

**Universidade do Minho**

Escola de Engenharia

Licenciatura em Engenharia Informática

## **Unidade Curricular de Redes de Computadores**

Ano Letivo de 2022/2023

### **Trabalho Prático Nº3** **Redes Ethernet e Protocolo ARP**

#### **Grupo 57**

A94942 Miguel Velho Raposo

A78823 João Carlos Cotinho Sotomaior Neto

A91775 José Pedro Batista Fonte

5 de maio de 2023

# Índice

Lista de Figuras . . . . .	3
<b>1 Introdução</b>	<b>4</b>
<b>2 Trabalho Desenvolvido</b>	<b>5</b>
2.0.1 Captura e Análise de Tramas Ethernet . . . . .	6
2.0.2 Protocolo ARP . . . . .	8
2.0.3 Domínios de colisão . . . . .	13
<b>3 Conclusão</b>	<b>15</b>

## Lista de Figuras

2.1	Tráfego entre Cliente e Servidor . . . . .	5
2.2	IPs do Servidor e Cliente . . . . .	5
2.3	Tráfego entre Cliente e Servidor . . . . .	6
2.4	Trama Application Data do Servidor para Cliente . . . . .	7
2.5	Topologia da rede . . . . .	8
2.6	ARP Request . . . . .	8
2.7	ARP Request . . . . .	9
2.8	ARP Request . . . . .	9
2.9	ARP Reply . . . . .	10
2.10	Comandos A1 . . . . .	11
2.11	ARP Reply . . . . .	12
2.12	Diagrama Cronológico . . . . .	12
2.13	Análise da LAN do departamento A . . . . .	13
2.14	Análise da LAN do departamento A . . . . .	13
2.15	Tabela do switch A . . . . .	14

# 1 Introdução

Este relatório tem como objetivo explorar a camada de ligação lógica em redes de computadores, com um foco especial na tecnologia Ethernet e no protocolo ARP (Address Resolution Protocol). A camada de ligação lógica é fundamental para a comunicação de dados numa rede, uma vez que é responsável pela transferência de dados entre nós adjacentes. Neste relatório, será abordado como os serviços oferecidos pela camada de ligação lógica se relacionam com as camadas superiores e inferiores da pilha protocolar. Além disso, serão discutidos os principais aspectos da tecnologia Ethernet, bem como o funcionamento do protocolo ARP, que é utilizado para o mapeamento de endereços MAC de uma rede.

