

# eTP3: Camada de Ligação Lógica: Ethernet e Protocolo ARP

Filipa Correia Parente, José André Martins Pereira, Ricardo André Gomes Petronilho

University of Minho, Department of Informatics, 4710-057 Braga, Portugal e-mail:  
[{a82145,a82880,a81744}@alunos.uminho.pt](mailto:{a82145,a82880,a81744}@alunos.uminho.pt)

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

```
Ethernet II, Src: HewlettP_fc:6b:36 (a0:8c:fd:fc:6b:36), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
```

Figura 1 - Endereços MAC origem e destino.

**Resposta:**

O endereço MAC de origem é o **a0:8c:fd:fc:6b:36** como se pode observar na região a vermelho e o endereço MAC de destino é o **00:0c:29:d2:19:f0** como se pode observar na região a azul.

2) Identifique a que sistemas se referem. Justifique.

**Resposta:**

Os MAC address referidos acima referem-se à placa de rede (NIC), do nosso computador (**HewlettP**) e a do servidor (**Vmware**).

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

```
Ethernet II, Src: HewlettP_fc:6b:36 (a0:8c:fd:fc:6b:36), 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: HewlettP_fc:6b:36 (a0:8c:fd:fc:6b:36)
Address: HewlettP_fc:6b:36 (a0:8c:fd:fc:6b:36)
....0. .... = LG bit: Globally unique address (factory default)
....0. .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
```

Figura 2 - Campo Type.

**Resposta:**

Identifica o tipo de encapsulamento usado para transportar os dados. Neste caso tem o valor **0x0800** o que significa que é um pacote do tipo **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.

0000	00 0c 29 d2 19 f0 a0 8c fd fc 6b 36 08 00 45 00	..). .... k6..E.
0010	01 f5 74 7d 40 00 40 06 ca 53 c0 a8 64 d9 c1 88	..t}@.@ .S..d..
0020	13 28 9f b6 00 50 0f 68 c1 11 c8 f4 4c 32 80 18	..( ...P.h ...L2..
0030	00 e5 5b 1b 00 00 01 01 08 0a 05 1e 66 d2 94 13	..[ ... ..f...
0040	82 c3 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31	GET / HTTP/1.1
0050	0d 0a 48 61 73 74 3a 20 6d 69 65 69 2e 64 69 2e	..Host: miei.di.
0060	75 6d 69 6e 68 6f 2e 70 74 0d 0a 43 6f 6e 6e 65	uminho.p t..Conne
0070	63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76	ction: k eep-aliv
0080	65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65 63	e..Upgra de-Insec
0090	75 72 65 2d 52 65 71 75 65 73 74 73 3a 20 31 0d	ure-Requ ests: 1.
00a0	0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a	..User-Ag ent: Moz
00b0	69 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 4c	illa/5.0 (X11; L
00c0	69 6e 75 78 20 78 38 36 5f 36 34 29 20 41 70 70	inux x86 _64) App
00d0	6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20	leWebKit /537.36
00e0	28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63	(KHTML, like Gec
00f0	6b 6f 29 20 43 68 72 6f 6d 65 2f 37 30 2e 30 2e	ko) Chro me/70.0.
0100	33 35 33 38 2e 31 30 32 20 53 61 66 61 72 69 2f	3538.102 Safari/
0110	35 33 37 2e 33 36 0d 0a 41 63 63 65 70 74 3a 20	537.36.. Accept:

Figura 3 – Pacote.

#### Resposta:

Desde o início da trama até ao carácter ASCII “G” são usados **66 B**. O tamanho total do pacote são **515 B** sendo que o tamanho da sobrecarga introduzida pela pilha protocolar são **66 B**, desta forma a percentagem de sobrecarga é dada por  $66 / 515 = 0.1282$  logo existe **12.82 %** de “overhead”.

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?

#### Resposta:

Verificamos que o campo FCS não está a ser usado, visto que a rede é composta por cabos, logo é bastante robusta garantindo transmissões de boa qualidade, não necessitando de deteção de erros.

A seguir responda às seguintes perguntas, baseado no conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP.

```

▼ Ethernet II, Src: Vmware_d2:19:f0 (00:0c:29:d2:19:f0), Dst: HewlettP_fc:6b:36 (a0:8c:fd:fc:6b:36)
  ► Destination: HewlettP_fc:6b:36 (a0:8c:fd:fc:6b:36)
  ► Source: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
  Type: IPv4 (0x0800)

```

Figura 4 – Pacote HTTP enviado pelo servidor.

