP	416	HTTP/1.1 200 OK (application/javascript)
394	5.448150	www6.di.uminho.pt	192.168.100.185	HTTP	225	HTTP/1.1 200 OK (text/css)
405	5.449107	www6.di.uminho.pt	192.168.100.185	HTTP	127	HTTP/1.1 200 OK (application/javascript)
416	5.450551	192.168.100.185	www6.di.uminho.pt	HTTP	432	GET /js/main.js HTTP/1.1
417	5.450723	192.168.100.185	www6.di.uminho.pt	HTTP	435	GET /js/plugins.js HTTP/1.1
425	5.450968	www6.di.uminho.pt	192.168.100.185	HTTP	1178	HTTP/1.1 200 OK (application/javascript)
439	5.453709	www6.di.uminho.pt	192.168.100.185	HTTP	1157	HTTP/1.1 200 OK (application/javascript)
457	5.455086	www6.di.uminho.pt	192.168.100.185	HTTP	1094	HTTP/1.1 200 OK (application/javascript)
493	5.460174	www6.di.uminho.pt	192.168.100.185	HTTP	1485	HTTP/1.1 200 OK (application/javascript)
495	5.463279	192.168.100.185	www6.di.uminho.pt	HTTP	443	GET /img/logotipo_eum.gif HTTP/1.1
496	5.463324	192.168.100.185	www6.di.uminho.pt	HTTP	441	GET /img/logotipo_um.gif HTTP/1.1
497	5.463335	192.168.100.185	www6.di.uminho.pt	HTTP	444	GET /img/contactos_over.gif HTTP/1.1
498	5.463353	192.168.100.185	www6.di.uminho.pt	HTTP	430	GET /img/home_wm.gif HTTP/1.1

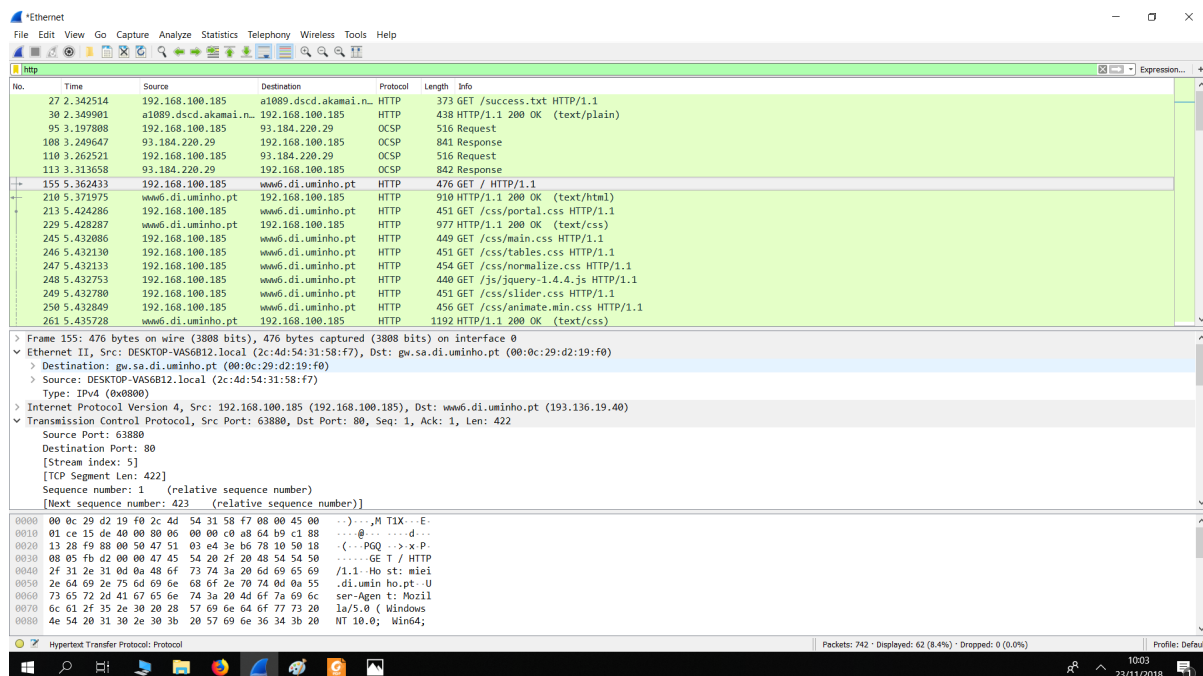
1) Anote os endereços MAC de origem e de destino da trama capturada.

Frame 155: 476 bytes on wire (3808 bits), 476 bytes captured (3808 bits) on interface 0
Ethernet II, Src: DESKTOP-VAS6812.local (2c:4d:54:31:58:f7), Dst: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0)
Destination: gw.sa.di.uminho.pt (00:0c:29:d2:19:f0)
Source: DESKTOP-VAS6812.local (2c:4d:54:31:58:f7)
Type: IPv4 (80000)
Internet Protocol Version 4, Src: 192.168.100.185 (192.168.100.185), Dst: www6.di.uminho.pt (193.136.140.40)
Transmission Control Protocol, Src Port: 63880, Dst Port: 80, Seq: 1, Ack: 1, Len: 422
Source Port: 63880
Destination Port: 80
[Stream index: 5]
[TCP Segment Len: 422]

0000 00 0c 29 d2 19 f0 2c 4d 54 31 58 f7 08 00 45 00 ...M TIX...E
0010 01 ee 15 4e 40 00 00 00 00 00 00 00 64 b4 b9 08 ...@...
0020 13 28 f9 88 00 50 47 51 03 ea 3e b6 78 10 50 18 ...PGQ...x P

O endereço de origem é o 2c:4d:54:31:58:f7 e o endereço MAC de destino é o 00:0c:29:d2:19:f0.

2) Identifique a que sistemas se referem. Justifique.

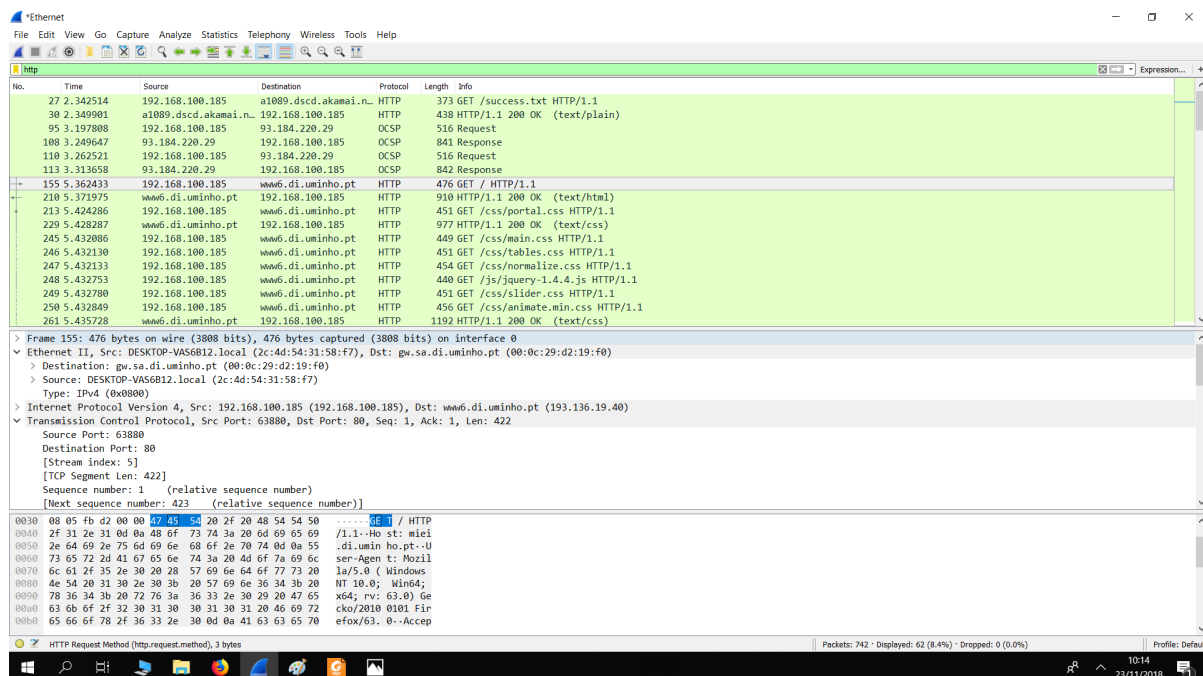


O 2c:4d:54:31:58:f7 corresponde ao endereço MAC da interface ativa do nosso computador (Source; o wireshark identifica como "DESKTOP-VAS6B12.local") e o endereço 00:0c:29:d2:19:f0 corresponde ao endereço MAC do equipamento (*Destination*; gw.sa.di.uminho.pt), ou seja, da *gateway*, que liga a rede em que estamos inseridos à rede exterior, isto é, rede em que se encontra o equipamento que detém aquele domínio.