## 2 Trabalho Desenvolvido

1434	18.892768	172.26.103.9	193.137.9.171	TCP	66 52300 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1435	18.894322	172.26.103.9	193.137.9.171	TCP	66 52301 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1438	18.895325	193.137.9.171	172.26.103.9	TCP	66 443 → 52300 [SYN, ACK] Seq=0 Ack=1 Win=12500 Len=0 MSS=1250 WS=4 SACK_PERM
1439	18.895375	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
1440	18.895656	172.26.103.9	193.137.9.171	TLSv1.2	571 Client Hello
1442	18.897151	193.137.9.171	172.26.103.9	TCP	66 443 → 52301 [SYN, ACK] Seq=0 Ack=1 Win=12500 Len=0 MSS=1250 WS=4 SACK_PERM
1443	18.897262	172.26.103.9	193.137.9.171	TCP	54 52301 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
1444	18.897592	172.26.103.9	193.137.9.171	TLSv1.2	571 Client Hello
1445	18.897857	193.137.9.171	172.26.103.9	TCP	54 443 → 52300 [ACK] Seq=1 Ack=518 Win=13016 Len=0
1446	18.100152	193.137.9.171	172.26.103.9	TLSv1.2	201 Server Hello, Change Cipher Spec, Encrypted Handshake Message
1447	18.100509	193.137.9.171	172.26.103.9	TCP	54 443 → 52301 [ACK] Seq=1 Ack=518 Win=13016 Len=0
1448	18.102019	172.26.103.9	193.137.9.171	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
1449	18.102129	193.137.9.171	172.26.103.9	TLSv1.2	201 Server Hello, Change Cipher Spec, Encrypted Handshake Message
1450	18.102175	172.26.103.9	193.137.9.171	TLSv1.2	798 Application Data
1451	18.102413	172.26.103.9	193.137.9.171	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
1452	18.103771	193.137.9.171	172.26.103.9	TCP	54 443 → 52300 [ACK] Seq=148 Ack=569 Win=13068 Len=0
1453	18.104022	193.137.9.171	172.26.103.9	TCP	54 [TCP Dup ACK 1452#1] 443 → 52300 [ACK] Seq=148 Ack=569 Win=13068 Len=0
1454	18.104152	193.137.9.171	172.26.103.9	TCP	54 443 → 52301 [ACK] Seq=148 Ack=569 Win=13068 Len=0
1455	18.104434	193.137.9.171	172.26.103.9	TCP	54 443 → 52300 [ACK] Seq=148 Ack=1313 Win=13812 Len=0
1456	18.104692	193.137.9.171	172.26.103.9	TCP	54 [TCP Dup ACK 1454#1] 443 → 52301 [ACK] Seq=148 Ack=569 Win=13068 Len=0
1496	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=148 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1497	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=1398 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1498	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=2648 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1499	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=3898 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1500	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=5148 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1501	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=6398 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1502	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=7648 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1503	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=8898 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1504	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=10148 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1505	18.334676	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [PSH, ACK] Seq=11398 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1506	18.334676	193.137.9.171	172.26.103.9	TCP	201 443 → 52300 [PSH, ACK] Seq=12648 Ack=1313 Win=13812 Len=147 [TCP segment of a reassembled PDU]
1507	18.334805	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1313 Ack=2648 Win=131072 Len=0
1508	18.334861	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1313 Ack=5148 Win=131072 Len=0
1509	18.334894	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1313 Ack=7648 Win=131072 Len=0
1510	18.334927	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1313 Ack=10148 Win=131072 Len=0
1511	18.334959	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1313 Ack=12648 Win=131072 Len=0
1512	18.337135	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=12795 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1513	18.337135	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=14045 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1514	18.337243	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1313 Ack=14045 Win=131072 Len=0
1515	18.337340	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=15295 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1516	18.337340	193.137.9.171	172.26.103.9	TLSv1.2	1304 Application Data
1517	18.337386	172.26.103.9	193.137.9.171	TCP	54 52300 → 443 [ACK] Seq=1313 Ack=16545 Win=131072 Len=0
1518	18.337754	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=17795 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1519	18.337754	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=19045 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1520	18.337754	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=20295 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1521	18.337754	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [PSH, ACK] Seq=21545 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]
1522	18.337754	193.137.9.171	172.26.103.9	TCP	1304 443 → 52300 [ACK] Seq=22795 Ack=1313 Win=13812 Len=1250 [TCP segment of a reassembled PDU]

Figura 2.1: Tráfego entre Cliente e Servidor

```

C:\Users\josef>nslookup
Default Server:  dns3.uminho.pt
Address: 193.137.16.65

> alunos.uminho.pt
Server: dns3.uminho.pt
Address: 193.137.16.65

Name: alunos.uminho.pt
Address: 193.137.9.171

```

```

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : eduroam.uminho.pt
Description . . . . . : Killer(R) Wi-Fi 6 AX1650x 160MHz Wireless Network Adapter (200NGW)
Physical Address. . . . . : 38-00-25-95-20-AB
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::f8d3:dcf2:877a:9838W7(Preferred)
IPv4 Address. . . . . : 172.26.103.9(Preferred)
Subnet Mask . . . . . : 255.255.0.0
Lease Obtained. . . . . : 3 de maio de 2023 10:23:13
Lease Expires . . . . . : 10 de maio de 2023 15:18:27
Default Gateway . . . . . : 172.26.254.254
DHCP Server . . . . . : 1.1.1.10
DHCPv6 IAID . . . . . : 104333349
DHCPv6 Client DUID . . . . . : 00-01-00-01-24-DA-35-3C-38-65-EC-C8-AE-52
DNS Servers . . . . . : 193.137.16.65
                        193.137.16.145
                        193.137.16.75
NetBIOS over Tcpip. . . . . : Enabled

```

(a) IP do server

(b) Endereço IP e MAC do Cliente

Figura 2.2: IPs do Servidor e Cliente

Para filtrar o tráfego, estabeleceu-se um filtro para mostrar só o tráfego vindo do servidor (193.137.9.171). Observando o tráfego deduz-se que cliente tem IP 172.26.103.9.