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

**Resposta:**

O endereço Ethernet ou endereço MAC da fonte é o **00:0c:29:d2:19:f0** como se pode observar na região a vermelho (Figura 4), e corresponde ao servidor.

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

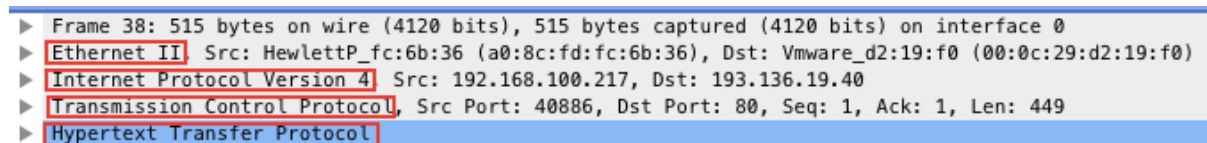
**Resposta:**

O endereço MAC do destino é o **a0:8c:fd:fc:6b:36** como se pode observar na região a azul (Figura 4) , e corresponde ao nosso computador.

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

**Resposta:**

De acordo com análise da captura identificou-se o protocolo **Ethernet II, IPv4, TCP, HTTP**.

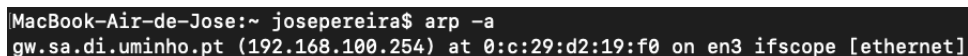


```
▶ Frame 38: 515 bytes on wire (4120 bits), 515 bytes captured (4120 bits) on interface 0
▶ Ethernet II, Src: HewlettP_fc:6b:36 (a0:8c:fd:fc:6b:36), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0)
▶ Internet Protocol Version 4, Src: 192.168.100.217, Dst: 193.136.19.40
▶ Transmission Control Protocol, Src Port: 40886, Dst Port: 80, Seq: 1, Ack: 1, Len: 449
▶ Hypertext Transfer Protocol
```

Figura 5 – Protocolos contidos na trama recebida.

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

**Resposta:**



```
MacBook-Air-de-Jose:~ josepereira$ arp -a
gw.sa.di.uminho.pt (192.168.100.254) at 0:c:29:d2:19:f0 on en3 ifscope [ethernet]
```

Figura 6 – Output do comando arp -a

A primeira coluna corresponde ao DNS - Domain Name System (**gw.sa.di.uminho.pt**) do endereço IP da segunda coluna (**192.168.100.254**), a terceira coluna corresponde ao endereço **MAC (0:c:29:d2:19:f0)**, e a quarta coluna identifica a interface rede que a máquina está a usar (**en3**) , e a última coluna identifica o protocolo **Ethernet** .

**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?

**Resposta:**

No.	Time	Source	Destination	Protocol	Length	Info
605	9.577228	AsustekC_29:f1:06	Broadcast	ARP	60	Who has 192.168.100.158? Tell 192.168.100.157
606	9.577253	Apple_45:c6:b0	AsustekC_29:f1:06	ARP	42	192.168.100.158 is at 38:c9:86:45:c6:b0

Figura 7 – Pacotes ARP capturados.

▶ Frame 605: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▼ Ethernet II, Src: AsustekC_29:f1:06 (70:8b:cd:29:f1:06), 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 factory default)
.... ...1 .... = IG bit: Group address (multicast/broadcast)
▼ Source: AsustekC_29:f1:06 (70:8b:cd:29:f1:06)
Address: AsustekC_29:f1:06 (70:8b:cd:29:f1:06)
.... ..0. .... = LG bit: Globally unique address (factory default)
.... ...0 .... = IG bit: Individual address (unicast)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
▶ Address Resolution Protocol (request)

Figura 8 – ARP Request.

O valor do endereço origem é **(70:8b:cd:29:f1:06)** e o destino é **(ff:ff:ff:ff:ff:ff)**. O endereço destino usado é o broadcast, uma vez que o pedido **ARP Request** é enviado a todos endereços da rede, com o objetivo do recetor identificar-se através de um **ARP Reply**, caso esse tenha o IP procurado.