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

Como se pode observar na figura relativa à pergunta 2, o valor hexadecimal do campo *Type* é 0x0800, o mesmo indica o tipo de dados que a trama encapsula. Neste caso, é o protocolo IPv4.

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.



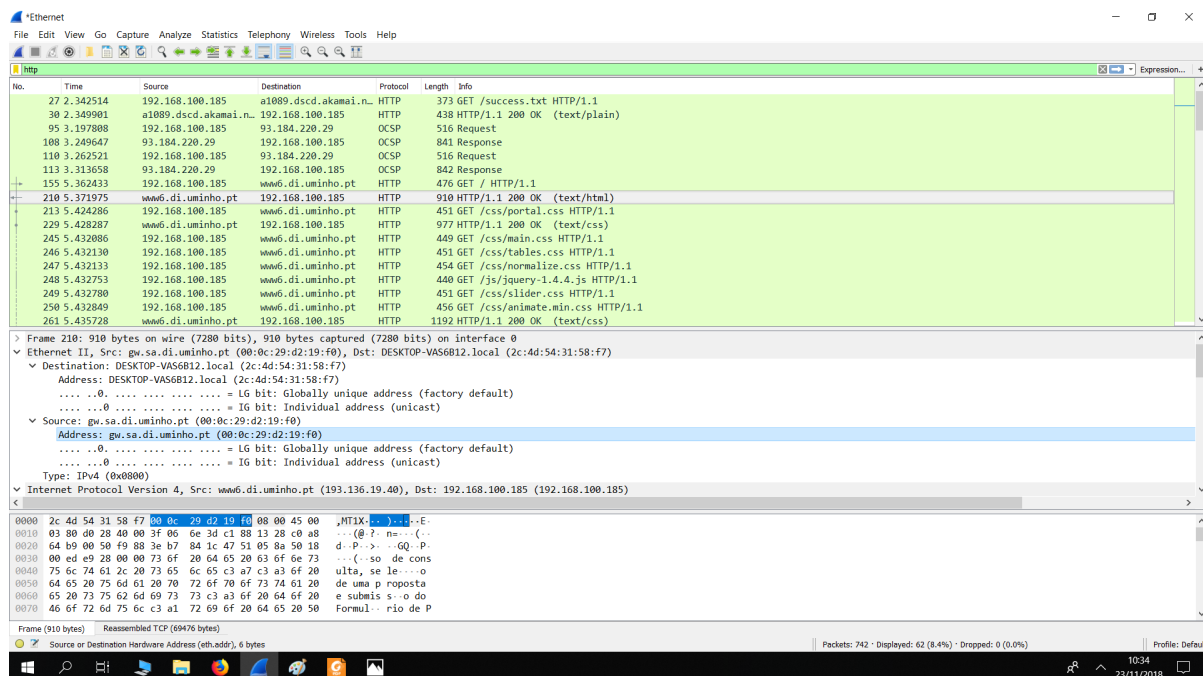
Temos desde o início até ao final da trama 476 bytes, mas apenas a partir do byte 54 teremos o *payload*. Assim o *overhead* vai do byte 0 até ao 53, o que corresponde a 54 bytes. Estes 54 bytes correspondem a $(54 \times 100 / 476) = 11.34$

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

Como podemos observar pelas figuras anteriores, o campo FCS não se encontra em EthernetII. O FCS não é usado devido à sua probabilidade de erros ser muito baixa e por esse motivo não compensar porque vai haver um maior overhead.

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

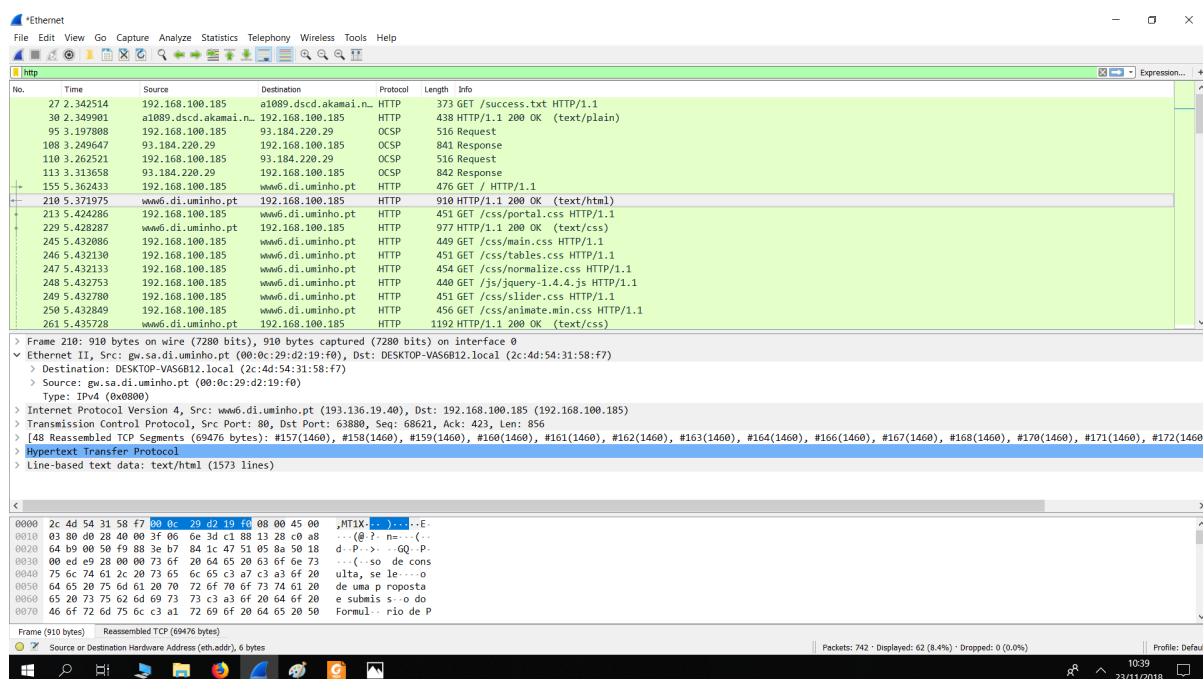
O endereço Ethernet da fonte é 00:0c:29:d2:19:f0 e corresponde ao *router* que faz a ligação da rede local com a rede exterior. À frente do campo *Source* temos até a identificação do equipamento("gw.sa.di.uminho.pt")



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

O endereço MAC do destino é o 2c:4d:54:31:58:f7 e corresponde à interface ativa do nosso computador. Para este também existe uma identificação, "DESKTOP-VAS6B12.local".