A conexão entre cliente e servidor é estabelecida entre a linha 1434 e 1442, através de um 3-way-handshake, visto que a conexão se trata de HTTP-over-TLS as linhas seguintes referem-se ao protocolo TLS. Os primeiros dados aplicativos entre cliente e servidor é a trama *Aplicação*

Data na linha 1450 e os primeiros entre servidor e cliente é a trama *Application Data* na linha 1516.

## 2.0.1 Captura e Análise de Tramas Ethernet

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

- **Endereço MAC de Origem** : 38:00:25:95:2d:ab (Client)
- **Endereço MAC de Destino** : 00:d0:03:ff:94:00 (Router LAN)

```
> Frame 1450: 798 bytes on wire (6384 bits), 798 bytes captured (6384 bits) on interface \Device\NPF_{3A2F04C2-F4D3-4E58-B008-679005EAA4F98}, id 0
  ✓ Ethernet II, Src: IntelCor_95:2d:ab (38:00:25:95:2d:ab), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
    > Destination: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
    > Source: IntelCor_95:2d:ab (38:00:25:95:2d:ab)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 172.26.103.9, Dst: 193.137.9.171
  > Transmission Control Protocol, Src Port: 52300, Dst Port: 443, Seq: 569, Ack: 148, Len: 744
  > Transport Layer Security
```

Figura 2.3: Tráfego entre Cliente e Servidor

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

O valor do campo `type` é 0x0800 (IPV4) que indica se tratar de um datagrama IP.

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.

Analisando a trama de 798 bytes esta apresenta a seguinte pilha protocolar com os seus overheads:

- Cabeçalho Ethernet : 14 bytes + 4 bytes de FCS
- Cabeçalho IP : 20 bytes
- Cabeçalho TCP : 20 bytes
- Cabeçalho TLS : 5 bytes
- Payload : 734 bytes

Assim sendo, o overhead introduzido pela pilha protocolar representa 7,8% do total.

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

```

Frame 1516: 1304 bytes on wire (10432 bits), 1304 bytes captured (10432 bits) on interface 'Device\NPF_{3A2F04C2-F403-4E58-B808-679D05EA4F90}.id 0
  Ethernet II, Src: ComdantH_ff:94:00 (00:0d:03:ff:94:00), Dst: IntelCor_95:2d:ab (38:00:25:95:2d:ab)
    Destination: IntelCor_95:2d:ab (38:00:25:95:2d:ab)
    Source: ComdantH_ff:94:00 (00:0d:03:ff:94:00)
    Type: IP4 (0x0800)
  Internet Protocol Version 4, Src: 193.137.9.171, Dst: 172.26.103.9
  Transmission Control Protocol, Src Port: 443, Dst Port: 52300, Seq: 16545, Ack: 1313, Len: 1250
  [15 Reassembled TCP Segments (16413 bytes): #1496(1250), #1497(1250), #1498(1250), #1499(1250), #1500(1250), #1501(1250), #1503(1250), #1504(1250), #1505(1250), #1506(147), #15
  Transport Layer Security

```

Figura 2.4: Trama Application Data do Servidor para Cliente

O endereço Ethernet da fonte é 00:d0:03:ff:94:00 que corresponde ao endereço MAC do router da isto porque esta trama vem do servidor para o cliente,

5. Qual é o endereço MAC do destino? A que sistema (host) corresponde?

O endereço Ethernet do destino é 38:00:25:95:2d:ab que corresponde ao endereço MAC do cliente.

