

Camada de Ligação Lógica: Ethernet e Protocolo ARP

André Gonçalves (A80368), Diogo Gonçalves (A81860), Maria Pires(A86268)

September 9, 2019

Questões e Respostas

1 Captura e análise de Tramas Ethernet

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

- Origem: 00:0c:29:d2:19:f0
- Destino: 4c:cc:6a:e1:cb:06

```
Ethernet II, Src: Micro-St_e1:cb:06 (4c:cc:6a:e1:cb:06), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
  Destination: 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)
  Source: Micro-St_e1:cb:06 (4c:cc:6a:e1:cb:06)
    Address: Micro-St_e1:cb:06 (4c:cc:6a:e1:cb:06)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
```

Figure 1: Campo Ethernet que contém os endereços MAC

1.2 Identifique a que sistemas se referem. Justifique.

A origem refere-se ao nosso computador e o destino ao router. Estamos a conectar o nosso computador à rede Ethernet local.

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

O valor hexadecimal do campo *Type* da trama Ethernet é 0x0800 e indica-nos que o tipo de encapsulamento presente é IPv4.



Figure 2: Type da trama Ethernet

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

Desde o início da trama até ao caractere ASCII “G” do método HTTP GET são usados 66 bytes. No total são usados 441 bytes, logo a percentagem de sobrecarga (overhead) introduzida pela pilha protocolar no envio do HTTP GET é de $(66/441) \times 100 = 14,97\%$

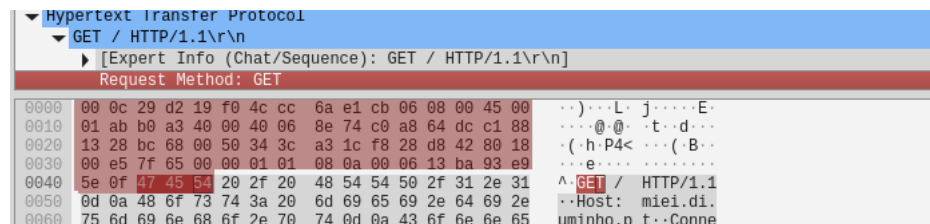


Figure 3: Trama do método HTTP GET

1.5 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?

O campo FCS (Frame check sequence) não aparece o que nos indica que não está a ser usado. Isto faz todo o sentido pois como estamos perante uma ligação Ethernet não é preciso usar o campo FCS uma vez que neste tipo de ligação uma trama danificada deve ser descartada.

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

O endereço da fonte é 4c:cc:6a:e1:cb:06 e corresponde ao router da rede local como se pode verificar pela Figura 4.

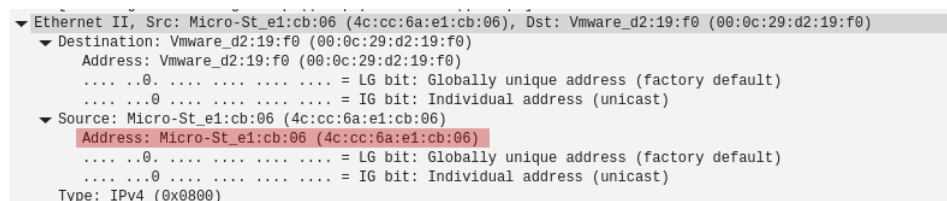


Figure 4: Endereço Ethernet da fonte

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

O endereço MAC do destino é 00:0c:29:d2:19:f0, que corresponde à interface de comunicação do nosso computador.

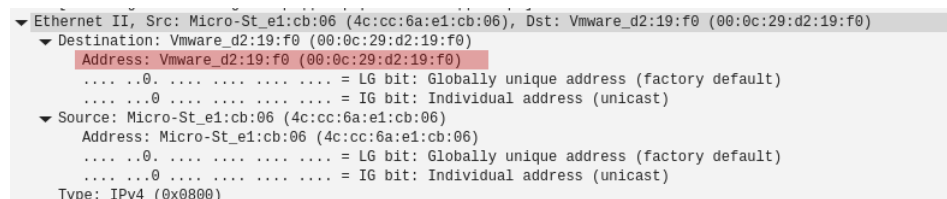


Figure 5: Endereço MAC do destino

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

A trama recebida contém 3 protocolos: TCP (nível de transporte), IPv4 (nível de rede) e HTTP (nível aplicacional).