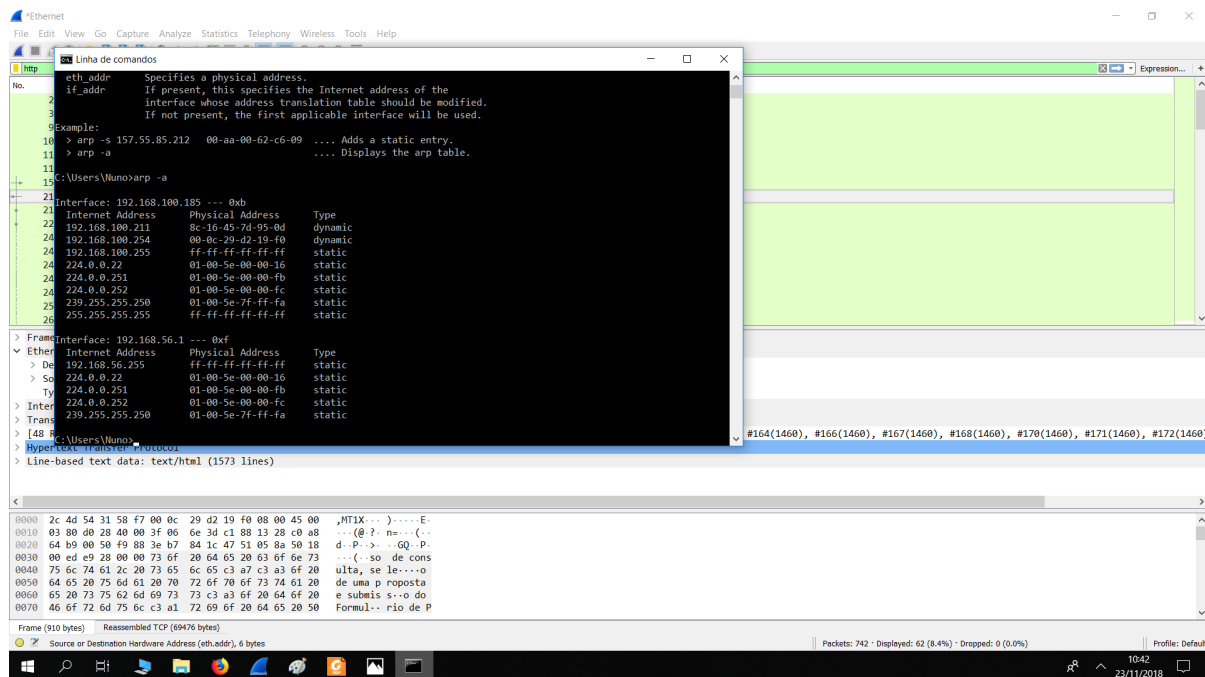
8) Atendendo ao conceito de desencapsulamento protocolar, identifique os vários protocolos contidos na trama recebida.



De acordo com a figura acima, os vários protocolos existentes na trama recebida são os seguintes: EthernetII, Internet Protocol Version 4(IPV4), Transmission Control Protocol(TCP) e Hypertext Transfer Protocol(HTP)

4. Protocolo ARP

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

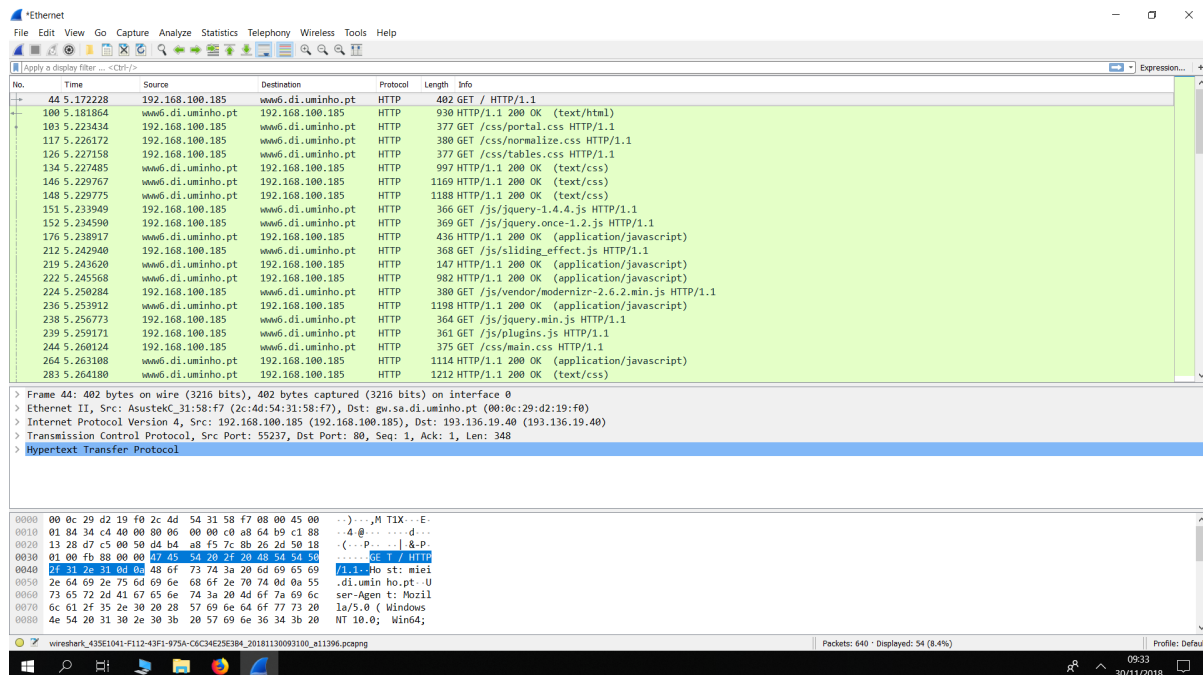


The screenshot shows the Wireshark interface with the 'arp' packet selected. The 'arp' packet details pane displays the ARP table. The table has four columns: 'Internet Address', 'Physical Address', and 'Type'. The 'Internet Address' column lists IP addresses, the 'Physical Address' column lists MAC addresses, and the 'Type' column indicates whether the entry is 'dynamic' or 'static'.

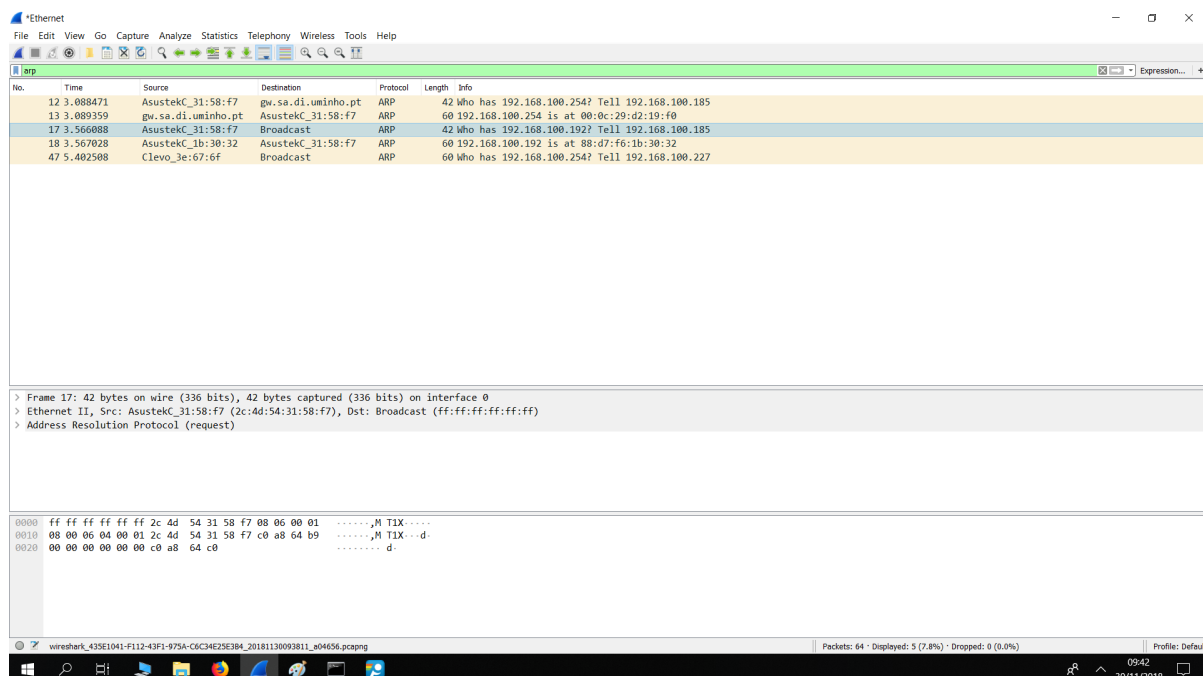
Internet Address	Physical Address	Type
192.168.100.185	0xb	
192.168.100.211	8c-16-45-7d-95-0d	dynamic
192.168.100.254	00-0c-29-d2-19-f0	dynamic
192.168.100.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-f0	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

A coluna *Internet Address* identifica todos os endereços que têm que ser acedidos. A segunda coluna contém o endereço MAC que tem que ser acedido de forma a chegar ao endereço IP. A terceira coluna corresponde ao tipo de endereçamento que está a ser usado. Esta tabela indica que endereços IP correspondem aos endereços MAC.

