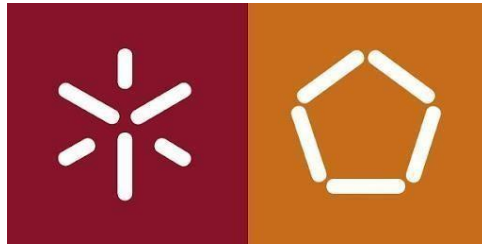


Universidade do Minho
Licenciatura em Engenharia Informática



Redes de Computadores
Trabalho prático 3

Braga, Maio 2023

Trabalho realizado por:

João Ricardo Ribeiro Rodrigues – a100598

Rafael Lima Mesquita - a95097

Sandra Fabiana Pires Cerqueira – a100681

Índice

Trabalho realizado por:	1
1 – Captura e análise de Tramas Ethernet	3
1.1 Exercício 1	3
1.2 Exercício 2	3
1.3 Exercício 3	4
1.4 Exercício 4	6
1.5 Exercício 5	6
1.6 Exercício 6	7
2 – Protocolo ARP	8
2.1 Exercício 1	9
2.1.a Alínea a)	9
2.1.b Alínea b)	10
2.2 Exercício 2	10
2.2.a Alínea a)	10
2.2.b Alínea b)	10
2.2.c Alínea c)	11
2.2.d Alínea d)	11
2.3 Exercício 3	11
2.3.a Alínea a)	12
2.3.b Alínea b)	12
2.3.c Alínea c)	13
2.3.d Alínea d)	13
2.4 Exercício 4	14
2.5 Exercício 5	14
2.6 Exercício 6	15
3- Domínios de colisão	16
3.1 Exercício 1	16
3.2 Exercício 2	18

1 – Captura e análise de Tramas Ethernet

1.1 Exercício 1

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

30	4.483646	172.26.10.125	193.137.9.171	TLSv1.2	791 Application Data
32	4.505431	193.137.9.171	172.26.10.125	TLSv1.2	804 Application Data

Figura 1 - Tramas utilizadas

Ethernet II, Src: CloudNet_12:1f:29 (f8:89:d2:12:1f:29), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
> Destination: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
> Source: CloudNet_12:1f:29 (f8:89:d2:12:1f:29)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 172.26.10.125, Dst: 193.137.9.171

Figura 2 - Endereços de origem e destino da trama Ethernet

Tendo em conta a trama capturada, temos que o endereço MAC de origem é **f8:89:d2:12:1f:29** e o endereço MAC de destino é **00:d0:03:ff:94:00**, a *source* refere-se ao nosso computador e o destino será o próximo *router* a que a máquina se encontra conectada, visto que este será o destino do próximo salto do pacote.

1.2 Exercício 2

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

O valor hexadecimal do campo *type* é **0x0800**, tal como se podemos observar na Fig.3, e representa o protocolo de camada superior utilizado, neste caso IPv4.

Ethernet II, Src: CloudNet_12:1f:29 (f8:89:d2:12:1f:29), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
> Destination: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
> Source: CloudNet_12:1f:29 (f8:89:d2:12:1f:29)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 172.26.10.125, Dst: 193.137.9.171

Figura 3 – Informação acerca da trama Ethernet onde podemos ver o valor hexadecimal

1.3 Exercício 3

Questão: 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.

São usados 54 bytes no encapsulamento protocolar, sendo 14 bytes correspondentes ao header Ethernet (Fig.4), 20 bytes ao IP (Fig.5) e 20 bytes ao TCP (Fig.6). Desta forma, tendo em conta que o tamanho total da trama é 791 bytes, podemos calcular a sobrecarga introduzida pela pilha protocolar da seguinte forma:

$$\text{overhead} = \frac{54}{791} \times 100 = 6.83\%$$

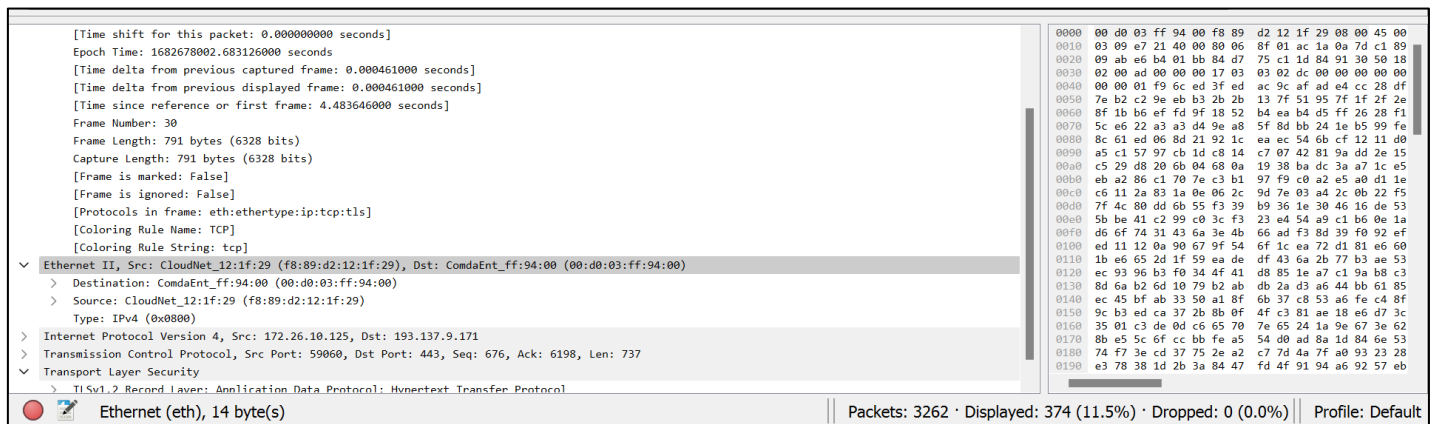


Figura 4 - Bytes Ethernet

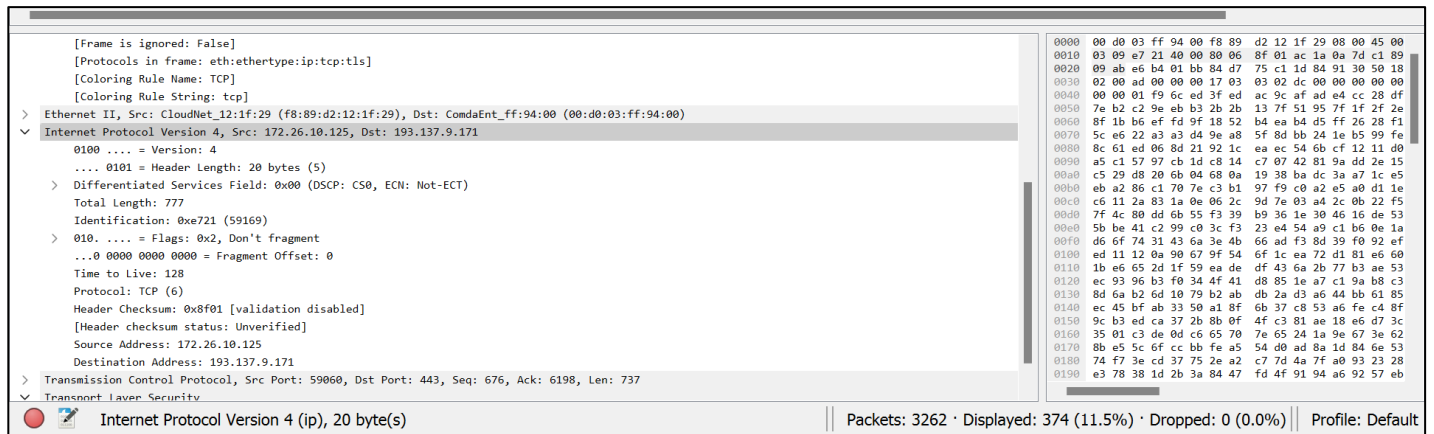
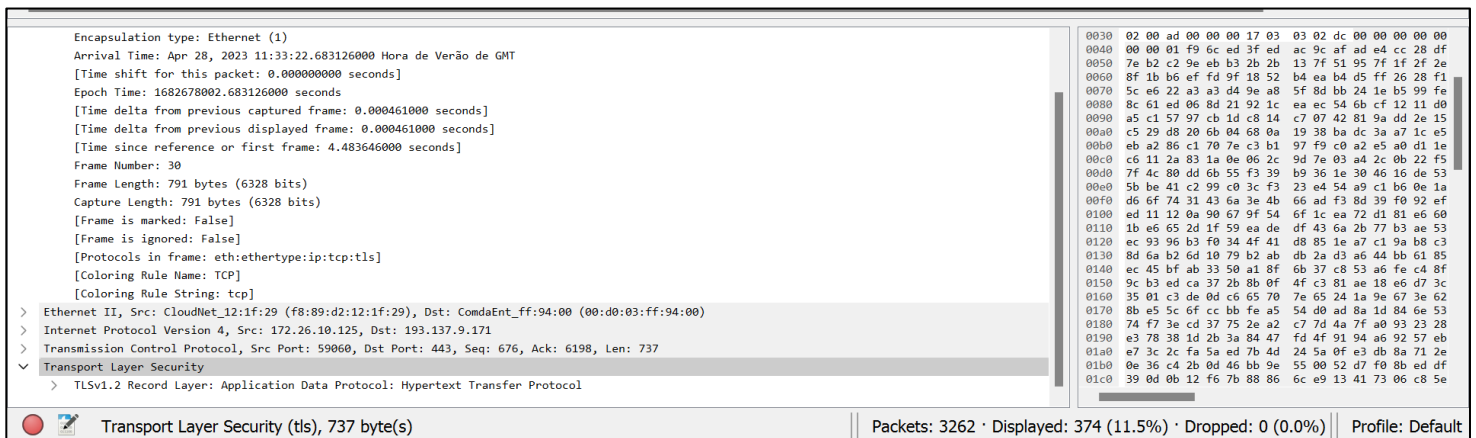
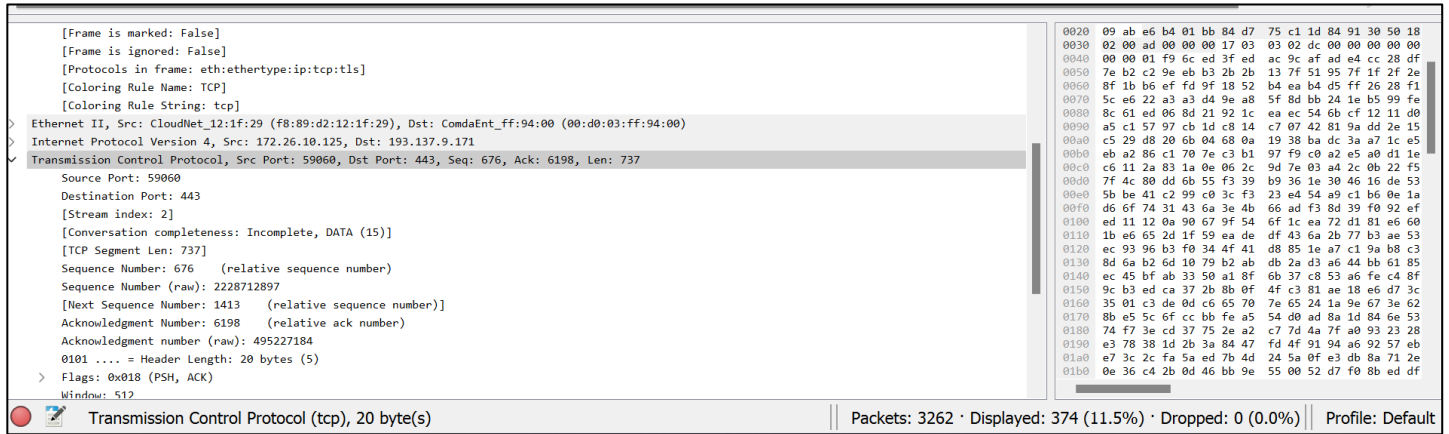


Figura 5 - Bytes IPv4



A seguir resposta às seguintes perguntas, baseado no conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP proveniente do servidor

1.4 Exercício 4

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

O endereço *Ethernet* da fonte é **00:d0:03:ff:94:00** e corresponde ao *router* ao qual a máquina se encontra conectada, pois, a resposta ao pedido anterior terá de voltar pelo *router* que permite à máquina aceder à rede exterior.

