



Universidade do Minho  
Escola de Engenharia

## **Licenciatura em Engenharia Informática**

### **Trabalho Prático 3** **Redes de Computadores**

Ano Letivo de 2022/2023  
a100838, Jorge Teixeira  
a100066, Ricardo Jesus  
a100659, Rui Pinto  
Braga, Maio de 2023

# Índice

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Captura e análise de Tramas Ethernet . . . . .	2
<b>2</b>	<b>Resolução de Questões</b>	<b>4</b>
2.1	Questões sobre tramas Ethernet . . . . .	4
2.2	Questões sobre protocolo ARP . . . . .	5
2.3	Questões sobre domínios de colisão. . . . .	9
<b>3</b>	<b>Conclusões</b>	<b>12</b>

## Lista de Figuras

1.1	Número de ordem da sequência de bytes capturada. . . . .	2
1.2	Número de ordem da trama com a resposta proveniente do servidor ao cliente (linha 31 do wireshark). . . . .	3
2.1	Tráfego de informação entre cliente e servido . . . . .	4
2.2	Presença do protocolo TLS. . . . .	5
2.3	Topologia de rede . . . . .	5
2.4	Pings feitos aos pcs n6 e n7 a partir de n2. . . . .	6
2.5	ARP request. . . . .	6
2.6	Trama do broadcast no wireshark. . . . .	6
2.7	ARP protocol . . . . .	7
2.8	ARP reply . . . . .	7
2.9	comando ifconfig sobre o pc n2. . . . .	8
2.10	netstat -rn sobre o pc n2. . . . .	8
2.11	arp -a sobre o pc n2. . . . .	8
2.12	Print Wireshark. . . . .	8
2.13	Campos selecionados. . . . .	9
2.14	Esquema Cronológico . . . . .	9
2.15	ping n2 para n3. . . . .	10
2.16	TCPDump em n1. . . . .	10
2.17	ping n6 para n5. . . . .	10
2.18	TCPDump em n7. . . . .	10
2.19	Tabela do switch A . . . . .	11

# 1 Introdução

Neste trabalho prático iremos abordar questões relacionadas com a camada de ligação lógica, em específico o uso da tecnologia Ethernet e do protocolo ARP (Address Resolution Protocol).

## 1.1 Captura e análise de Tramas Ethernet

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	162.159.134.234	172.26.103.36	TLSv1.2	111	Application Data
2	0.000047	172.26.103.36	162.159.134.234	TCP	66	51486 → 443 [ACK] Seq=1 Ack=4294967126 Win=512 Len=0 SLE=1 SRE=58
3	0.001794	162.159.134.234	172.26.103.36	TCP	225	[TCP Out-Of-Order] 443 → 51486 [PSH, ACK] Seq=4294967126 Ack=1 Win=8 Len=171
4	0.001831	172.26.103.36	162.159.134.234	TCP	54	51486 → 443 [ACK] Seq=1 Ack=58 Win=511 Len=0
5	0.405175	172.26.103.36	142.250.200.68	TCP	66	52325 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
6	0.423782	142.250.200.68	172.26.103.36	TCP	66	443 → 52325 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 SACK_PERM WS=256
7	0.423998	172.26.103.36	142.250.200.68	TCP	54	52325 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
8	0.424816	172.26.103.36	142.250.200.68	TLSv1.3	819	Client Hello
9	0.446530	142.250.200.68	172.26.103.36	TCP	54	443 → 52325 [ACK] Seq=1 Ack=766 Win=67072 Len=0
10	0.466333	142.250.200.68	172.26.103.36	TLSv1.3	429	Server Hello, Change Cipher Spec, Application Data
11	0.466772	172.26.103.36	142.250.200.68	TLSv1.3	128	Change Cipher Spec, Application Data
12	0.492836	142.250.200.68	172.26.103.36	TCP	54	443 → 52325 [ACK] Seq=376 Ack=840 Win=67072 Len=0
13	0.653912	172.26.103.36	193.137.9.171	TLSv1.2	902	Application Data
14	0.678272	193.137.9.171	172.26.103.36	TCP	54	443 → 52303 [ACK] Seq=1 Ack=849 Win=5610 Len=0
15	0.706150	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=1 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
16	0.706150	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=1251 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
17	0.706230	172.26.103.36	193.137.9.171	TCP	54	52303 → 443 [ACK] Seq=849 Ack=2501 Win=512 Len=0
18	0.706543	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=2501 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
19	0.708092	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=3751 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
20	0.708142	172.26.103.36	193.137.9.171	TCP	54	52303 → 443 [ACK] Seq=849 Ack=5001 Win=512 Len=0
21	0.711888	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=5001 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
22	0.711888	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=6251 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
23	0.711888	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=7501 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
24	0.711888	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=8751 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
25	0.711888	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=10001 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
26	0.711949	172.26.103.36	193.137.9.171	TCP	54	52303 → 443 [ACK] Seq=849 Ack=11251 Win=512 Len=0
27	0.712927	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=11251 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
28	0.712927	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=12501 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
29	0.712927	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=13751 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
30	0.712927	193.137.9.171	172.26.103.36	TCP	1304	443 → 52303 [ACK] Seq=15001 Ack=849 Win=5610 Len=1250 [TCP segment of a reassembled PDU]
31	0.712927	193.137.9.171	172.26.103.36	TLSv1.2	1304	Application Data