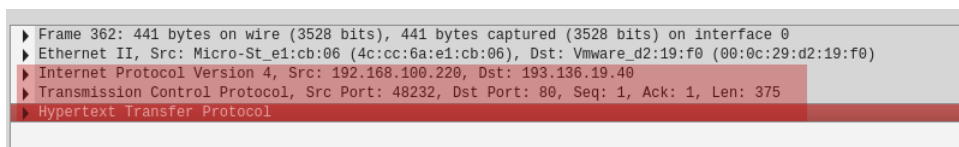


Figure 6: Protocolos contidos na trama

2 Protocolo ARP

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

A tabela ARP constituida por 5 colunas:

Coluna	Significado
Address	Representa o endereço IP do destino
HWtype	Representa o meio de ligação até ao destino(e.g. ether é ethernet)
HWaddress	Representa o MAC address do destino
Flags Mask	Representa o tipo de entrada(e.g. C é <i>complete entry</i>)
Iface	Representa o tipo de interface

```
[~] arp master
Address      HWtype  HWaddress    Flags Mask    Iface
gw.sa.di.uminho.pt ether    00:0c:29:d2:19:f0 C             enp3s
192.168.100.213 ether    08:60:6e:04:71:b6 C             enp3s
```

Figure 7: Tabeça ARP

2.2 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 destino é 4c:cc:6a:e1:cb:06 e da origem é 08:60:6e:04:71:b6 que representam os MAC addresses. O valor obtido do destino é o MAC address da máquina para onde estamos a tentar usar o ping.

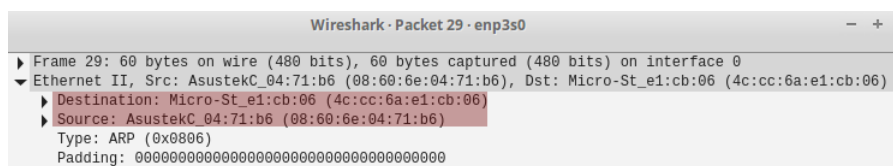


Figure 8: Campo Ethernet que contém os endereços MAC

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

ARP(0x0806) é o valor hexadecimal do campo tipo da trama Ethernet e indica-nos o protocolo usado.

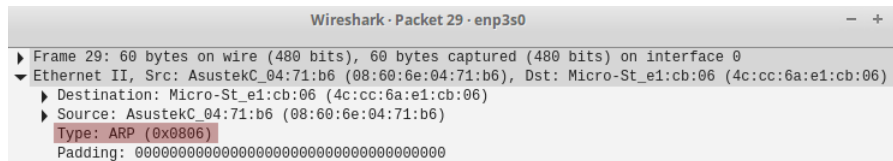


Figure 9: Type da trama Ethernet

2.4 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>.

RFC é um método de conversão de endereços IP em endereços Ethernet. O valor do campo *opcode* é *Request (1)* que especifica o objetivo da trama, sendo neste caso o pedido do endereço MAC destino da trama em questão.

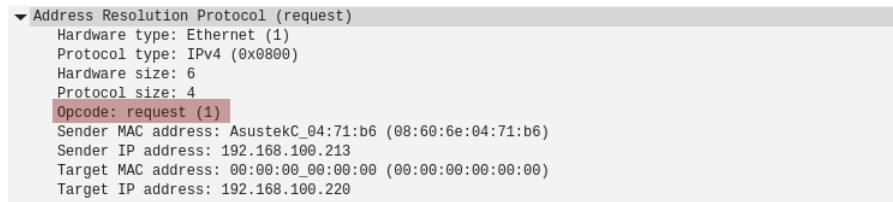


Figure 10: Campo ARP *opcode*

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

Os endereços contidos na mensagem ARP são o IP e MAC do emissor e do receptor de modo a que seja possível a troca de tramas entre eles.

2.6 Explícite que tipo de pedido ou pergunta é feita pelo host de origem?

O host de origem pede o endereço MAC para o qual pretende enviar a trama.

27	8.986655419	Micro-St_e1:cb:06	AsustekC_04:71:b6	0x0800	98	IPv4
28	8.986994027	AsustekC_04:71:b6	Micro-St_e1:cb:06	0x0800	98	IPv4
29	9.951993546	AsustekC_04:71:b6	Micro-St_e1:cb:06	ARP	60	Who has 192.168.100.220? Tell 192.168.100.213
30	9.952015940	Micro-St_e1:cb:06	AsustekC_04:71:b6	ARP	42	192.168.100.220 is at 4c:cc:6a:e1:cb:06
31	9.987726196	Cisco_5b:13:51	Spanning-tree-(for-...	STP	60	Conf. Root = 4096/720/00:0a:8a:97:74:80 Cost =

Figure 11: Pergunta/Pedido do host de origem

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

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

Reply (2) é o campo do ARP *opcode* que retorna o endereço MAC.