**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.**

Os diferentes protocolos contidos são **Ethernet-IP-TCP-TLS**, isto foi baseado nos seguintes campos:

- **Protocolo IP** : presente no campo *Type* do cabeçalho Ethernet
- **Protocolo TCP** : presente no campo *Protocol* do cabeçalho IP
- **Protocolo TLS** : presente no campo *version* do cabeçalho TLS

## 2.0.2 Protocolo ARP

O departamento A ficou com endereço IP 192.168.57.0/25 com os hosts:

- **PC A1** : 192.168.57.21
- **PC A2** : 192.168.57.22
- **PC A3** : 192.168.57.23
- **PC SA1** : 192.168.57.30

O departamento B ficou com endereço IP 192.168.185.0/25 com os hosts:

- **PC B1** : 192.168.185.20
- **PC B2** : 192.168.185.21
- **PC B3** : 192.168.185.22

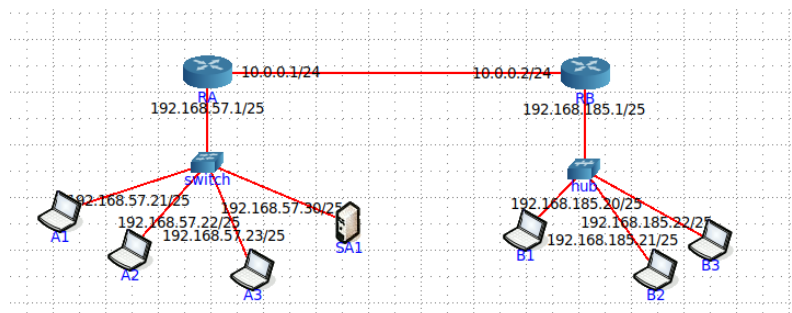


Figura 2.5: Topologia da rede

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

```
root@A1:/tmp/pycore.37653/A1.conf# arp -a
? (192.168.57.1) at 00:00:00:aa:00:02 [ether] on eth0
root@A1:/tmp/pycore.37653/A1.conf#
```

Figura 2.6: ARP Request

No exemplo da figura denota-se que existe um endereço IP, um endereço MAC e uma porta de interface. O endereço IP está associado ao endereço MAC, o que significa que o tráfego enviado para esse endereço IP tem um dispositivo com esse endereço MAC. Por último, a interface significa a porta do dispositivo de origem por onde circula o tráfego de/para aquele endereço IP.

Assim sendo, neste caso o tráfego de/para 192.168.57.1 tem como endereço MAC origem/destino 00:00:00:aa:00:02 e é recebido/enviado pela porta eth0. Olhando para a topologia isto signi-



fica que o tráfego na LAN circula entre A1 e RA (IP: 192.168.57.1 e MAC: 00:00:00:aa:00:02) pela porta eth0.

**1.b) Indique, justificando, qual o equipamento da intranet em causa que poderá apresentar a maior tabela ARP em termos de número de entradas.**

O equipamento com mais entradas será o router RA pois tem de associar tráfego para todos os PCs e para a rede entre routers. Confirma-se correndo o comando `arp -a`.

```
root@RA:/tmp/pycore.37653/RA.conf# arp -a
? (192.168.57.21) at 00:00:00:aa:00:04 [ether] on eth1
? (10.0.0.2) at 00:00:00:aa:00:01 [ether] on eth0
```

Figura 2.7: ARP Request

Neste caso apenas tem uma entrada dos pcs (pc A1) da LAN pois ainda só foi feito um ping de A1 para uma sub-rede fora da local, caso o pc A2 ou A3 fosse utilizado para uma tarefa similar, o protocolo ARP seria executado novamente e seria adicionada uma nova entrada à tabela ARP do router.

**2.a) Qual é o valor hexadecimal dos endereços MAC origem e destino? Como interpreta e justifica o endereço destino usado?**

O endereço de origem é o endereço MAC do dispositivo A1 (00:00:00:aa:00:04), o endereço de destino é o Broadcast (00:00:00:00:00:00), o que significa que é enviado por todas as portas do dispositivo. O ARP Request utiliza o Broadcast porque não sabe qual o endereço MAC do destino, logo não sabe porque porta fará a ligação, desse modo envia para todas as portas de forma a garantir que terá uma resposta.

```
Frame 31: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth5.0.86, 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)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
  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: 192.168.57.21
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.57.1
```

Figura 2.8: ARP Request

**2.b) Qual o valor hexadecimal do campo Tipo da trama Ethernet? O que indica?**

O valor do type é 0x0800 (ARP) que indica o próximo encapsulamento protocolar da trama, neste caso protocolo ARP.