Figura 1.1: Número de ordem da sequência de bytes capturada.

No intervalo de linhas 1 a 31, podemos ver que a ligação entre **Servidor** e **Cliente** é estabelecida ("Client Hello" e "Server Hello"), que se dá a primeira troca de dados entre Cliente e servidor e vice versa (Application Data, nas linhas 13 e 31, respetivamente).

No.	Time	Source	Destination	Protocol	Length	Info
31	0.712927	193.137.9.171	172.26.103.36	TLSv1.2	1304	Application Data

Frame 31: 1304 bytes on wire (10432 bits), 1304 bytes captured (10432 bits) on interface \Device\NPF\_{3F4678F9-7395-40D4-95A1-BF877F4907EF}, id 0

Ethernet II, Src: ComdaEnt\_ff:94:00 (00:d0:03:ff:94:00), Dst: IntelCor\_cd:5b:18 (8c:55:4a:cd:5b:18)

Destination: IntelCor\_cd:5b:18 (8c:55:4a:cd:5b:18)

Source: ComdaEnt\_ff:94:00 (00:d0:03:ff:94:00)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 193.137.9.171, Dst: 172.26.103.36

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 1290

Identification: 0x9a66 (39526)

010. .... = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 252

Protocol: TCP (6)

Header Checksum: 0x0114 [validation disabled]

[Header checksum status: Unverified]

Source Address: 193.137.9.171

Destination Address: 172.26.103.36

Transmission Control Protocol, Src Port: 443, Dst Port: 52303, Seq: 16251, Ack: 849, Len: 1250

Source Port: 443

Destination Port: 52303

[Stream index: 2]

[Conversation completeness: Incomplete (12)]

[TCP Segment Len: 1250]

Sequence Number: 16251 (relative sequence number)

Sequence Number (raw): 838969125

[Next Sequence Number: 17501 (relative sequence number)]

Acknowledgment Number: 849 (relative ack number)

Acknowledgment number (raw): 2626483561

0101 .... = Header Length: 20 bytes (5)

Flags: 0x010 (ACK)

Window: 5610

[Calculated window size: 5610]

[Window size scaling factor: -1 (unknown)]

Checksum: 0x73dc [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

[Timestamps]

[SEQ/ACK analysis]

TCP payload (1250 bytes)

TCP segment data (163 bytes)

[Reassembled PDU in frame: 43]

TCP segment data (1087 bytes)

[14 Reassembled TCP Segments (16413 bytes): #15(1250), #16(1250), #18(1250), #19(1250), #21(1250), #22(1250), #23(1250), #24(1250), #25(1250), #27(1250), #28(1250), #29(1250), #30(1250), #31(163)]

Transport Layer Security

TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol

Figura 1.2: Número de ordem da trama com a resposta proveniente do servidor ao cliente (linha 31 do wireshark).