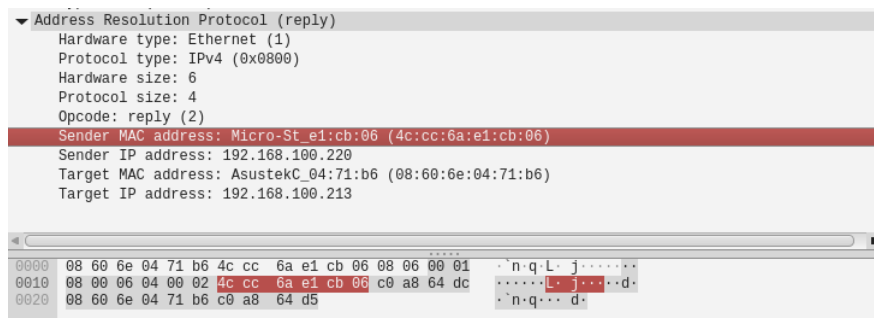


Figure 12: Campo ARP *opcode*

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

A resposta ao pedido ARP encontra-se no intervalo 22-27 bytes.

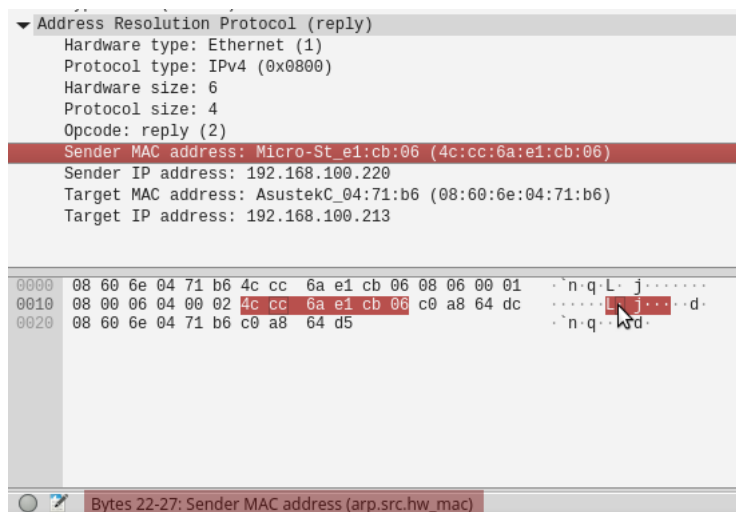


Figure 13: Resposta ao pedido ARP

3 ARP Gratuito

3.1 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?

O ARP gratuito distingue-se dos restantes pedidos na medida em que o endereço IP de destino e de origem são o IP da máquina emitindo o pacote. Para além disso, o endereço MAC de destino é o endereço de broadcast ff:ff:ff:ff:ff:ff.

O resultado esperado é que todos os os sistemas na rede atualizem as suas tabelas ARP.