**2.c) Observando a mensagem ARP, como pode saber que se trata efetivamente de um pedido ARP? Refira duas formas distintas de obter essa informação.**

Trata-se de um pedido ARP dado o Opcode ser 1 (request) e o target MAC address ser broadcast.

**2.d) Explícite, em linguagem comum, que tipo de pedido ou pergunta é feita pelo host de origem à rede?**

A pergunta feita é um pedido do endereço MAC do dispositivo com o IP no campo *Target IP address*, neste caso o valor é 192.168.57.1 .

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

O valor do campo ARP Opcode é 2 que significa que se trata de um ARP reply.

```
Frame 32: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth5.0.86, id 0
Ethernet II, Src: 00:00:00_aa:00:02 (00:00:00:aa:00:02), 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)
  Source: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  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:00:02 (00:00:00:aa:00:02)
  Sender IP address: 192.168.57.1
  Target MAC address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
  Target IP address: 192.168.57.21
```

Figura 2.9: ARP Reply

**3.b) Em que posição da mensagem ARP está a resposta ao pedido ARP efetuado?**

A resposta está no MAC address do sender, pois é enviado pelo dispositivo que tem a resposta ao ARP request anterior.

**3.c) 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.**

A origem é o router RA e o destino o pc A1. Os comandos executados em A1 confirmam os endereços MAC.

```

root@A1:/tmp/pycore.37653/A1.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.57.21 netmask 255.255.255.128 broadcast 0.0.0.0
    inet6 2001::1:20 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:4 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:00:04 txqueuelen 1000 (Ethernet)
    RX packets 2023 bytes 164438 (164.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26 bytes 2180 (2.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

(a) Ifconfig de A1

```

root@A1:/tmp/pycore.37653/A1.conf# arp -a
? (192.168.57.1) at 00:00:00:aa:00:02 [ether] on eth0

```

(b) Tabela Arp de A1

```

root@A1:/tmp/pycore.37653/A1.conf# netstat -rn
Kernel IP routing table

```

Destination	Gateway	Genmask	Flags	MSS Window	irtt	Iface
0.0.0.0	192.168.57.1	0.0.0.0	UG	0 0	0	eth0
192.168.57.0	0.0.0.0	255.255.255.128	U	0 0	0	eth0

(c) Tabela de Encaminhamento de A1

Figura 2.10: Comandos A1

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

O modo Unicast significa que a mensagem é enviada para um dispositivo em específico logo observa-se a tabela ARP e só se utiliza uma interface, ao contrário do broadcast, que não especifica o dispositivo e envia para todas as portas do dispositivo. O modo unicast é utilizado porque no caso do ARP reply já se tem toda a informação necessária (endereço IP e MAC) sobre o destino.

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

Observando o tráfego no pc A1 não há mais pacotes ARP Request/Reply, isto porque o A1 apenas está ligado ao router RA logo após o primeiro pedido ARP já não mais entradas a preencher.

### 5. 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.

Os campos que definem o tipo e o tamanho dos endereços das camadas de rede e de ligação lógica são descritos, respetivamente, nos campos:

- **Protocol Size (IP address)** : 4 bytes
- **Hardware Size (MAC address)** : 6 bytes