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

**Resposta:**

▼ Ethernet II, Src: AsustekC_29:f1:06 (70:8b:cd:29:f1:06), 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 factory default)
.... ...1 .... = IG bit: Group address (multicast/broadcast)
▼ Source: AsustekC_29:f1:06 (70:8b:cd:29:f1:06)
Address: AsustekC_29:f1:06 (70:8b:cd:29:f1:06)
.... ..0. .... = LG bit: Globally unique address (factory default)
.... ...0 .... = IG bit: Individual address (unicast)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000

Figura 9 – O campo do Type.

O valor do campo Type é 0x0806 e indica o tipo de dados encapsulado, que neste caso corresponde ao protocolo ARP (Address Resolution Protocol).

**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>.

**Resposta:**

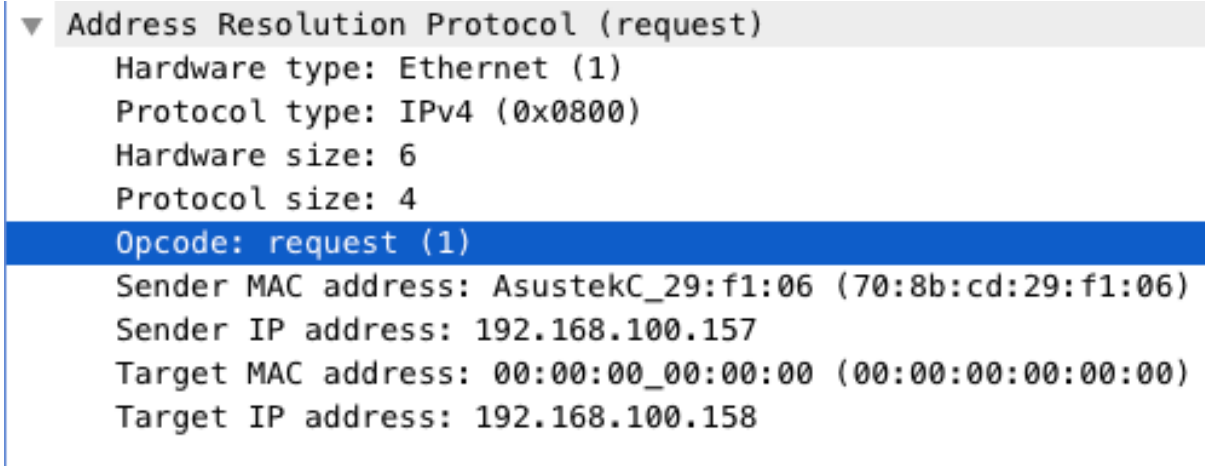


Figura 10 – Campo Opcode.

O valor do campo Opcode é 1 e especifica o tipo mensagem **ARP**, que pode ser **REQUEST** ou **REPLY**. Como se pode observar na Figura 10, o tipo neste caso é **REQUEST(1)**.

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

**Resposta:**

Os endereços contidos na mensagem **ARP** são: **Sender MAC address (70:8b:cd:29:f1:06)**, **Sender IP address (192.168.100.157)**, **Target MAC address (00:00:00:00:00:00)**, **Target IP address (192.168.100.158)**. Conclui-se que o campo **Target MAC address** está a zeros, pois ainda não foi encontrado o seu **MAC address**, pois isto é a mensagem **REQUEST(1)**.

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

**Resposta:**

O *host* de origem “questiona” todos os dispositivos (broadcast) conectados à rede, qual o dispositivo com o endereço IP procurado, que neste caso é o da nossa máquina (**192.168.100.158**).

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

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

Resposta:

```
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Apple_45:c6:b0 (38:c9:86:45:c6:b0)
  Sender IP address: 192.168.100.158
  Target MAC address: AsustekC_29:f1:06 (70:8b:cd:29:f1:06)
  Target IP address: 192.168.100.157
```

Figura 11 – Campo Opcode

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

Resposta:

A resposta do ARP está no campo Target **MAC address (70:8b:cd:29:f1:06)**, como se pode observar na Figura 11

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?