## 2 Resolução de Questões

### 2.1 Questões sobre tramas Ethernet

#### 2.1.1 Anote os endereços MAC de origem e de destino da trama capturada. Identifique a que sistemas se referem. Justifique.

**R:** O endereço MAC origem é **8c:55:4a:cd:5b:18** e o endereço MAC destino é **00:d0:03:ff:94:00** correspondentes, respectivamente ao Cliente e ao Router LAN.

```
> Frame 13: 902 bytes on wire (7216 bits), 902 bytes captured (7216 bits) on interface \Device\NPF_{3
  Ethernet II, Src: IntelCor_cd:5b:18 (8c:55:4a:cd:5b:18), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
    > Destination: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
    > Source: IntelCor_cd:5b:18 (8c:55:4a:cd:5b:18)
    Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: 172.26.103.36, Dst: 193.137.9.171
  Transmission Control Protocol, Src Port: 52303, Dst Port: 443, Seq: 1, Ack: 1, Len: 848
  Transport Layer Security
```

Figura 2.1: Tráfego de informação entre cliente e servidor

#### 2.1.2 Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?

**R:** O valor hexadecimal do campo Type da trama Ethernet é 0x0800, este indica que se trata de um pacote IPV4.

#### 2.1.3 Quantos bytes são usados no encapsulamento protocolar, i.e. desde o início da trama até ao início dos dados do nível aplicacional (Application Data Protocol: http-over-tls, no caso de HTTPS)? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar.

**R:** Relativamente à trama anterior, esta possui 902 bytes sendo, deste valor os cabeçalhos:

- **Ethernet** = 14 + 4 = 18 bytes
- **IP** = 20 bytes
- **TCP** = 20 bytes
- **TLS** = 5 bytes

com um **Payload** de 838 bytes calculamos que o overhead introduzido pela pilha protocolar seja de 7,0% da totalidade de bytes.

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

**R:** Tal como é possível ver na figura 1.2, o endereço Ethernet da fonte é **00:d0:03:ff:94:00** que corresponde ao endereço MAC do router (visto que esta trama corresponde respectivamente à resposta do servidor para o cliente).

#### 2.1.5 Qual é o endereço MAC do destino? A que sistema (host) corresponde?

**R:** Na mesma imagem que verificamos na pergunta anterior (fig. 1.2) também conseguimos ver o endereço MAC, através do endereço Ethernet que lhe corresponde, sendo então este **8c:55:4a:cd:5b:18**.

**2.1.6 Atendendo ao conceito de encapsulamento protocolar, identifique os vários protocolos contidos na trama recebida. Justifique, indicando em que campos dos cabeçalhos capturados se baseou**

**R:** Continuando na análise da mesma figura das questões anteriores podemos ver que a trama possui os protocolos **Ethernet, IP, TCP e TLS**.

IP - É possível ver no campo "type" do cabeçalho Ethernet.

TCP - Presente no campo "protocol" no cabeçalho IP.

TLS - presente no campo "version" do próprio cabeçalho TLS. (visto a figura 1.2 não conseguir demonstrar completamente os campos do protocolo TLS fica em baixo a print com os restantes campos que não estavam visíveis).

```
▼ TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 16408
Encrypted Application Data: 42099e21969d9ace87adb639ca4bd2ba513d850b5c2556b5c30d2ea6592abc38945b624e...
[Application Data Protocol: Hypertext Transfer Protocol]
```

Figura 2.2: Presença do protocolo TLS.

## 2.2 Questões sobre protocolo ARP

Seguindo o enunciado construímos a seguinte topologia da rede.