```

Frame 32: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth5.0.86, id 0
Ethernet II, Src: 00:00:00_aa:00:02 (00:00:00:aa:00:02), 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)
  Source: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  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:00:02 (00:00:00:aa:00:02)
  Sender IP address: 192.168.57.1
  Target MAC address: 00:00:00_aa:00:04 (00:00:00:aa:00:04)
  Target IP address: 192.168.57.21

```

Figura 2.11: ARP Reply

6. 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.

O diagrama da figura representa um ping do A1 para B1, logo tem como nós intervenientes os pc A1, o router RA, o router RB e o pc B1.

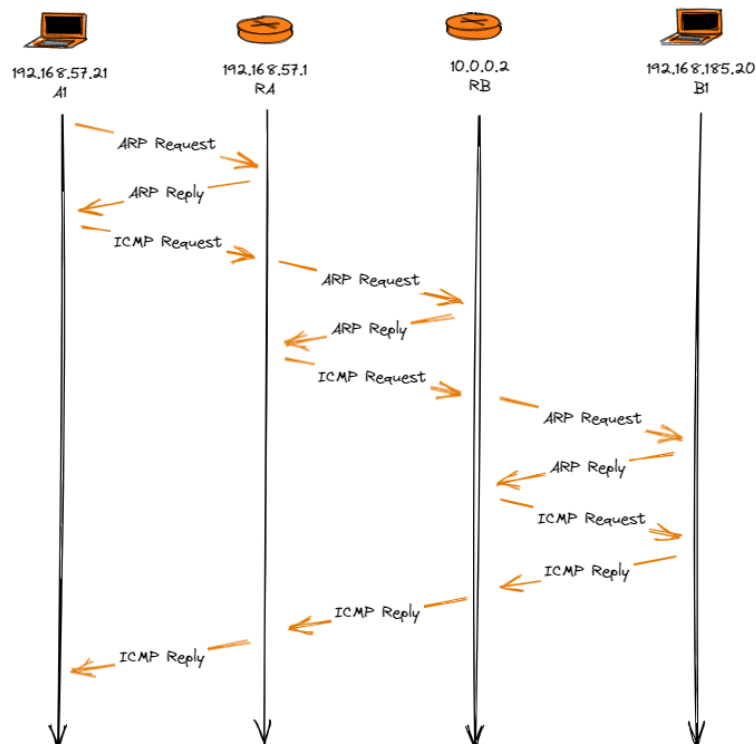
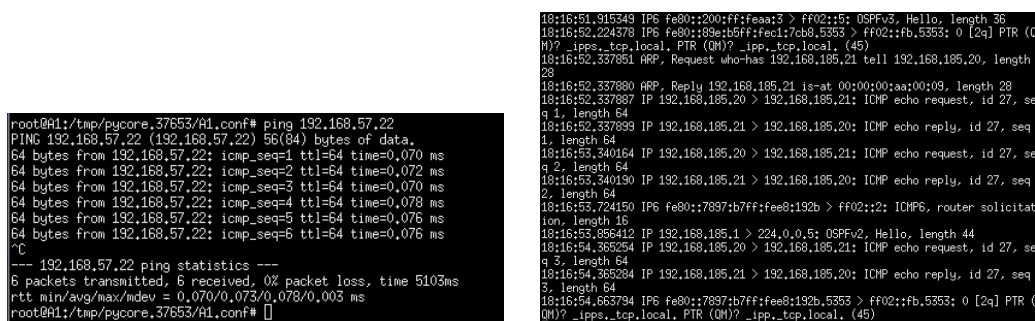


Figura 2.12: Diagrama Cronológico

## 2.0.3 Domínios de colisão

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.

Para testar a LAN do Departamento A faz-se um ping de A1 para A2 com o `tcpdump` em A3. Isto permite testar se o A3 captura tráfego direcionado ao A2.



```
root@A1:/tmp/pycore.37653/A1.conf# ping 192.168.57.22
PING 192.168.57.22 (192.168.57.22) 56(84) bytes of data.
64 bytes from 192.168.57.22: icmp_seq=1 ttl=64 time=0.070 ms
64 bytes from 192.168.57.22: icmp_seq=2 ttl=64 time=0.072 ms
64 bytes from 192.168.57.22: icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from 192.168.57.22: icmp_seq=4 ttl=64 time=0.078 ms
64 bytes from 192.168.57.22: icmp_seq=5 ttl=64 time=0.076 ms
64 bytes from 192.168.57.22: icmp_seq=6 ttl=64 time=0.076 ms
--- 192.168.57.22 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5103ms
rtt min/avg/max/mdev = 0.070/0.073/0.078/0.003 ms
root@A1:/tmp/pycore.37653/A1.conf#
```