Na segunda aula para este trabalho prático, foi necessário obter novamente o número de ordem da sequência de bytes capturada, que se encontra em baixo.

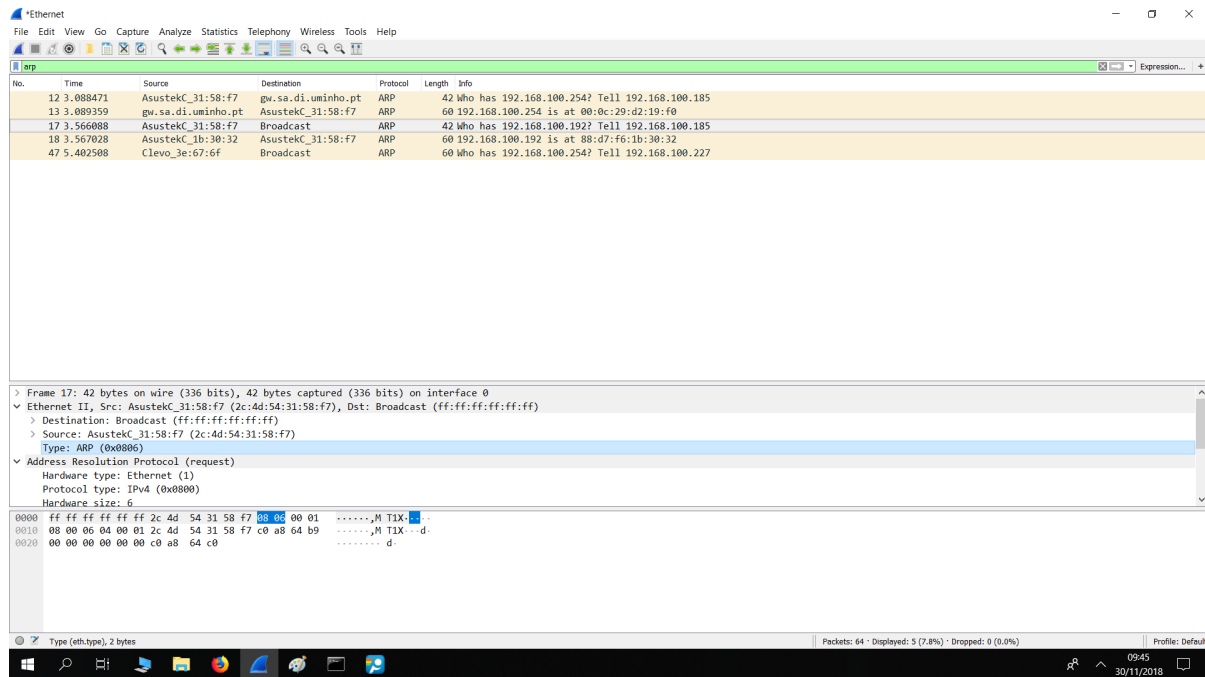


10) Qual é o valor hexadecimal dos endereços origem e destino na trama Ethernet que contém a mensagem com o pedido ARP(*ARP Request*)? Como interpreta e justifica o endereço destino usado?



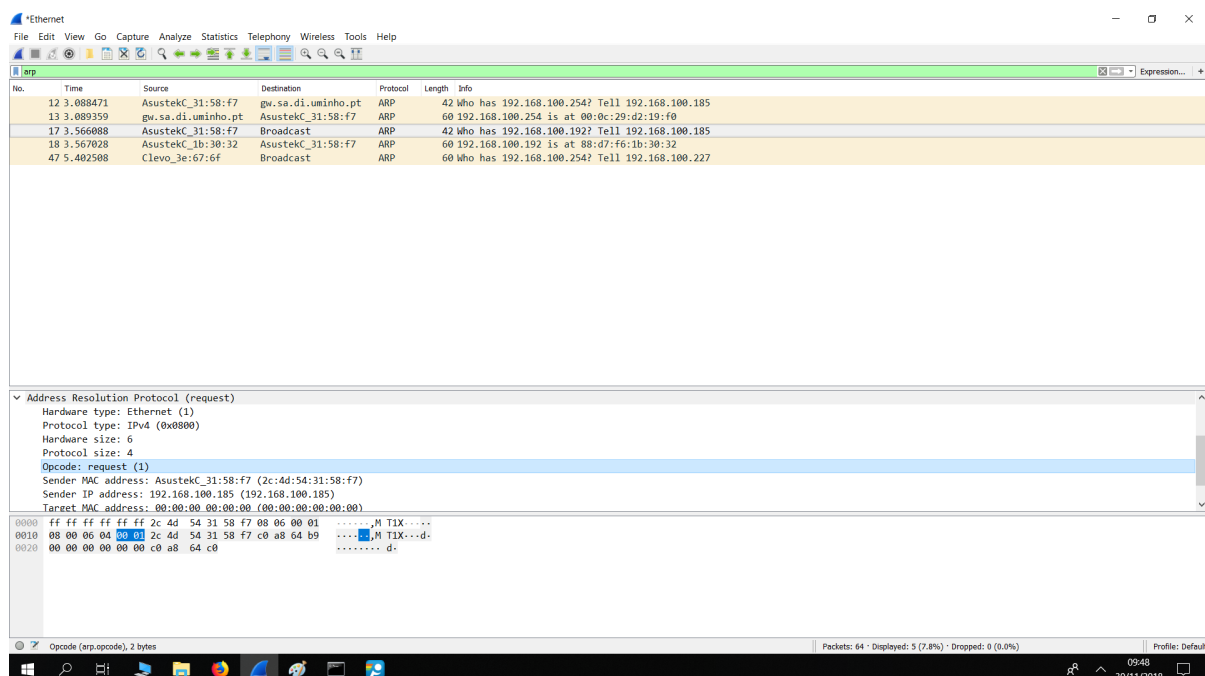
O valor hexadecimal do endereço de origem é 2c:4d:54:31:58:f7 e o de destino é ff:ff:ff:ff:ff:ff. O endereço destino tem que ser um endereço que possa ser captado por todas as máquinas. Depois de o endereço ff:ff:ff:ff:ff:ff ser capturado por todas as máquinas, só a que tiver o endereço pretendido é que irá responder com o endereço MAC.

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



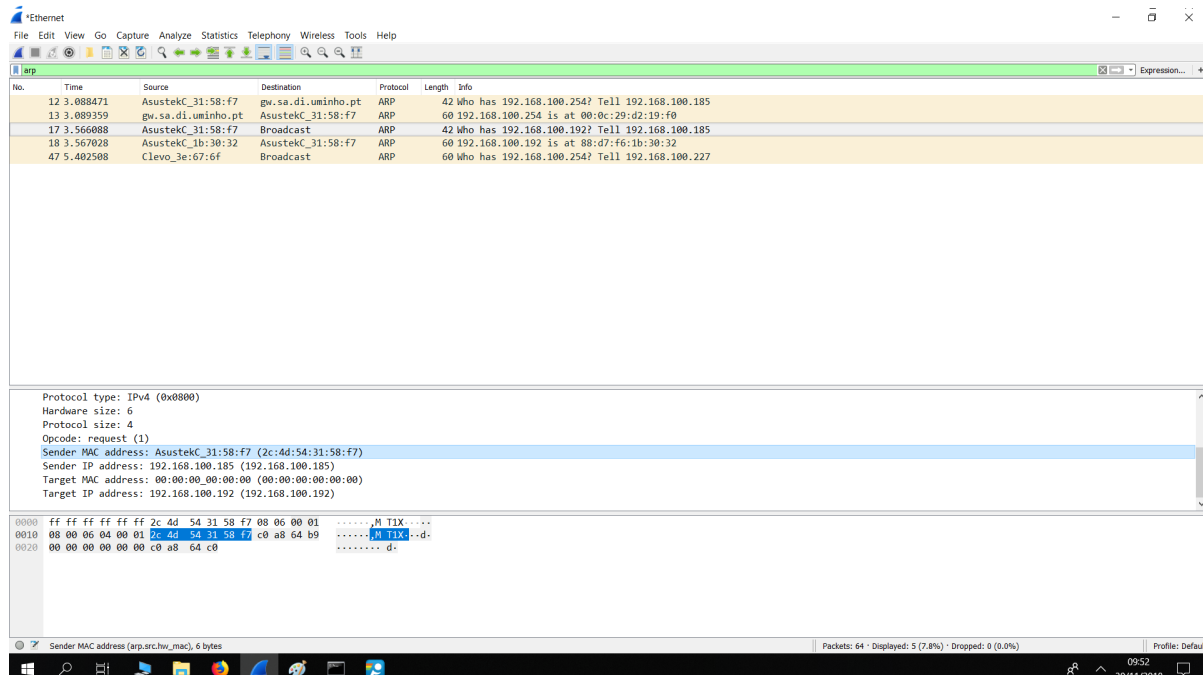
O valor hexadecimal do campo tipo é 0x0806 e o mesmo indica o tipo de dados da trama que nesta situação é ARP.