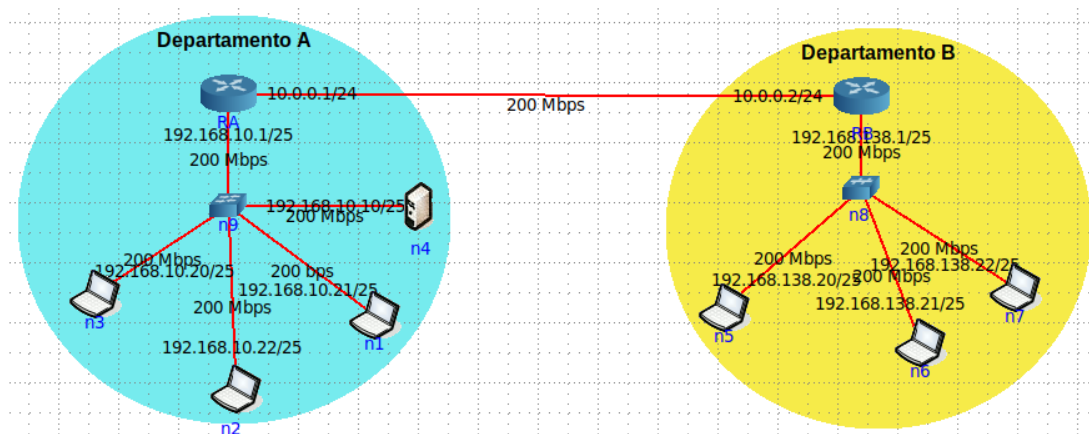


Figura 2.3: Topologia de rede

**2.2.1 Com a ajuda do manual ARP (man arp), interprete o significado de cada uma das colunas da tabela.**

**R:** Conseguimos ver, através da figura, que existe um IP que está associado a um endereço MAC. No caso, o tráfego que circula em **192.168.10.1** (isto é, tanto o tráfego que recebe, como o que envia) tem como origem/destino o endereço MAC **00:00:00:aa:00:00** e é recebido/enviado através de **eth0**. Assim, na nossa topologia, o tráfego que circula entre **n2** e **RA** fá-lo através da porta eth0.

```

root@n3:/tmp/pycore.44483/n3.conf# ping 192.168.138.21
PING 192.168.138.21 (192.168.138.21) 56(84) bytes of data.
64 bytes from 192.168.138.21: icmp_seq=1 ttl=62 time=0.858 ms
64 bytes from 192.168.138.21: icmp_seq=2 ttl=62 time=0.329 ms
^C
--- 192.168.138.21 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1030ms
rtt min/avg/max/mdev = 0.329/0.593/0.858/0.264 ms
root@n3:/tmp/pycore.44483/n3.conf# ping 192.168.138.22
PING 192.168.138.22 (192.168.138.22) 56(84) bytes of data.
64 bytes from 192.168.138.22: icmp_seq=1 ttl=62 time=0.681 ms
64 bytes from 192.168.138.22: icmp_seq=2 ttl=62 time=0.258 ms
^C
--- 192.168.138.22 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.258/0.469/0.681/0.211 ms

```

Figura 2.4: Pings feitos aos pcs n6 e n7 a partir de n2.

```

root@n3:/tmp/pycore.44483/n3.conf# arp -a
? (192.168.10.1) at 00:00:00:aa:00:00 [ether] on eth0

```

Figura 2.5: ARP request.

### 2.2.2 Indique, justificando, qual o equipamento da intranet em causa que poderá apresentar a maior tabela ARP em termos de número de entradas.

**R:** O equipamento da intranet em causa que poderá apresentar a maior tabela ARP em termos de número de entradas é o router RA, isto porque ele vai fazer a ligação da sua subrede (que é a maior) à subrede do departamento B, que adicionará um maior número de entradas à tabela arp.

### 2.2.3 Qual é o valor hexadecimal dos endereços MAC origem e destino? Como interpreta e justifica o endereço destino usado?