```
18:16:51.915349 IP6 fe80::200:ff:feaa:3 > ff02::5: OSPFv3, Hello, length 36
18:16:52.224378 IP6 fe80::89e:b5ff:fecl:7cb8.5353 > ff02::fb.5353: 0 [2q] PTR (Q
M)? _ipps_tcp.local. PTR (QM)? _ipp_tcp.local. (45)
18:16:52.337851 ARP, Request who-has 192.168.185.21 tell 192.168.185.20, length
28
18:16:52.337890 ARP, Reply 192.168.185.21 is-at 00:00:00:aa:00:09, length 28
18:16:52.337887 IP 192.168.185.20 > 192.168.185.21: ICMP echo request, id 27, se
q 1, length 64
18:16:52.337899 IP 192.168.185.21 > 192.168.185.20: ICMP echo reply, id 27, seq
1, length 64
18:16:53.340164 IP 192.168.185.20 > 192.168.185.21: ICMP echo request, id 27, se
q 2, length 64
18:16:53.340190 IP 192.168.185.21 > 192.168.185.20: ICMP echo reply, id 27, seq
2, length 64
18:16:53.724150 IP6 fe80::7897:b7ff:fee8:192b > ff02::2: ICMPv6, router solicitat
ion, length 16
18:16:53.896412 IP 192.168.185.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:16:54.365254 IP 192.168.185.20 > 192.168.185.21: ICMP echo request, id 27, se
q 3, length 64
18:16:54.365284 IP 192.168.185.21 > 192.168.185.20: ICMP echo reply, id 27, seq
3, length 64
18:16:54.663794 IP6 fe80::7897:b7ff:fee8:192b.5353 > ff02::fb.5353: 0 [2q] PTR (
QM)? _ipps_tcp.local. PTR (QM)? _ipp_tcp.local. (45)
```

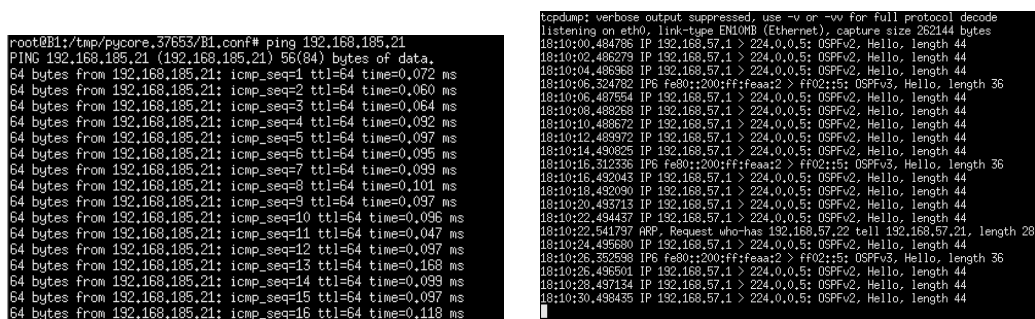
(a) Ping de A1 a A2

(b) TCPDump em A3

Figura 2.13: Análise da LAN do departamento A

Pela imagem (b) verifica-se que o router A3 não recebe qualquer pacote ICMP o que significa que o switch não partilha o tráfego com todos os elementos da rede, daí ser uma LAN comutada. O único pacote que recebe do router A1 é o ARP request e isto é porque o mesmo é emitido da origem em modo Broadcast.

Para testar a LAN do Departamento B faz-se um ping de B1 para B2 com o `tcpdump` em B3. Isto permite testar se o B3 captura tráfego direcionado ao B2.