No.	Time	Source	Destination	Protocol	Length	Info
30	4.483646	172.26.10.125	193.137.9.171	TLSv1.2	791	Application Data
32	4.505431	193.137.9.171	172.26.10.125	TLSv1.2	804	Application Data
33	4.523101	172.26.10.125	193.137.9.171	TLSv1.2	815	Application Data
42	4.653216	172.26.10.125	13.42.111.94	TLSv1.2	591	Client Hello
44	4.758383	13.42.111.94	172.26.10.125	TLSv1.2	1304	Server Hello
47	4.758383	13.42.111.94	172.26.10.125	TLSv1.2	972	Certificate, Server Key Exchange, Server Hello Done
49	4.759649	172.26.10.125	13.42.111.94	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
50	4.759916	172.26.10.125	13.42.111.94	TLSv1.2	1160	Application Data

Arrival Time: Apr 28, 2023 11:33:22.704911000 Hora de Verão de GMT	0000 f8 89 d2 12 1f 29 00 03 ff 94 00 08 00 45 00
[Time shift for this packet: 0.000000000 seconds]	0010 03 16 03 28 40 00 fc 06 f6 ed c1 89 09 ab ac 1a
Epoch Time: 1682678802.704911000 seconds	0020 0a 7d 01 bb e6 b4 1d 84 91 30 84 d7 78 a2 50 18
[Time delta from previous captured frame: 0.018430000 seconds]	0030 0d 96 4c 81 00 00 17 03 03 02 e9 e0 d7 85 17 79
[Time delta from previous displayed frame: 0.021785000 seconds]	0040 79 45 56 9c 95 87 ef 00 dd e9 e9 ae 60 1f 84 9c
[Time since reference or first frame: 4.505431000 seconds]	0050 13 68 5f 75 bc cd cc aa 17 10 38 79 d2 6a 51 73
Frame Number: 32	0060 55 a9 8d 36 9a 7c fe 32 07 e0 c7 40 1b a9 87 a9
Frame Length: 804 bytes (6432 bits)	0070 75 22 8f fa 95 b0 cb f2 6b 47 50 a9 0a bb 1b f2
Capture Length: 804 bytes (6432 bits)	0080 a2 7f a3 fe e2 34 d7 bd 1d 62 66 8f db 64 57 4f
[Frame is marked: False]	0090 f9 59 d7 40 fc 70 85 1e 0f 26 ec b7 67 63 83 bc
[Frame is ignored: False]	00a0 3c 20 af 96 73 ee 8c 33 93 9b db 4a 2f 49 70 be
[Protocols in frame: eth:ethertype:ip:tcp:tls]	00b0 a4 62 5a 17 08 76 25 51 4a 84 7e 6a f1 e9 7d 90
[Coloring Rule Name: TCP]	00c0 93 74 f5 d2 b2 50 b6 9a b2 e9 0b 18 31 6b f8 95
[Coloring Rule String: tcp]	00d0 a1 32 91 6e c5 b4 1d de fe 7d 65 88 a3 f0 12 bc
	00e0 7f 40 8c 0e c3 0b 12 c9 ff a9 63 ca 4e 8a b2 d9
	00f0 32 ce e2 6c 3c d4 95 aa 72 72 61 c3 5e 63 cc f9
	0100 25 21 c8 eb 42 99 19 3e a8 f3 14 93 1f 65 af 54
	0110 43 37 43 0f 0d 15 9f 0f f5 8a 58 1a 82 0a 6a 4b
	0120 72 e4 c9 7f d6 e2 27 cc dd 91 e0 51 d1 6b ca 20
	0130 67 6b 73 12 34 7a d8 26 33 e9 1f b5 0f fe bb 87
	0140 23 63 8c ed de 3d e5 d9 aa fe 01 f7 49 6c 1e af
	0150 6d 8b bc be d9 d1 2f cb 29 74 d2 8f 36 bf 43 a1
	0160 74 ee df 4f 7a 58 5c c7 69 f3 36 38 26 72 1d e5
	0170 5a 0b 82 5d 84 26 bd 1b d0 56 29 eb 44 4b c1 d8
	0180 f0 13 98 ba 17 50 70 7d 1d 6a 41 e6 c3 9b e5 8d
	0190 3b 93 55 62 b0 f7 6f 6b 87 64 5c 34 5b e3 42 32

Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00), Dst: CloudNet_12:1f:29 (f8:89:d2:12:1f:29)	
> Destination: CloudNet_12:1f:29 (f8:89:d2:12:1f:29)	
> Source: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)	
> Type: IPv4 (0x0800)	
> Internet Protocol Version 4, Src: 193.137.9.171, Dst: 172.26.10.125	
> Transmission Control Protocol, Src Port: 443, Dst Port: 59060, Seq: 6198, Ack: 1413, Len: 750	
> Transport Layer Security	

Ethernet (eth), 14 byte(s) | Packets: 3262 · Displayed: 374 (11.5%) · Dropped: 0 (0.0%) | Profile: Default

Figura 8 - Conteúdo da trama Ethernet quee contém o primeiro byte da resposta HTTP

1.5 Exercício 5

Questão: Qual é o endereço MAC do destino? A que sistema (host) corresponde?

Como se pode ver na Fig.8, o endereço MAC do destino é **f8:89:d2:12:1f:29**, e este corresponde à máquina local utilizada para efetuar o pedido de acesso ao website pretendido.

1.6 Exercício 6

Questão: *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 protocolos contidos na trama obtida, tal como se pode verificar na Fig.9, são o Ethernet, o IPv4 (Internet Protocol Version 4), o TCP (Transmission Control Protocol) e o TLS (Transport Layer Security).

tls						
No.	Time	Source	Destination	Protocol	Length	Info
30	4.483646	172.26.10.125	193.137.9.171	TLSv1.2	791	Application Data
32	4.505431	193.137.9.171	172.26.10.125	TLSv1.2	804	Application Data
33	4.523101	172.26.10.125	193.137.9.171	TLSv1.2	815	Application Data
42	4.653216	172.26.10.125	13.42.111.94	TLSv1.2	591	Client Hello
44	4.758383	13.42.111.94	172.26.10.125	TLSv1.2	1304	Server Hello
47	4.758383	13.42.111.94	172.26.10.125	TLSv1.2	972	Certificate, Server Key Exchange
49	4.759649	172.26.10.125	13.42.111.94	TLSv1.2	180	Client Key Exchange, Change Cipher
50	4.759916	172.26.10.125	13.42.111.94	TLSv1.2	1160	Application Data
51	4.857310	193.137.9.171	172.26.10.125	TLSv1.2	850	Application Data
> Frame 32: 804 bytes on wire (6432 bits), 804 bytes captured (6432 bits) on interface \Device\NPF_{88EE451C-BD32-4735-B321-CC2BB959A39F}, id 0 > Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00), Dst: CloudNet_12:1f:29 (f8:89:d2:12:1f:29) > Internet Protocol Version 4, Src: 193.137.9.171, Dst: 172.26.10.125 > Transmission Control Protocol, Src Port: 443, Dst Port: 59060, Seq: 6198, Ack: 1413, Len: 750 > Transport Layer Security						

Figura 9- Protocolos contidos na trama obtida

2 – Protocolo ARP

Seguindo as instruções do enunciado adotando a terminologia usada no *CORE*, considerando que o departamento A contém três *PCs* e um *host* (servidor) ligados a um *switch*, que por sua vez liga ao router RA. O departamento B tem três *PCs* ligados a um *hub*, que por sua vez liga ao router RB. Os dois routers estão ligados entre si por uma ligação física, cujo endereço de rede é atribuído automaticamente pelo *CORE*. Todos os links têm uma largura de banda de 200 Mbps. Construímos a seguinte topologia core:

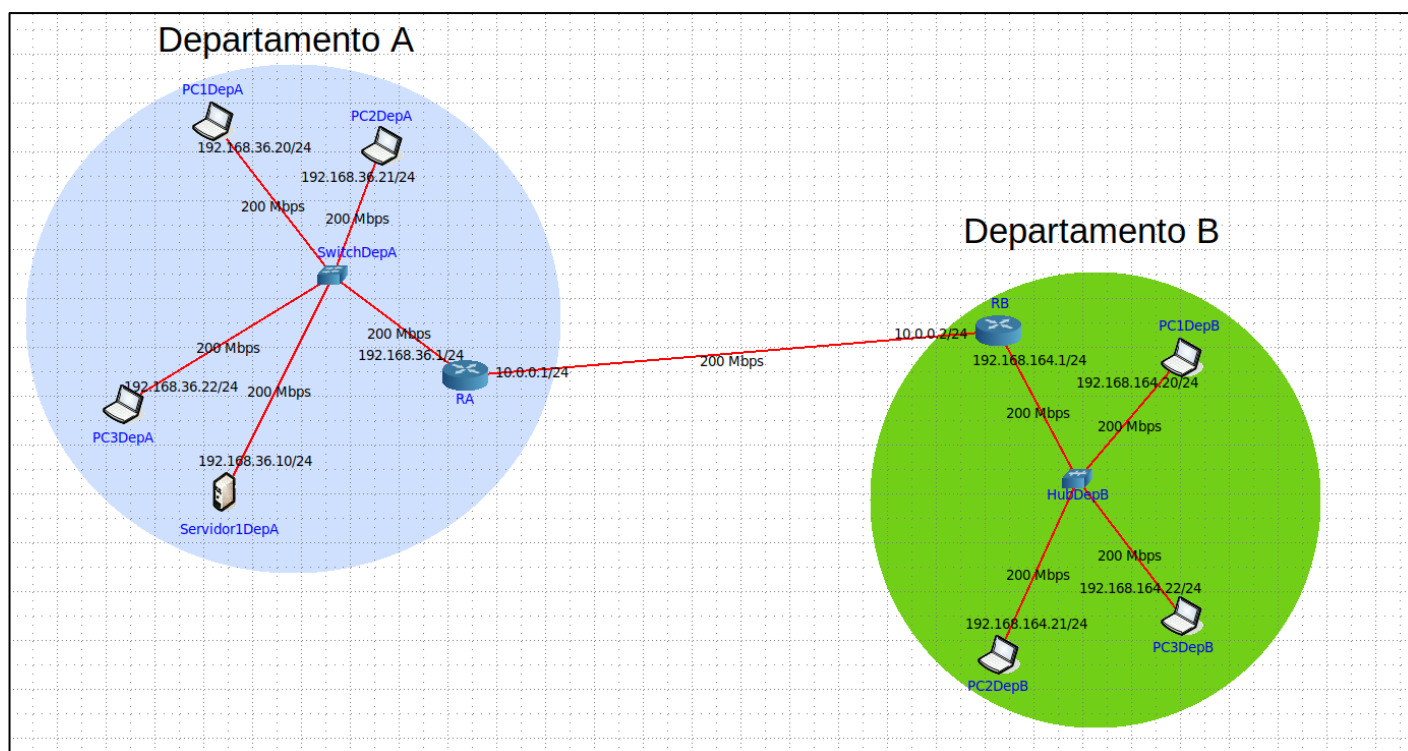


Figura 10- Topologia Core pedida

2.1 Exercício 1

Abra uma consola no PC onde efetuou o ping. Observe o conteúdo da tabela ARP com o comando “arp -a”