Resposta:

No.	Time	Source	Destination	Protocol	Length	Info
4	2.604424	Apple_45:c6:b0	Broadcast	ARP	42	Who has 192.168.100.100? Tell 0.0.0.0
5	2.925386	Apple_45:c6:b0	Broadcast	ARP	42	Who has 192.168.100.100? Tell 0.0.0.0
6	3.247982	Apple_45:c6:b0	Broadcast	ARP	42	Who has 192.168.100.100? Tell 0.0.0.0
7	3.570237	Apple_45:c6:b0	Broadcast	ARP	42	Gratuitous ARP for 192.168.100.100 (Request)
8	3.892690	Apple_45:c6:b0	Broadcast	ARP	42	Gratuitous ARP for 192.168.100.100 (Request)
10	4.213159	Apple_45:c6:b0	Broadcast	ARP	42	Gratuitous ARP for 192.168.100.100 (Request)
11	4.213647	Apple_45:c6:b0	Broadcast	ARP	42	Who has 192.168.100.254? Tell 192.168.100.100
12	4.213887	Vmware_d2:19:f0	Apple_45:c6:b0	ARP	60	192.168.100.254 is at 00:0c:29:d2:19:f0

Figura 12 – Pacotes ARP Gratuitos.

O campo **Sender IP address** e **Target IP address** têm o mesmo valor, uma vez que, o endereço procurado pelo pedido **ARP** é igual ao endereço da própria máquina.

O resultado esperado é não ter resposta, uma vez que, se houver significa que o **endereço IP** que sugerimos está a ser ocupado por outro dispositivo, o que origina conflitos.

Através do **ARP Gratuito** é esperado que os dispositivos de nível 2, que tenham tabelas de endereçamento **MAC**, como por exemplo **switchs**, ou mesmo **hosts** sejam atualizadas

- 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:**

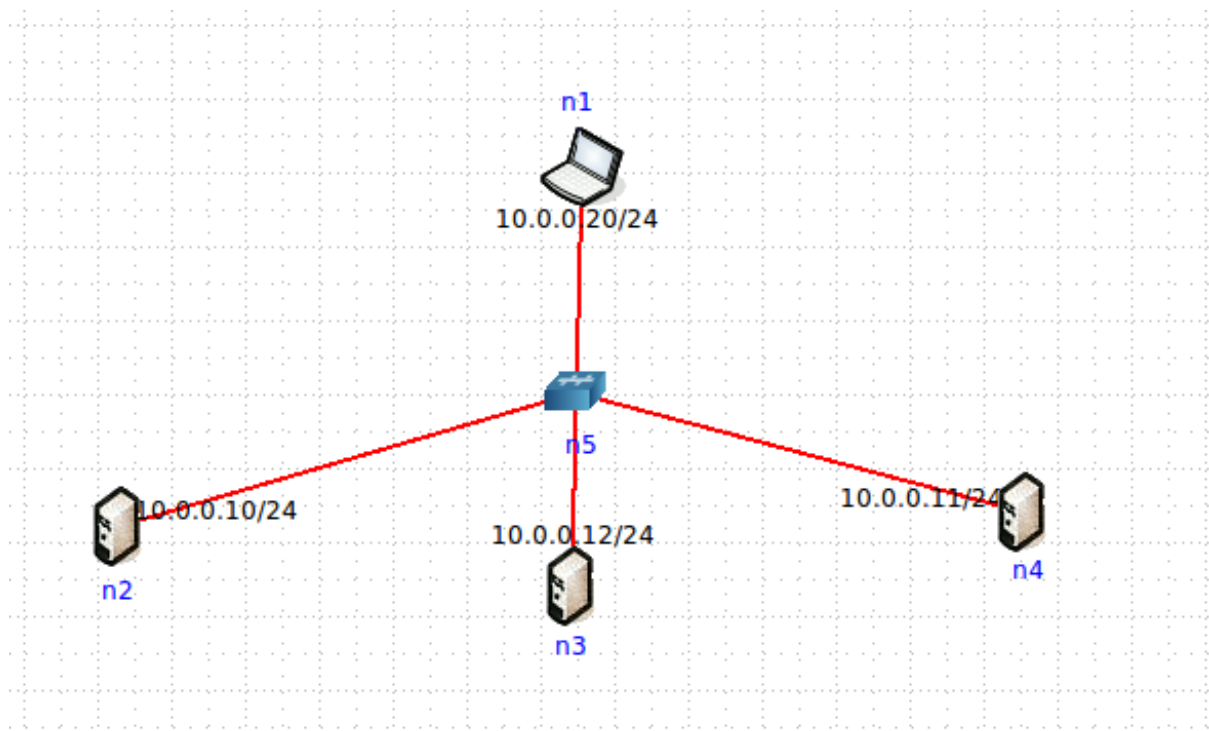


Figura 13 – Topologia Core com Hub.

```
root@n1: /tmp/pycore.42436/n1.conf
root@n1: /tmp/pycore.42436/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=0.282 ms
^C
--- 10.0.0.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.282/0.282/0.282/0.000 ms
root@n1: /tmp/pycore.42436/n1.conf#
```