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>.



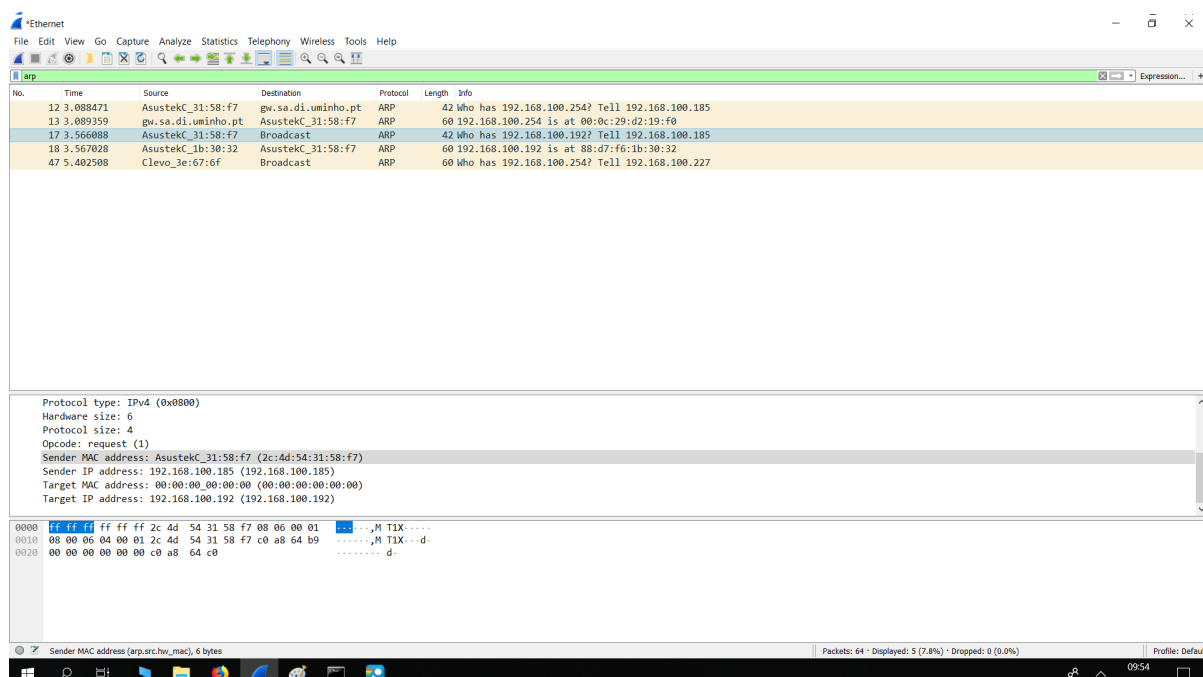
O valor do campo ARP *opcode* é 0x0001 da trama Ethernet e diz-nos se se trata de uma resposta a um pedido,reply, ou se é um pedido,request. Neste caso, refere-se a um pedido(request(1)).

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



Os endereços incluídos na mensagem ARP são endereços IP e endereços MAC. A mensagem contém o endereço IP de origem, 192.168.100.185, o endereço MAC de origem, ac:22:0b:ab:7d:dd. Também abrange o endereço IP do qual se quer conhecer o endereço MAC (target IP address:192.168.100.192). Como podemos observar o target endereço MAC é 00:00:00:00:00:00 pois ainda não sabemos de qual se trata.

14) Explícite que tipo de pedido ou pergunta é feita pelo *host* de origem?



O HOST de origem pretende saber o endereço MAC de quem tem aquele endereço IP. Deste modo, pergunta "Quem tem o endereço 192.168.100.192?" indicando que foi a

nossa máquina que fez a pergunta. Em resposta, é dado o endereço MAC correspondente àquele endereço IP.

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

Wireshark capture of ARP traffic. The packet list shows five ARP packets. Packet 18 is the response from AsustekC_1b:30:32 to AsustekC_31:58:f7. The packet details pane shows the Ethernet II, Src, Dst, and ARP fields. The packet bytes pane shows the raw data with MTIX markers.

No.	Time	Source	Destination	Protocol	Length	Info
12	3.088471	AsustekC_31:58:f7	gw.sa.di.uminho.pt	ARP	42	Who has 192.168.100.254? Tell 192.168.100.185
13	3.089359	gw.sa.di.uminho.pt	AsustekC_31:58:f7	ARP	60	192.168.100.254 is at 00:0c:29:d2:19:f0
17	3.566088	AsustekC_31:58:f7	Broadcast	ARP	42	Who has 192.168.100.192? Tell 192.168.100.185
18	3.567028	AsustekC_1b:30:32	AsustekC_31:58:f7	ARP	60	192.168.100.192 is at 88:d7:f6:1b:30:32
47	5.402508	Clevo_3e:67:6f	Broadcast	ARP	60	Who has 192.168.100.254? Tell 192.168.100.227

> Frame 18: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: AsustekC_1b:30:32 (88:d7:f6:1b:30:32), Dst: AsustekC_31:58:f7 (2c:4d:54:31:58:f7)
> Address Resolution Protocol (reply)

0000 2c 4d 54 31 58 f7 88 d7 f6 1b 30 32 08 06 00 01 ,MTIX... ..02....
0010 08 00 06 04 00 02 88 d7 f6 1b 30 32 c0 a8 64 c002..d..
0020 2c 4d 54 31 58 f7 c0 a8 64 b9 00 00 00 00 00 ,MTIX... d:.....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

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

Wireshark capture of ARP traffic. The packet list shows five ARP packets. Packet 18 is the response from AsustekC_1b:30:32 to AsustekC_31:58:f7. The packet details pane shows the Ethernet II, Src, Dst, and ARP fields. The packet bytes pane shows the raw data with MTIX markers.