Departamento A: 192.168.36.X/25

Departamento B: 192.168.164.X/25

```

vcmd
root@PC1DepA:/tmp/pycore.45303/PC1DepA.conf# ping 192.168.164.20
PING 192.168.164.20 (192.168.164.20) 56(84) bytes of data.
64 bytes from 192.168.164.20: icmp_seq=1 ttl=62 time=2.94 ms
64 bytes from 192.168.164.20: icmp_seq=2 ttl=62 time=1.60 ms
64 bytes from 192.168.164.20: icmp_seq=3 ttl=62 time=1.58 ms
64 bytes from 192.168.164.20: icmp_seq=4 ttl=62 time=1.24 ms
64 bytes from 192.168.164.20: icmp_seq=5 ttl=62 time=1.23 ms
64 bytes from 192.168.164.20: icmp_seq=6 ttl=62 time=1.39 ms
64 bytes from 192.168.164.20: icmp_seq=7 ttl=62 time=1.22 ms
^C
--- 192.168.164.20 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6015ms
rtt min/avg/max/mdev = 1.222/1.598/2.938/0.566 ms
root@PC1DepA:/tmp/pycore.45303/PC1DepA.conf# ping 192.168.164.21
PING 192.168.164.21 (192.168.164.21) 56(84) bytes of data.
64 bytes from 192.168.164.21: icmp_seq=1 ttl=62 time=2.58 ms
64 bytes from 192.168.164.21: icmp_seq=2 ttl=62 time=8.66 ms
64 bytes from 192.168.164.21: icmp_seq=3 ttl=62 time=19.7 ms
64 bytes from 192.168.164.21: icmp_seq=4 ttl=62 time=34.8 ms
64 bytes from 192.168.164.21: icmp_seq=5 ttl=62 time=45.7 ms
64 bytes from 192.168.164.21: icmp_seq=6 ttl=62 time=4.91 ms
64 bytes from 192.168.164.21: icmp_seq=7 ttl=62 time=11.3 ms
64 bytes from 192.168.164.21: icmp_seq=8 ttl=62 time=9.53 ms
64 bytes from 192.168.164.21: icmp_seq=9 ttl=62 time=9.86 ms
^C
--- 192.168.164.21 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8056ms
rtt min/avg/max/mdev = 2.577/16.333/45.719/13.775 ms
root@PC1DepA:/tmp/pycore.45303/PC1DepA.conf#

```

Figura 11

No.	Time	Source	Destination	Protocol	Length	Info
13	10.145647593	00:00:00_aa:00:00	Broadcast	ARP		42 Who has 192.168.36.1? Tell 192.168.36.20
14	10.148201665	00:00:00_aa:00:04	00:00:00_aa:00:00	ARP		42 192.168.36.1 is at 00:00:00:aa:00:04
28	15.196858816	00:00:00_aa:00:04	00:00:00_aa:00:00	ARP		42 Who has 192.168.36.20? Tell 192.168.36.1
29	15.196871493	00:00:00_aa:00:00	00:00:00_aa:00:04	ARP		42 192.168.36.20 is at 00:00:00:aa:00:00

Figura 12

2.1.a Alínea a)

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

Utilizando **arp -a**:

```

root@PC1DepA:/tmp/pycore.45303/PC1DepA.conf# arp -a
? (192.168.36.1) at 00:00:00:aa:00:00 [ether] on eth0

```

Figura 13- Comando arp -a.

Utilizando **arp**:

```

re.45303/PC1DepA.conf# arp
Address          HWtype  HWaddress      Flags Mask    Iface
192.168.36.1     ether   00:00:00:aa:00:00  C             eth0

```

Figura 14- Tabela ARP do nosso computador.

Recorrendo ao manual *ARP (man arp)*, foi-nos possível interpretar o significado de cada uma das colunas da tabela *ARP*. Concluímos que, a **coluna Address** corresponde aos endereços (*host*), neste caso temos o *gateway* da rede local; a **coluna HWtype** fornece-nos o protocolo da camada física utilizado; a **coluna HWaddress** diz-nos o endereço MAC (neste caso endereço *Ethernet*, visto que, o protocolo da camada física é do tipo *Ethernet*); a **coluna Flags** mostra-nos o tipo de registo que está a ser introduzido em memória (neste caso o valor é C, o que significa que este registo foi obtido dinamicamente pelo protocolo *ARP* e não introduzido manualmente); a **coluna Mask** diz-nos a máscara da sub-rede utilizada e a **coluna Iface** diz-nos a *interface* de rede, neste caso *eth0*.

2.1.b Alínea b)

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

O equipamento da intranet que poderá apresentar a maior tabela *ARP* será o **router R_A**, pois é o equipamento com mais ligações adjacentes a outros equipamentos, e o objetivo do protocolo *ARP* é permitir fazer um mapeamento entre endereços do nível de rede e endereços do nível de ligação lógica, de forma a possibilitar a entrega de dados entre nós adjacentes.