**R:** Os valores hexadecimais estão presentes na figura seguinte. sendo o endereço MAC origem **00:00:00:aa:00:04** e destino **00:00:00:00:00:00**. O endereço destino utilizado é o de broadcast porque nesse momento a tabela ainda não tem uma entrada, na tabela arp, para esse endereço. Deste modo envia para todas as portas de moodo a garantir uma resposta.

```

▼ Ethernet II, Src: 00:00:00_aa:00:04 (00:00:00:aa:00:04), 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: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
    Address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
  Sender IP address: 192.168.10.20
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.10.1

```

Figura 2.6: Trama do broadcast no wireshark.

### 2.2.4 Qual o valor hexadecimal do campo Tipo da trama Ethernet? O que indica?

**R:** O valor hexadecimal do campo tipo da trama é **0x0806**, que nos indica que o encapsualemnto protocolar seguinte da trama é o protocolo ARP.

### 2.2.5 Observando a mensagem ARP, como pode saber que se trata efetivamente de um pedido ARP? Refira duas formas distintas de obter essa informação

**R:** É possível verificar que se trata de um pedido ARP dado o Opcode ser (request) 1 e o Target Mac address e IP address não serem nulos (neste caso são o broadcast).

```
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
  Sender IP address: 192.168.10.20
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.10.1
```

Figura 2.7: ARP protocol

### 2.2.6 Explícite, em linguagem comum, que tipo de pedido ou pergunta é feita pelo host de origem à rede?

**R:** O host manda um pedido aos vários dispositivos da rede para saber o endereço MAC do dispositivo que quer comunicar, o dispositivo com o endereço IP solicitado, responde com o seu endereço MAC.

### 2.2.7 Qual o valor do campo ARP opcode? O que especifica?

**R:** O valor do campo Opcode na mensagem ARP de resposta é 2 (reply).

```
▶ Frame 6: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth3.0.6e, id 0
▼ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
  ▼ Destination: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
    Address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    Address: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  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: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    Sender IP address: 192.168.10.1
    Target MAC address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
    Target IP address: 192.168.10.20
0000 00 00 00 aa 00 04 00 00 00 aa 00 00 08 06 00 01 .....
0010 08 00 06 04 00 02 00 00 00 aa 00 00 c0 a8 0a 01 .....
0020 00 00 00 aa 00 04 c0 a8 0a 14 .....
```

Figura 2.8: ARP reply

### 2.2.8 Em que posição da mensagem ARP está a resposta ao pedido ARP efetuado?

**R:** A resposta ao pedido ARP efetuado encontra-se no MAC address do remetente (sender), pois esta é enviada através do dispositivo que contém a resposta do ARP request anterior.



## 2.2.9 Identifique a que sistemas correspondem os endereços MAC de origem e de destino da trama em causa, recorrendo aos comandos ifconfig, netstat -rn e arp executados no PC selecionado

**R:** Sendo o destino o pc n2 e a origem o router RA

```
root@n2:/tmp/pycore.44483/n2.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.22 netmask 255.255.255.128 broadcast 0.0.0.0
    inet6 2001::22 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:5 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:00:06 txqueuelen 1000 (Ethernet)
    RX packets 7535 bytes 605204 (605,2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19 bytes 1522 (1,5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0,0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0,0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 2.9: comando ifconfig sobre o pc n2.

```
root@n2:/tmp/pycore.44483/n2.conf# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 192.168.10.1 0.0.0.0 UG 0 0 0 eth0
192.168.10.0 0.0.0.0 255.255.255.128 U 0 0 0 eth0
```

Figura 2.10: netstat -rn sobre o pc n2.