```
root@B1:/tmp/pycore.37653/B1.conf# ping 192.168.185.21
PING 192.168.185.21 (192.168.185.21) 56(84) bytes of data.
64 bytes from 192.168.185.21: icmp_seq=1 ttl=64 time=0.072 ms
64 bytes from 192.168.185.21: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 192.168.185.21: icmp_seq=3 ttl=64 time=0.064 ms
64 bytes from 192.168.185.21: icmp_seq=4 ttl=64 time=0.092 ms
64 bytes from 192.168.185.21: icmp_seq=5 ttl=64 time=0.097 ms
64 bytes from 192.168.185.21: icmp_seq=6 ttl=64 time=0.095 ms
64 bytes from 192.168.185.21: icmp_seq=7 ttl=64 time=0.099 ms
64 bytes from 192.168.185.21: icmp_seq=8 ttl=64 time=0.101 ms
64 bytes from 192.168.185.21: icmp_seq=9 ttl=64 time=0.097 ms
64 bytes from 192.168.185.21: icmp_seq=10 ttl=64 time=0.096 ms
64 bytes from 192.168.185.21: icmp_seq=11 ttl=64 time=0.047 ms
64 bytes from 192.168.185.21: icmp_seq=12 ttl=64 time=0.037 ms
64 bytes from 192.168.185.21: icmp_seq=13 ttl=64 time=0.169 ms
64 bytes from 192.168.185.21: icmp_seq=14 ttl=64 time=0.099 ms
64 bytes from 192.168.185.21: icmp_seq=15 ttl=64 time=0.097 ms
64 bytes from 192.168.185.21: icmp_seq=16 ttl=64 time=0.116 ms
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:10:00.484785 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:02.486279 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:04.488368 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:06.324782 IP6 fe80::200:ff:feaa:2 > ff02::5: OSPFv3, Hello, length 36
18:10:06.487554 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:08.488268 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:10.488672 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:12.489972 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:14.490825 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:16.312336 IP6 fe80::200:ff:feaa:2 > ff02::5: OSPFv3, Hello, length 36
18:10:16.492043 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:18.492090 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:20.493713 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:22.494457 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:22.541787 ARP, Request who-has 192.168.57.22 tell 192.168.57.21, length 28
18:10:24.495680 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:26.352598 IP6 fe80::200:ff:feaa:2 > ff02::5: OSPFv3, Hello, length 36
18:10:26.496501 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:28.497134 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
18:10:30.498435 IP 192.168.57.1 > 224.0.0.5: OSPFv2, Hello, length 44
```

(a) Ping de B1 a B2

(b) TCPDump em B3

Figura 2.14: Análise da LAN do departamento A

Pela imagem (b) verifica-se que o router B3 recebe todos pacote ICMP Reply/Request o que

significa que o hub partilha o tráfego com todos os elementos da rede, daí ser uma LAN partilhada.

Estas diferenças de tráfego dão devido às características do switch e do hub, isto porque, o switch mantém uma tabela ARP com os endereços MAC que permite a transmissão Unicast, pelo contrário o hub não cria a tabela o que resulta numa transmissão Broadcast.

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

Uma tabela ARP tem os seguintes campos:

- IP nodo adjacente
- Endereço MAC
- Interface
- Tempo de Vida
- Tipo (dynamic/static)

Para criar a tabela do switch A apenas se preencheu os 3 primeiros campos pois são os com relevância para o enunciado.

IP Nodo Adjacente	MAC address	Interface
192.168.57.1	00:00:00:aa:00:02	eth0
192.168.57.21	00:00:00:aa:00:04	eth1
192.168.57.22	00:00:00:aa:00:05	eth2
192.168.57.23	00:00:00:aa:00:06	eth3
192.168.57.30	00:00:00:aa:00:07	eth4

Figura 2.15: Tabela do switch A

## 3 Conclusão

Em conclusão, o estudo dos temas abordados neste trabalho forneceu uma compreensão mais profunda sobre o funcionamento da camada de ligação lógica da pilha protocolar. A análise do formato de um cabeçalho Ethernet, os endereços MAC, o protocolo ARP e domínios de colisão permitiu entender como os dispositivos de rede comunicam uns com os outros e como a informação é transmitida de um ponto 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, e que o endereço MAC é uma identificação única atribuída a cada dispositivo de rede. O protocolo ARP é utilizado para mapear endereços IP em endereços MAC e os domínios de colisão podem afetar a eficiência da rede, especialmente em grandes redes.