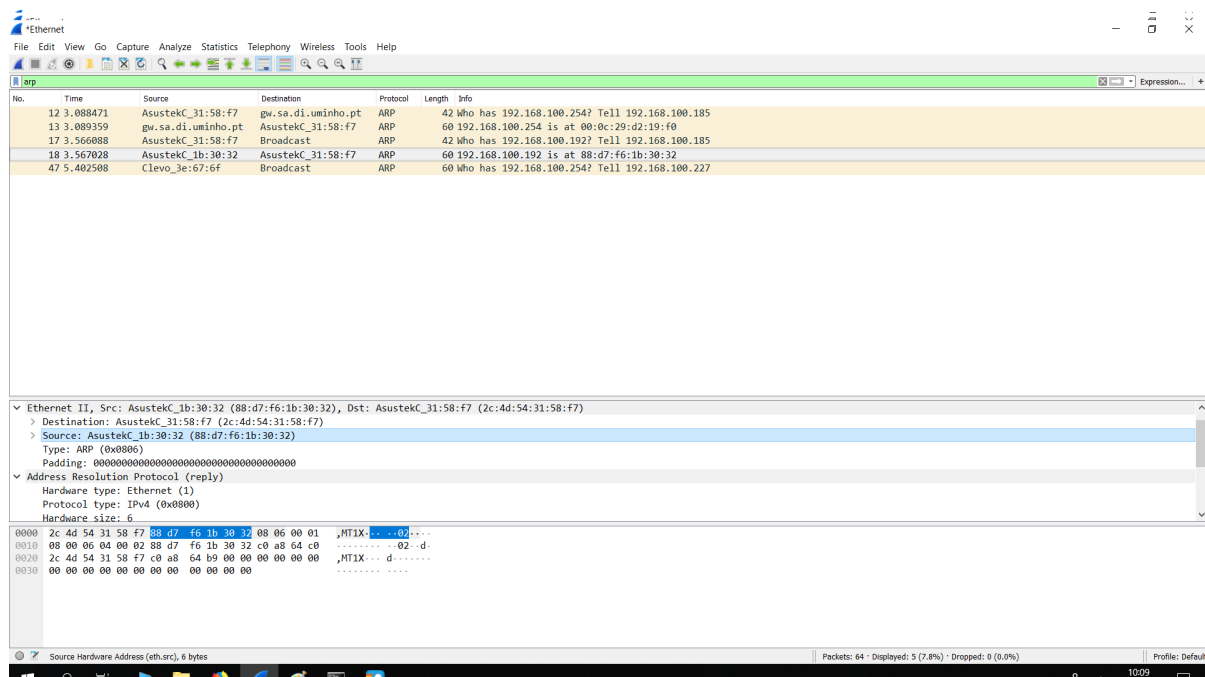
No.	Time	Source	Destination	Protocol	Length	Info
12	3.088471	AsustekC_31:58:f7	gw.sa.di.uminho.pt	ARP	42	Who has 192.168.100.254? Tell 192.168.100.185
13	3.089359	gw.sa.di.uminho.pt	AsustekC_31:58:f7	ARP	60	192.168.100.254 is at 00:0c:29:d2:19:f0
17	3.566088	AsustekC_31:58:f7	Broadcast	ARP	42	Who has 192.168.100.192? Tell 192.168.100.185
18	3.567028	AsustekC_1b:30:32	AsustekC_31:58:f7	ARP	60	192.168.100.192 is at 88:d7:f6:1b:30:32
47	5.402508	Clevo_3e:67:6f	Broadcast	ARP	60	Who has 192.168.100.254? Tell 192.168.100.227

Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: AsustekC_1b:30:32 (88:d7:f6:1b:30:32)
Sender IP address: 192.168.100.192 (192.168.100.192)
Target MAC address: AsustekC_31:58:f7 (2c:4d:54:31:58:f7)
Target IP address: 192.168.100.185 (192.168.100.185)

0000 2c 4d 54 31 58 f7 88 d7 f6 1b 30 32 08 06 00 01 ,MTIX... ..02....
0010 08 00 06 04 00 02 88 d7 f6 1b 30 32 c0 a8 64 c002..d..
0020 2c 4d 54 31 58 f7 c0 a8 64 b9 00 00 00 00 00 ,MTIX... d:.....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

O campo ARP *opcode* tem o valor 0x0002. Trata-se da resposta ao pedido ARP realizado.

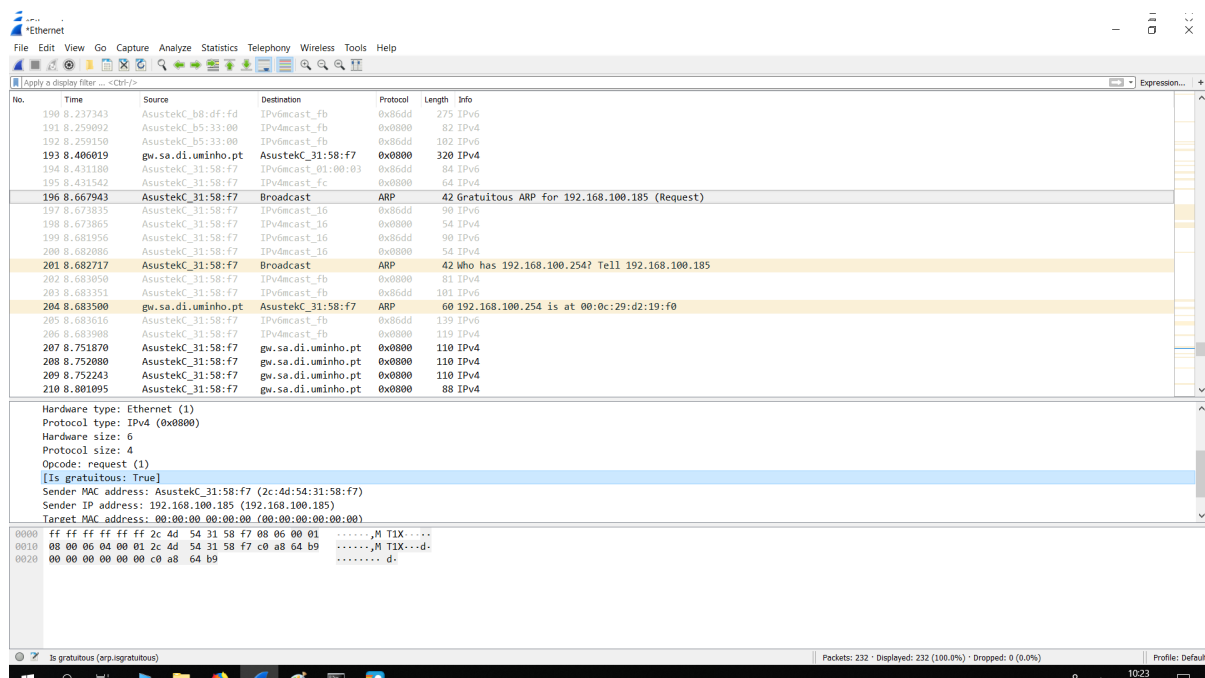
b) Em que posição da mensagem ARP está a resposta ao pedido ARP?

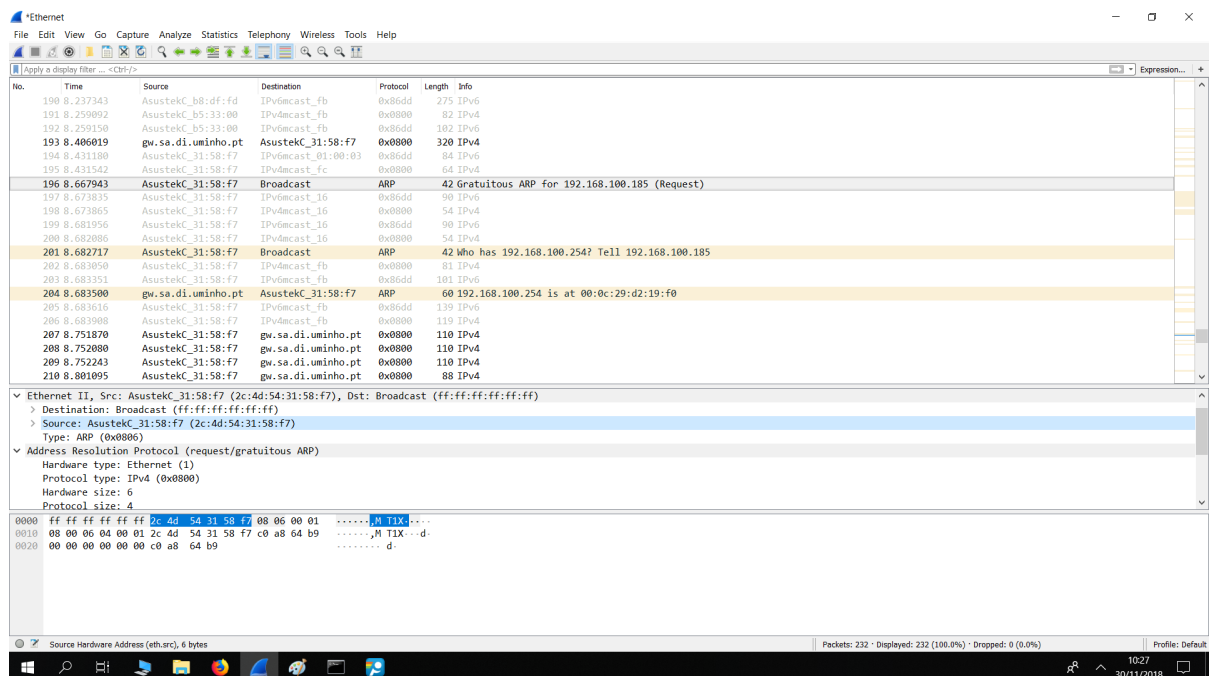


A resposta ao pedido ARP é o endereço MAC que está contido dos bytes 6 a 11.