```
root@n2:/tmp/pycore.44483/n2.conf# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 192.168.10.1 0.0.0.0 UG 0 0 0 eth0
192.168.10.0 0.0.0.0 255.255.255.128 U 0 0 0 eth0
```

Figura 2.11: arp -a sobre o pc n2.

## 2.2.10 Justifique o modo de comunicação (unicast vs. broadcast) usado no envio da resposta ARP (ARP Reply).

**R:** No modo de comunicação unicast a mensagem é enviada diretamente para um dispositivo específico, diretamente, enquanto que no modo broadcast o dispositivo não é especificado e a mensagem é enviada para todas as portas do dispositivo. Neste caso o modo unicast é utilizado porque no caso do ARP reply toda a informação necessária (IP e MAC) sobre o destino já é conhecida.

## 2.2.11 Verifique se o ping feito ao segundo PC originou pacotes ARP. Justifique a situação observada.

**R:** O ping feito ao segundo PC não originou pacotes ARP, pois já foi criada a entrada na tabela ARP no primeiro ping.

5	5.136377652	00:00:00_aa:00:04	Broadcast	ARP	42 Who has 192.168.10.1? Tell 192.168.10.20
6	5.136939062	00:00:00_aa:00:00	00:00:00_aa:00:04	ARP	42 192.168.10.1 is at 00:00:00_aa:00:00
22	10.326213336	00:00:00_aa:00:00	00:00:00_aa:00:04	ARP	42 Who has 192.168.10.20? Tell 192.168.10.1
23	10.326226448	00:00:00_aa:00:04	00:00:00_aa:00:00	ARP	42 192.168.10.20 is at 00:00:00_aa:00:04

Figura 2.12: Print Wireshark.

## 2.2.12 Identifique na mensagem ARP os campos que permitem definir o tipo e o tamanho dos endereços das camadas de rede e de ligação lógica que se pretendem mapear. Justifique os valores apresentados nesses campos.

**R:** Os campos que permitem definir o tipo e o tamanho são:

- Hardware size:6
- Protocol size: 4

```

> Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth3.0.6e, id 0
> Ethernet II, Src: 00:00:00_aa:00:04 (00:00:00:aa:00:04), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
    Sender IP address: 192.168.10.20
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.10.1

```

Figura 2.13: Campos selecionados.

**2.2.13** Na situação em que efetua um ping a um PC não local à sua sub-rede, esboce um diagrama em que indique claramente, e de forma cronológica, todas as mensagens ARP e ICMP trocadas, até à recepção da resposta ICMP do sistema destino (represente apenas os nós intervenientes). Assuma que todas as tabelas ARP se encontram inicialmente vazias.

**R:** Sendo n2 (192.168.10.22) representado pelo "PC" mais à esquerda, n6 (192.168.138.21) pelo "PC" mais à direita do esquema, RA (192.168.10.1) sendo o router A e RB (10.0.0.2) sendo o router B.

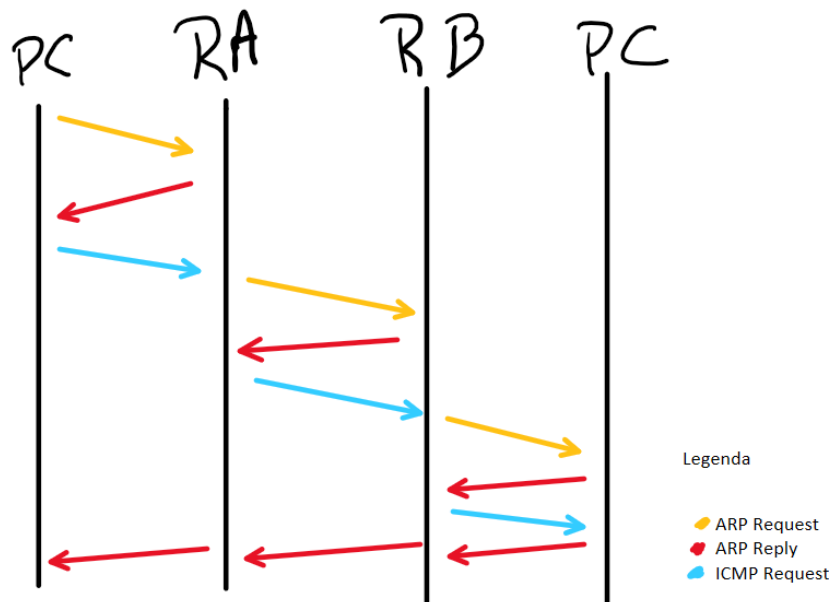


Figura 2.14: Esquema Cronológico

## 2.3 Questões sobre domínios de colisão.

**2.3.1** Através da opção tcpdump, verifique e compare como flui o tráfego nas diversas interfaces dos vários dispositivos no departamento A (LAN comutada) e no departamento B (LAN partilhada) quando é gerado tráfego intra-departamento (por exemplo, através do comando ping). Que conclui? 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

**R:** Começamos por testar a LAN do Departamento A fazendo um ping de n2 para n3 com o TCPDump em n1. Tentamos verificar se n1 recebia algum do tráfego que fosse enviado a n3.