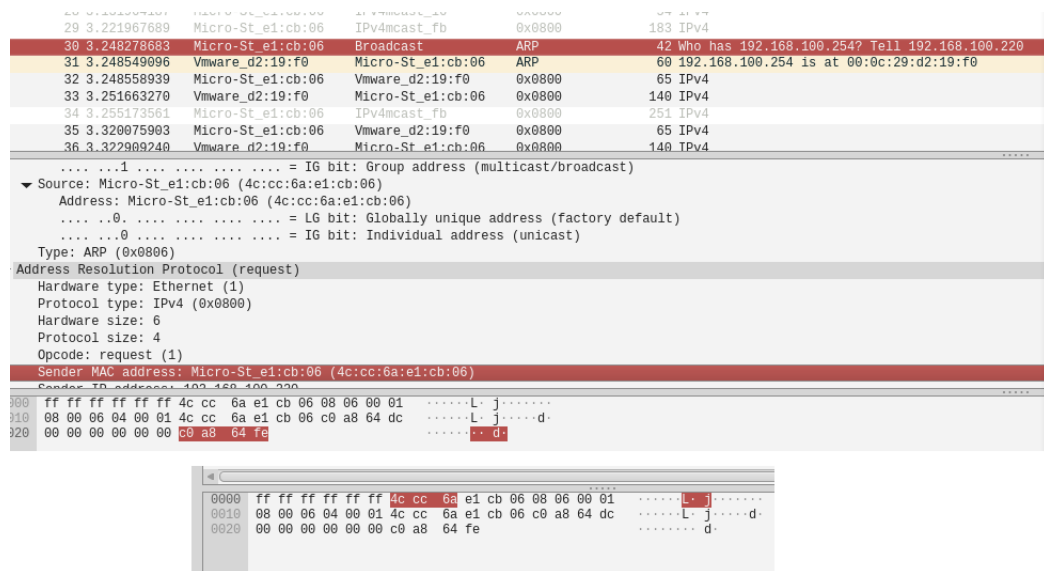
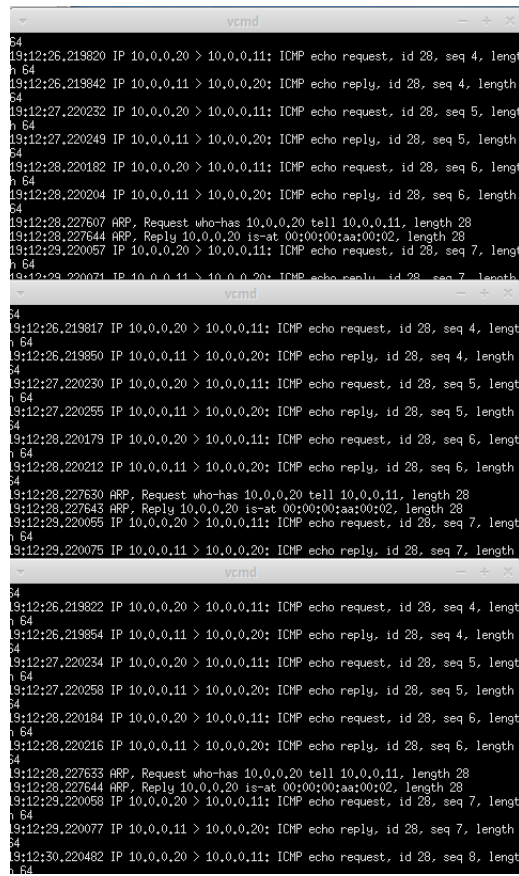


Figure 14: Broadcast

4 Domínios de Colisão

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

Depois de realizar o comando ping do laptop n1 para o servidor n2, o laptop envia um ARP request para a rede de modo a descobrir o MACaddress do destino, depois disso começa a enviar para o n2. Como no centro desta rede se encontra um hub quando são enviados os pacotes estes são repetidos para os outros 2 servidores na rede(n3,n4).



```
vcmd
19:12:26.219820 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 4, length 64
19:12:26.219842 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 4, length 64
19:12:27.220232 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 5, length 64
19:12:27.220249 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 5, length 64
19:12:28.220182 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 6, length 64
19:12:28.220204 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 6, length 64
19:12:28.227607 ARP, Request who-has 10.0.0.20 tell 10.0.0.11, length 28
19:12:28.227644 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:02, length 28
19:12:29.220057 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 7, length 64
19:12:29.220074 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 7, length 64

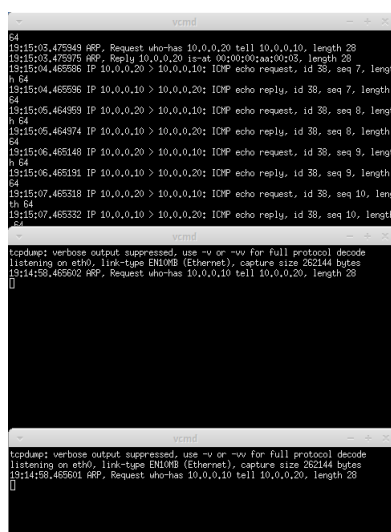
vcmd
19:12:26.219817 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 4, length 64
19:12:26.219850 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 4, length 64
19:12:27.220230 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 5, length 64
19:12:27.220255 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 5, length 64
19:12:28.220179 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 6, length 64
19:12:28.220212 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 6, length 64
19:12:28.227630 ARP, Request who-has 10.0.0.20 tell 10.0.0.11, length 28
19:12:28.227643 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:02, length 28
19:12:29.220055 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 7, length 64
19:12:29.220075 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 7, length 64

vcmd
19:12:26.219822 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 4, length 64
19:12:26.219854 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 4, length 64
19:12:27.220234 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 5, length 64
19:12:27.220258 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 5, length 64
19:12:28.220184 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 6, length 64
19:12:28.220216 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 6, length 64
19:12:28.227633 ARP, Request who-has 10.0.0.20 tell 10.0.0.11, length 28
19:12:28.227644 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:02, length 28
19:12:29.220058 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 7, length 64
19:12:29.220077 IP 10.0.0.11 > 10.0.0.20: ICMP echo reply, id 28, seq 7, length 64
19:12:30.220482 IP 10.0.0.20 > 10.0.0.11: ICMP echo request, id 28, seq 8, length 64
```