5. ARP numa topologia CORE

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



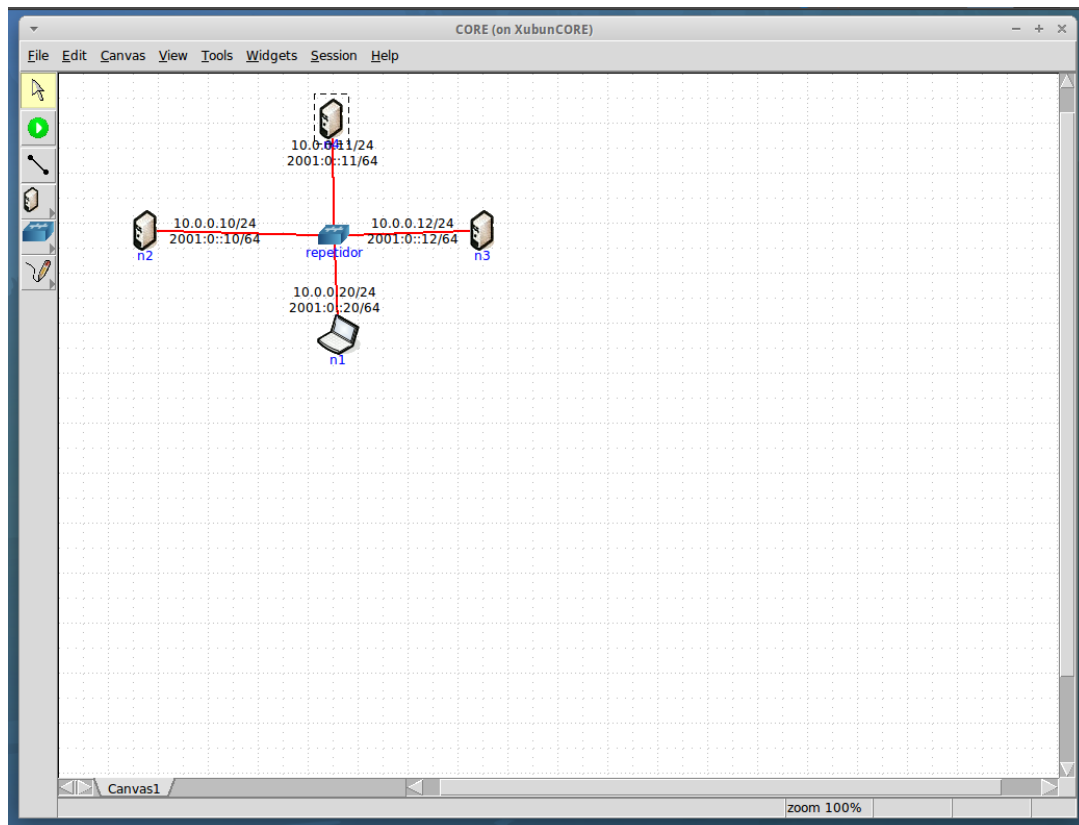


A principal diferença entre os dois é que o pedido ARP gratuito tem o endereço IP de origem igual ao endereço IP de destino. Isto permite-nos saber se existe alguma máquina da rede com o mesmo endereço IP que a nossa. Se houvesse, iríamos receber uma mensagem de resposta, pois a mensagem de pedido é enviada para todos os equipamentos e respondida pelo equipamento com o endereço de destino. Como não havia nenhuma máquina com endereço IP igual, logo não houve resposta ao pedido ARP gratuito.

Também é possível tornar o envio e receção das mensagens na rede mais eficiente, pois o pedido ARP gratuito permite informar outros equipamentos do endereço MAC o nosso computador.

6. Domínios de colisão

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



```
vcmd
th 64
10:45:10.088638 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 67, length 64
10:45:11.066660 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 68, length 64
10:45:11.066840 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 68, length 64
10:45:12.074129 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 69, length 64
10:45:12.074397 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 69, length 64
10:45:13.073106 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 70, length 64
10:45:13.073392 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 70, length 64
10:45:14.111662 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 71, length 64
10:45:14.111969 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 71, length 64
10:45:15.122502 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 72, length 64
10:45:15.123416 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 72, length 64
[]

root@n1:/tmp/pscore.49766/n1.conf
64 bytes from 10.0.0.10: icmp_req=54 ttl=64 time=4.10 ms
64 bytes from 10.0.0.10: icmp_req=55 ttl=64 time=7.24 ms
64 bytes from 10.0.0.10: icmp_req=56 ttl=64 time=0.460 ms
64 bytes from 10.0.0.10: icmp_req=57 ttl=64 time=8.04 ms
64 bytes from 10.0.0.10: icmp_req=58 ttl=64 time=0.457 ms
64 bytes from 10.0.0.10: icmp_req=59 ttl=64 time=7.82 ms
64 bytes from 10.0.0.10: icmp_req=60 ttl=64 time=0.480 ms
64 bytes from 10.0.0.10: icmp_req=61 ttl=64 time=1.02 ms
64 bytes from 10.0.0.10: icmp_req=62 ttl=64 time=1.75 ms
64 bytes from 10.0.0.10: icmp_req=63 ttl=64 time=11.4 ms
64 bytes from 10.0.0.10: icmp_req=64 ttl=64 time=6.35 ms
64 bytes from 10.0.0.10: icmp_req=65 ttl=64 time=1.78 ms
64 bytes from 10.0.0.10: icmp_req=66 ttl=64 time=4.78 ms
64 bytes from 10.0.0.10: icmp_req=67 ttl=64 time=27.8 ms
64 bytes from 10.0.0.10: icmp_req=68 ttl=64 time=2.94 ms
64 bytes from 10.0.0.10: icmp_req=69 ttl=64 time=4.42 ms
64 bytes from 10.0.0.10: icmp_req=70 ttl=64 time=4.49 ms
64 bytes from 10.0.0.10: icmp_req=71 ttl=64 time=4.77 ms
64 bytes from 10.0.0.10: icmp_req=72 ttl=64 time=2.40 ms
-- 10.0.0.10 ping statistics --
72 packets transmitted, 72 received, 0% packet loss, time 71594ms
rtt min/avg/max/ndev = 0.392/4.173/27.063/5.058 ms
root@n1:/tmp/pscore.49766/n1.conf# []

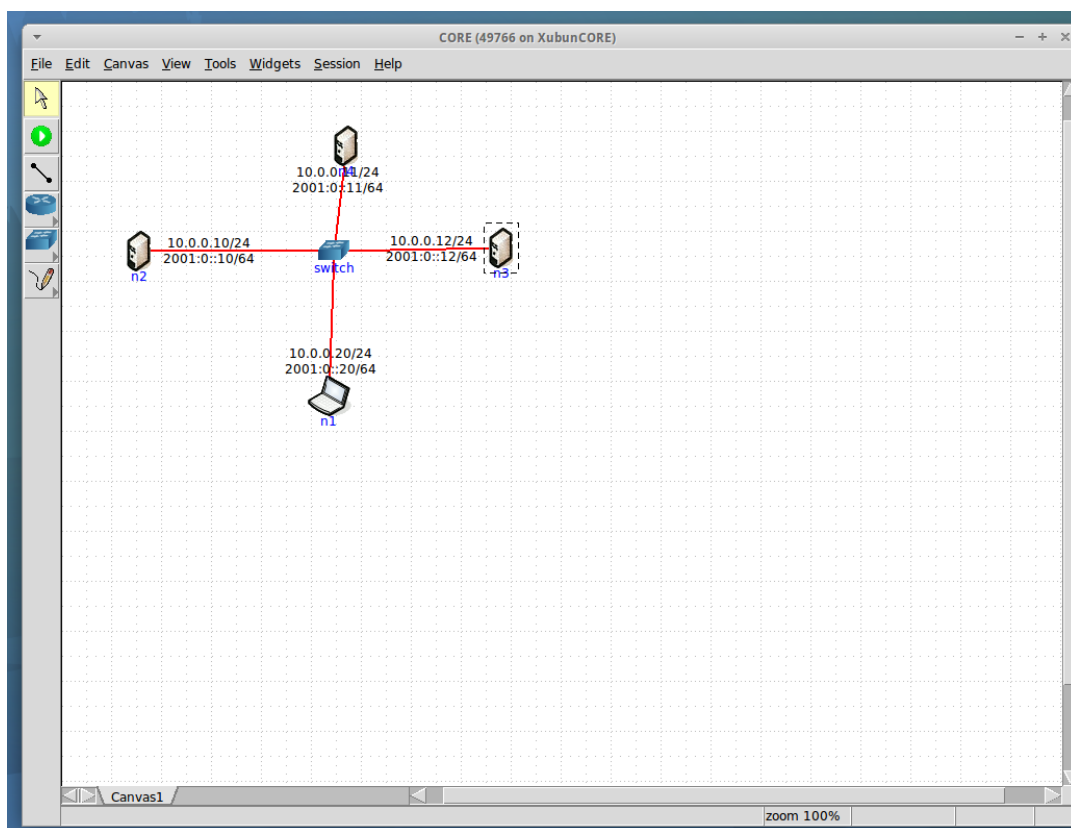
vcmd
th 64
10:45:10.091088 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 67, length 64
10:45:11.066360 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 68, length 64
10:45:11.067285 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 68, length 64
10:45:12.070878 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 69, length 64
10:45:12.074780 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 69, length 64
10:45:13.075916 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 70, length 64
10:45:13.080119 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 70, length 64
10:45:14.111182 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 71, length 64
10:45:14.115669 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 71, length 64
10:45:15.122022 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 52, seq 72, length 64
10:45:15.124138 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 52, seq 72, length 64
[]
```