```

root@n2:/tmp/pycore.44483/n2.conf# ping 192.168.10.20
PING 192.168.10.20 (192.168.10.20) 56(84) bytes of data.
64 bytes from 192.168.10.20: icmp_seq=1 ttl=64 time=0.580 ms
64 bytes from 192.168.10.20: icmp_seq=2 ttl=64 time=0.111 ms
64 bytes from 192.168.10.20: icmp_seq=3 ttl=64 time=0.583 ms
^C
--- 192.168.10.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.111/0.424/0.583/0.221 ms

```

Figura 2.15: ping n2 para n3.

```

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:35:38.424555 IP6 fe80::3884:59ff:fe8a:4448 > ff02::16: HBH ICMP6, multicast l
istener report v2, 2 group record(s), length 48
21:35:46.543964 IP6 fe80::3884:59ff:fe8a:4448.5353 > ff02::fb.5353: 0* [0q] 2/0
/0 (Cache flush) PTR xubuncore.local., (Cache flush) AAAA fe80::3884:59ff:fe8a:4
448 (141)
21:35:50.328195 IP6 fe80::200:ff:feaa:5 > ff02::2: ICMP6, router solicitation, l
ength 16
21:35:50.344293 IP6 fe80::200:ff:feaa:6 > ff02::16: HBH ICMP6, multicast listene
r report v2, 2 group record(s), length 48
21:35:55.223992 IP6 fe80::3884:59ff:fe8a:4448.5353 > ff02::fb.5353: 0 [2q] PTR (
0W)? -ipps.top.local. PTR (0W)? -ipps.top.local. (45)
21:35:59.503959 IP6 fe80::4c7c:73ff:fe66:a536.5353 > ff02::fb.5353: 0 [2q] PTR (
0W)? -ipps.top.local. PTR (0W)? -ipps.top.local. (45)
21:36:02.624370 IP 192.168.10.1 > 224.0.0.5: OSPFv2, Hello, length 44
21:36:10.744356 IP6 fe80::3884:59ff:fe8a:4448.5353 > ff02::fb.5353: 0* [0q] 2/0
/0 (Cache flush) PTR xubuncore.local., (Cache flush) AAAA fe80::3884:59ff:fe8a:4
448 (141)

```

Figura 2.16: TCPDump em n1.

Como conseguimos ver, o router n1 não recebe qualquer pacote ICMP, o que significa que o switch (n9) não partilha o tráfego com os elementos da rede (LAN comutada). Apesar disso conseguimos ver que um pacote é recebido que é o ARP request (foi previamente emitido em modo Broadcast).

De seguida, tentamos fazer o mesmo para o Departamento B, isto é, fazer um ping de n6 para n5 com o TCPDump em n7. Novamente, tentamos verificar se n7 consegue capturar algum tráfego com destino a n5.

```

root@n6:/tmp/pycore.44483/n6.conf# ping 192.168.138.20
PING 192.168.138.20 (192.168.138.20) 56(84) bytes of data.
64 bytes from 192.168.138.20: icmp_seq=1 ttl=64 time=1.84 ms
64 bytes from 192.168.138.20: icmp_seq=2 ttl=64 time=0.683 ms
64 bytes from 192.168.138.20: icmp_seq=3 ttl=64 time=0.823 ms
^C
--- 192.168.138.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.683/1.115/1.841/0.516 ms

```

Figura 2.17: ping n6 para n5.