Figura 14 – Comando ping através da Shell de n1.

```
vcmd
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:19:03.292767 ARP, Request who-has 10.0.0.10 tell 10.0.0.20, length 28
18:19:03.292821 ARP, Reply 10.0.0.10 is-at 00:00:00:aa:00:00, length 28
18:19:03.292848 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 28, seq 1, length 64
18:19:03.292892 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 28, seq 1, length 64
18:19:08.306979 ARP, Request who-has 10.0.0.20 tell 10.0.0.10, length 28
18:19:08.307081 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:01, length 28
```

Figura 15 – Tráfego capturado em n3.

Assim, conclui-se que com a utilização de um Hub, qualquer envio de mensagem entre dispositivos irá ser enviado para todos os dispositivos conectados à rede. Isto pode ser observado com as **Figuras 13 e 14**, onde se fez um **ping** do **laptop n1** para o **servidor n2 (10.0.0.10)**, e ao analisar o tráfego no **servidor n3**, verificou-se que este captura os pacotes enviados, de **n1** para **n2**, e o mesmo se verifica no servidor n4.

- 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:**

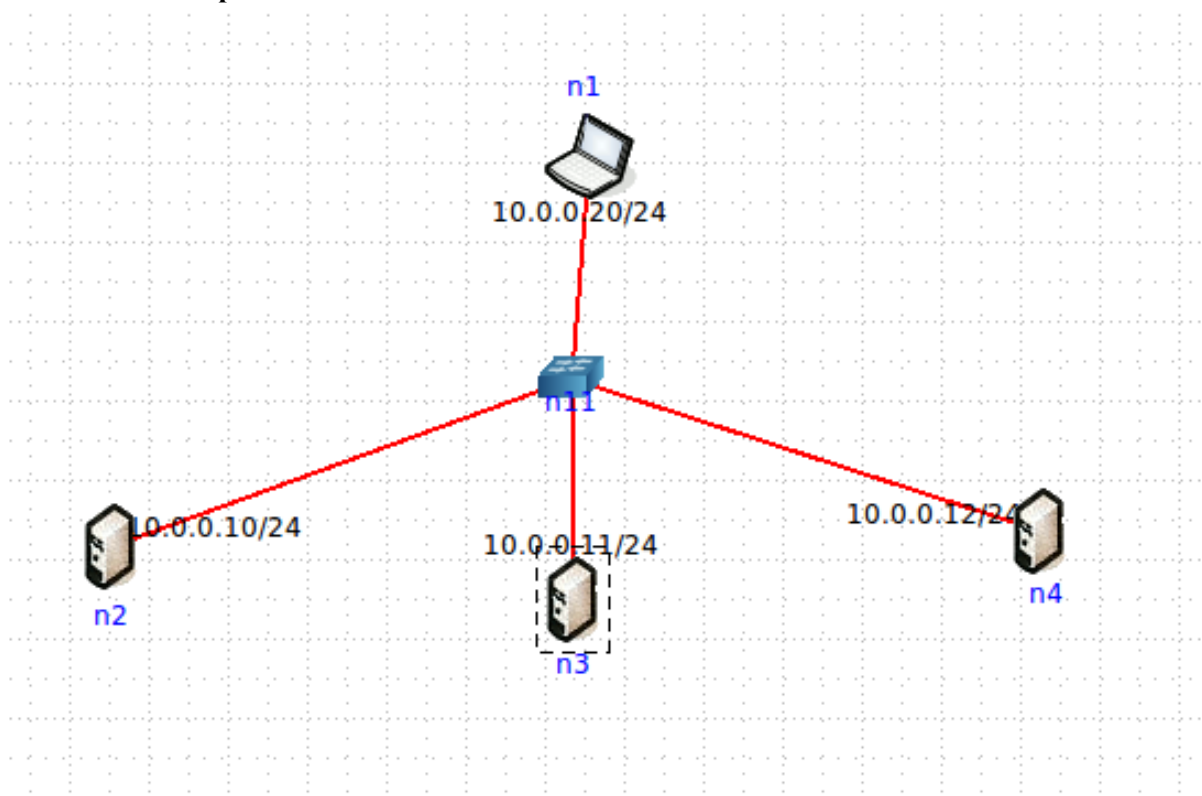


Figura 16 – Topologia Core com switch.



```
root@n1: /tmp/pycore.42438/n1.conf
root@n1:/tmp/pycore.42438/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=0.228 ms
64 bytes from 10.0.0.10: icmp_req=2 ttl=64 time=0.286 ms
^C
--- 10.0.0.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.228/0.257/0.286/0.029 ms
root@n1:/tmp/pycore.42438/n1.conf#
```

Figura 17 – Comando ping de n1 para n2 (10.0.0.10).