Observando os prints em baixo conseguimos ver que os pacotes que chegam a n2, n3 e n4 são os mesmos.

Depois de executarmos o comando ping do computador n1 para o host n2 com a

opção *tcpdump* ativa, podemos observar que o tráfego gerado chega de igual forma a todos os dispositivos. Antes de chegar a qualquer host, a informação emitida por n1 com destino n2, é também compartilhada pelos restantes equipamentos (hosts n3 e n4) ligados ao *hub*(repetidor). Tal acontece, porque o *hub* tem como função retransmitir o sinal e enviá-lo para todos os equipamentos ligados a ele.

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.



The image displays four terminal windows arranged in a 2x2 grid, showing network traffic and ping statistics. The top-left window, titled 'vcmd', shows a series of ICMP echo requests and replies between IP addresses 10.0.0.10 and 10.0.0.20. The top-right window, also titled 'vcmd', shows a message about tcpdump output being suppressed. The bottom-left window, titled 'root@n1:/tmp/pycore.49766/n1.conf', shows a list of ICMP requests from 10.0.0.10 to 10.0.0.10, along with ping statistics for 10.0.0.10. The bottom-right window, titled 'vcmd', shows a message about tcpdump output being suppressed.

```
th 64
10:48:52.703464 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 48, seq 42, length
64
10:48:53.703729 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 48, seq 43, leng
th 64
10:48:53.703798 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 48, seq 43, length
64
10:48:54.711400 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 48, seq 44, leng
th 64
10:48:54.711658 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 48, seq 44, length
64
10:48:55.718085 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 48, seq 45, leng
th 64
10:48:55.718223 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 48, seq 45, length
64
10:48:56.719873 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 48, seq 46, leng
th 64
10:48:56.723862 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 48, seq 46, length
64
10:48:57.721029 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 48, seq 47, leng
th 64
10:48:57.721087 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 48, seq 47, length
64
[]

64 bytes from 10.0.0.10: icmp_req=29 ttl=64 time=13.9 ms
64 bytes from 10.0.0.10: icmp_req=30 ttl=64 time=3.92 ms
64 bytes from 10.0.0.10: icmp_req=31 ttl=64 time=0.334 ms
64 bytes from 10.0.0.10: icmp_req=32 ttl=64 time=0.341 ms
64 bytes from 10.0.0.10: icmp_req=33 ttl=64 time=16.2 ms
64 bytes from 10.0.0.10: icmp_req=34 ttl=64 time=0.274 ms
64 bytes from 10.0.0.10: icmp_req=35 ttl=64 time=0.514 ms
64 bytes from 10.0.0.10: icmp_req=36 ttl=64 time=0.728 ms
64 bytes from 10.0.0.10: icmp_req=37 ttl=64 time=3.52 ms
64 bytes from 10.0.0.10: icmp_req=38 ttl=64 time=4.59 ms
64 bytes from 10.0.0.10: icmp_req=39 ttl=64 time=0.772 ms
64 bytes from 10.0.0.10: icmp_req=40 ttl=64 time=7.28 ms
64 bytes from 10.0.0.10: icmp_req=41 ttl=64 time=1.27 ms
64 bytes from 10.0.0.10: icmp_req=42 ttl=64 time=1.59 ms
64 bytes from 10.0.0.10: icmp_req=43 ttl=64 time=0.391 ms
64 bytes from 10.0.0.10: icmp_req=44 ttl=64 time=1.27 ms
64 bytes from 10.0.0.10: icmp_req=45 ttl=64 time=2.93 ms
64 bytes from 10.0.0.10: icmp_req=46 ttl=64 time=5.00 ms
64 bytes from 10.0.0.10: icmp_req=47 ttl=64 time=0.326 ms
^C
--- 10.0.0.10 ping statistics ---
47 packets transmitted, 47 received, 0% packet loss, time 46208ms
rtt min/avg/max/mdev = 0.325/2.629/16.247/3.209 ms
root@n1:/tmp/pycore.49766/n1.conf# []

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
[]
```

Através da observação dos prints em cima, conseguimos perceber que após ser feito o comando ping, o host n3 e n4 apenas receberam uma mensagem ARP e o n2 recebeu os pacotes todos que lhe eram destinados. O aparelho que faz a distribuição do sinal, *switch*, emite um pedido ARP para todos os equipamentos e após receber a resposta a esta primeira mensagem ARP, proveniente de n2, identifica o endereço MAC do host destino. Deste modo, o host n3 e n4 apenas recebem a mensagem ARP, enquanto que o host n2 recebe a informação que lhe é destinada. Isto ajuda na diminuição da quantidade de informação desnecessária que circula na rede uma vez que a mesma não vai para equipamentos que não desejamos, tornando a opção de partilha da informação muito mais eficiente. Esta característica do switch de o registo dos endereços MAC associados a cada porta de entrada permite evitar muitas colisões devido à existência de mais domínios de colisão. Faz com que seja possível que n3 possa transmitir tramas enquanto n1 transmite para n2. Portanto, pode concluir-se que, neste aspeto, um switch é mais funcional que um hub

Conclusão

Com a realização deste trabalho adquirimos conhecimento que nos permite como analisar tramas Ethernet. Estas mesmas são organizadas em bytes que ,por sua vez, sozinhos ou em conjunto com outros, permitem-nos retirar informação pertinente acerca do modo como os pacotes fluem na rede pela qual circulam, quais os endereços IP e MAC de origem e destino, protocolos usados, e que parte da trama é dedicada à informação que se quer fazer chegar ao destino, que se podem incluir campos para deteção de erros, entre outros.

Ficamos familiarizados com mensagens ARP, com o seu conteúdo e com a forma como estas funcionam, como é feito o seu mapeamento de endereços MAC(envio de pedidos precedidos das respetivas respostas) e a associação de cada uma delas à respetiva porta de entrada de um equipamento de distribuição de sinais.

Ainda relativo ao parágrafo anterior, foi-nos possível entender como funcionam e para que servem os ARP gratuitos. Estes permitem-nos detetar vários problemas na nossa rede, nomeadamente a duplicação de endereços IP, e podemos fazê-lo com o envio de uma simples mensagem para a rede.

Por fim, ficamos a perceber a diferença entre um hub e um switch, nomeadamente na maneira como estes retransmitem o sinal que lhes chega. O switch é capaz de seleccionar os locais para onde deve enviar a informação enquanto o hub envia para todas as suas outras portas que estiverem conectadas. Deste modo, concluímos que o switch, nesta situação, é bem mais vantajoso que o hub.

Em virtude do que foi mencionado, concluímos que foi um trabalho muito enriquecedor, que nos permitiu aprofundar vários conceitos e ademais entender o funcionamento de alguns equipamentos ao qual lidamos no nosso quotidiano.