Figure 15: tcp-dump dos servidores com um hub

4.2 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 *switchs* no contexto de controlar ou dividir domínios de colisão. Documente as suas observações e conclusões com base no tráfego observado/capturado.

Com o uso de um switch em vez de um hub o que acontece é que em vez do sinal ser repetido para toda a rede, com o switch apenas é enviado para o destino pretendido como é possível verificar na Figura 16. Em relação aos hubs, estes são dispositivos de nível 1 que não usam lógica de endereçamento e por isso fazem broadcast a todas as tramas que recebem. Todas as ligações ao hub formam apenas um domínio de colisão como é possível verificar na figura 15. Em relação aos switch como este já se encontra no nível 2 utiliza o endereçamento, logo quando é enviada uma mensagem para um switch este enviará apenas para o destino pretendido. Neste caso ira existir tantos domínios de colisão quantas ligações existirem ao switch, no caso estudado existe um total de 4 domínios de colisão.



```
tcpdump -i eth0 -s 262144 -n -v -w -
13:15:03.475949 ARP, Request who-has 10.0.0.20 tell 10.0.0.10, length 28
13:15:03.475975 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:03, length 28
13:15:04.465586 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 38, seq 7, length 64
13:15:04.465596 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 38, seq 7, length 64
13:15:05.464959 IP 10.0.0.10 > 10.0.0.20: ICMP echo request, id 38, seq 8, length 64
13:15:05.464974 IP 10.0.0.20 > 10.0.0.10: ICMP echo reply, id 38, seq 8, length 64
13:15:06.465131 IP 10.0.0.10 > 10.0.0.20: ICMP echo request, id 38, seq 9, length 64
13:15:06.465148 IP 10.0.0.20 > 10.0.0.10: ICMP echo reply, id 38, seq 9, length 64
13:15:07.465318 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 38, seq 10, length 64
13:15:07.465332 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 38, seq 10, length 64

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type ENLMB (Ethernet), capture size 262144 bytes
13:14:58.455610 ARP, Request who-has 10.0.0.10 tell 10.0.0.20, length 28
```

Figure 16: tcp-dump dos servidores com um switch

5 Conclusões

Os datagramas são encapsulados numa trama que contém um cabeçalho onde é incluído o endereço MAC (endereço físico) de modo a identificar o emissor e o recetor da informação. Sendo o objetivo deste trabalho estudar, de uma forma genérica, a camada de ligação lógica focando na tecnologia Ethernet, através do programa *Wireshark* executamos a captura e análise de tráfego entre o nosso computador e a rede local do DI através da qual observamos a transferência de tráfego prestada pela camada de ligação lógica. Esta camada é a mais eficiente da pilha protocolar para a deteção e correção de erros, contudo estando a usar a rede Ethernet na eventual deteção de um erro a trama seria descartada. O protocolo ARP (Address Resolution Protocol) permite mapear entre endereços ao nível de rede (IP) e ao nível de ligação lógica (endereços MAC), este protocolo possui uma cache de endereços para os quais foi enviada informação recentemente, na eventualidade de querer enviar tráfego para um destino que não seja conhecido é feito um broadcast de modo a conhecer o endereço MAC desse destino e poder enviar a trama que será posteriormente desencapsulada no nível superior que observamos ao fazer ping para um host da sala de aula.

O endereço MAC de uma trama tem que ser examinado de modo a conseguir encaminhá-la até ao destino correto. Numa rede local como a do DI, que é partilhada por vários hosts, existe o risco de o envio de uma trama coincidir temporalmente com um trama de um outro host havendo colisão e, mais uma vez, estando perante uma rede Ethernet as tramas danificadas seriam descartadas. Através da simulação executada no CORE observamos o tratamento do fluxo de tráfego para evitar colisões através do uso de hubs e *switches*.