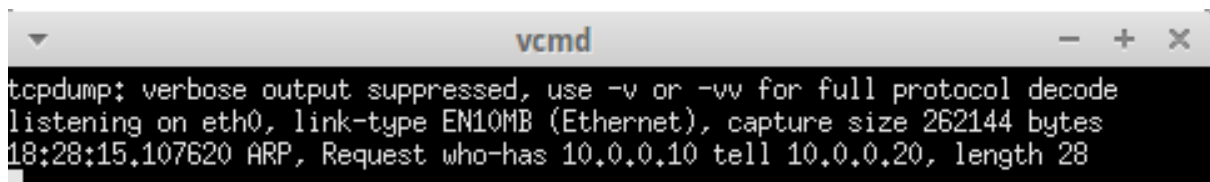
```
vcmd
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:28:15.107622 ARP, Request who-has 10.0.0.10 tell 10.0.0.20, length 28
18:28:15.107666 ARP, Reply 10.0.0.10 is-at 00:00:00:aa:00:05, length 28
18:28:15.107697 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 28, seq 1, length 64
18:28:15.107706 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 28, seq 1, length 64
18:28:16.109549 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 28, seq 2, length 64
18:28:16.109596 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 28, seq 2, length 64
18:28:20.114542 ARP, Request who-has 10.0.0.20 tell 10.0.0.10, length 28
18:28:20.114690 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:04, length 28
```

Figura 18 – Tráfego em n2.

Como se pode observar na **Figura 16** executou-se um **ping** do **laptop n1** para **servidor n2**. Deste modo, em **n2** foi capturado o tráfego que se pode observar na **Figura 17**, onde está incluído pacotes de **ARP request** e **reply**, responsáveis pela identificação dos MAC address dos dispositivos envolvidos e também os pacotes ICMP que retornam a informação do comando ping.

```
vcmd
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:28:15.107617 ARP, Request who-has 10.0.0.10 tell 10.0.0.20, length 28
```

Figura 19 – Tráfego em n3.



```
vcmd
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:28:15.107620 ARP, Request who-has 10.0.0.10 tell 10.0.0.20, length 28
```

Figura 20 – Tráfego em n4.

Ao contrário do aconteceu no exercício 17, como se está a utilizar um switch, significa que envios realizados entre dispositivos, apenas são capturados nestes.

Isto verifica-se nas **Figuras 18 e 19**, onde se pode observar que apenas se capturou o pacote **ARP Request**, pois este foi enviado para **broadcast**, no entanto, não receberam os **ARP Reply** e **ICMP** enviados pelo **servidor n2**, sendo que este respondeu ao **laptop n1** para informar que é o dispositivo com o **endereço IP** que ele procura.

Isto acontece, visto que o **switch** reserva uma porta única para cada dispositivo, evitando assim o envio de dados para dispositivos, que não são o destino pretendido e reduzir as colisões.

## Conclusões:

A realização do trabalho prático três permitiu uma melhor perceção dos conceitos **Ethernet** e do protocolo **ARP**. A nível de **Ethernet** aprendeu-se que todos os dispositivos são identificados por endereço único, associado à placa de rede (**NIC**) denominado por **MAC address**.

Compreendeu-se que a utilização do protocolo **ARP**, para identificação de **MAC address** numa rede, com mensagens de **ARP Request** e **Reply**, permite descobrir **os MAC address** dos dispositivos de modo que se consiga comunicar entre estes sem a necessidade da utilização do **endereço IP**.

O uso de **ARP Gratuitos**, permite aos dispositivos verificar a disponibilidade de um **endereço IP** numa rede, e também evitar a colisões entre mensagens dos mesmos.

Do mesmo modo, o **ARP Gratuito**, informa todos os dispositivos (**hosts e switches**) de um novo endereço, com objetivo destes atualizarem as suas tabelas.

Verificou-se também no último exercício, as técnicas utilizadas para o controlo de colisões, tais como a utilização de **switchs**, e as suas diferenças em relação aos **hubs**.