```

21:50:36.739889 ARP, Request who-has 192.168.138.20 tell 192.168.138.21, length
28
21:50:36.740265 ARP, Reply 192.168.138.20 is-at 00:00:00:aa:00:08, length 28
21:50:36.740610 IP 192.168.138.21 > 192.168.138.20: ICMP echo request, id 27, se
q 1, length 64
21:50:36.741180 IP 192.168.138.20 > 192.168.138.21: ICMP echo reply, id 27, seq
1, length 64
21:50:37.741069 IP 192.168.138.21 > 192.168.138.20: ICMP echo request, id 27, se
q 2, length 64
21:50:37.741672 IP 192.168.138.20 > 192.168.138.21: ICMP echo reply, id 27, seq
2, length 64
21:50:38.267483 IP 192.168.138.1 > 224.0.0.5: OSPFv2, Hello, length 44
21:50:38.749359 IP 192.168.138.21 > 192.168.138.20: ICMP echo request, id 27, se
q 3, length 64
21:50:38.750688 IP 192.168.138.20 > 192.168.138.21: ICMP echo reply, id 27, seq
3, length 64
21:50:40.258033 IP 192.168.138.1 > 224.0.0.5: OSPFv2, Hello, length 44
21:50:41.885327 ARP, Request who-has 192.168.138.21 tell 192.168.138.20, length
28
21:50:41.885806 ARP, Reply 192.168.138.21 is-at 00:00:00:aa:00:09, length 28
21:50:42.180880 IP6 fe80::200:ff:feaa:1 > ff02::5: OSPFv3, Hello, length 36
21:50:42.259039 IP 192.168.138.1 > 224.0.0.5: OSPFv2, Hello, length 44
21:50:44.260333 IP 192.168.138.1 > 224.0.0.5: OSPFv2, Hello, length 44

```

Figura 2.18: TCPDump em n7.

Neste caso conseguimos ver que o n7 recebe os pacotes ICMP reply/request o que significa que o hub(n8) partilha o tráfego com os elementos da rede (LAN partilhada).

Estas LAN's têm diferenças que são características do switch e hub (n9 e n8, respetivamente). O switch mantém uma tabela ARP com os endereços MAC que faz com que transmissão seja Unicast e o hub como não cria a tabela faz com que a transmissão seja feita em Broadcast.

### 2.3.2 Construa manualmente a tabela de comutação do switch do Departamento A, atribuindo números de porta à sua escolha.

**R:** Preenchendo manualmente a tabela de comutação do switch do Departamento A, obtivemos a seguinte tabela :

IP	MAC Address	Interface
192.168.10.1	00:00:00:aa:00:00	eth0
192.168.10.10	00:00:00:aa:00:07	eth1
192.168.10.21	00:00:00:aa:00:05	eth2
192.168.10.22	00:00:00:aa:00:06	eth3
192.168.10.20	00:00:00:aa:00:04	eth4

Figura 2.19: Tabela do switch A

Existem outros campos que fazem parte de uma tabela ARP, a nossa tabela apresenta apenas estes 3 porque foram os que consideramos mais relevantes para o problema porposto.

### **3 Conclusões**

Através do estudo dos temas abordados neste trabalho como a análise do formato de um cabeçalho Ethernet, os endereços MAC, o protocolo ARP e os domínios de colisão conseguimos compreender como os dispositivos de rede comunicam uns com os outros e como a informação é transmitida de um ponto até ao outro.

Aprendemos que o cabeçalho Ethernet é composto por várias informações importantes, como endereços de origem e destino, tipo de protocolo e tamanho do pacote. O endereço MAC é uma identificação única atribuída a cada dispositivo de rede. O protocolo ARP é utilizado para fazer a correspondência entre endereços IP e endereços MAC, e os domínios de colisão podem afetar a eficiência da rede, especialmente em redes de grande dimensão.

Em conclusão, através do estudo e realização deste trabalho conseguimos perceber mais sobre o funcionamento das camadas protocolares da pilha (protocolar).