2.2 Exercício 2

Observe a trama *Ethernet* que contém a mensagem com o pedido *ARP* (*ARP Request*).

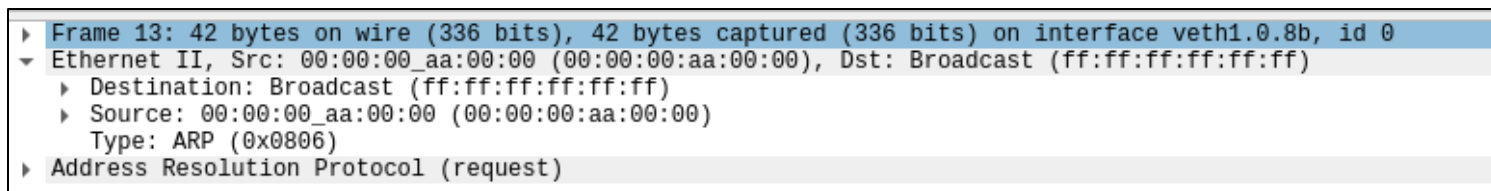


Figura 15- Trama *Ethernet* que contém a mensagem com pedido *ARP*.

2.2.a Alínea a)

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

Tal como se pode ver na Fig.15, o valor em hexadecimal do **endereço de destino** da trama *Ethernet* é **ff:ff:ff:ff:ff:ff** (Broadcast) e o do **endereço de origem** é **00:00:00:aa:00:00**.

O endereço destino é o de *Broadcast* pois, a máquina que envia o *ARP request* necessita de saber qual o endereço MAC destino, logo envia uma mensagem para o endereço *Broadcast* (o que corresponde a enviar para todas as interfaces adjacentes) e espera uma resposta da máquina destino com o seu endereço MAC e assim que a receber adiciona o seu valor à tabela *ARP*.

2.2.b Alínea b)

Questão: Qual o valor hexadecimal do campo Tipo da trama *Ethernet*? O que indica?

O valor em hexadecimal do campo Tipo da trama *Ethernet* é **0x0806** e indica que se trata do protocolo *ARP*.

2.2.c Alínea c)

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

```

▶ Frame 13: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth1.0.8b, id 0
▶ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), 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:00:00 (00:00:00:aa:00:00)
  Sender IP address: 192.168.36.20
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.36.1

```

Figura 16- Pedido ARP.

O campo **opcode** apresenta o valor 1, logo indica que se trata, efetivamente, de um pedido ARP (ARP request). Para além disso, é possível também identificar o endereço destino como sendo um *Broadcast* e, assim como referido na alínea 2.2.a), *Broadcast* é o envio de uma mensagem a todas as interfaces adjacentes com o intuito de pedir (request) ao endereço destino, o que nos permite identificar que a mensagem é um pedido ARP (ARP request).

2.2.d Alínea d)

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

```

Info
Who has 192.168.36.1? Tell 192.168.36.20

```

Figura 17- Pergunta feita pelo host de origem.

De uma forma sucinta, através da pergunta “Who has 192.168.36.1? Tell 192.168.36.20” (Fig.17) o *host* de origem pretende saber que interface tem o endereço IP 192.168.36.1 e então pergunta a todos os *hosts* para saber qual deles tem esse endereço e pede para enviar a resposta para o endereço IP 192.168.36.20

2.3 Exercício 3

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

No.	Time	Source	Destination	Protocol	Length	Info
13	10.145647593	00:00:00_aa:00:00	Broadcast	ARP	42	Who has 192.168.36.1? Tell 192.168.36.20
14	10.148201665	00:00:00_aa:00:04	00:00:00_aa:00:00	ARP	42	192.168.36.1 is at 00:00:00:aa:00:04
28	15.196858816	00:00:00_aa:00:04	00:00:00_aa:00:00	ARP	42	Who has 192.168.36.20? Tell 192.168.36.1
29	15.196871493	00:00:00_aa:00:00	00:00:00_aa:00:04	ARP	42	192.168.36.20 is at 00:00:00:aa:00:00

Figura 18- ARP reply.

2.3.a Alínea a)

Questão: Qual o valor do campo ARP opcode? O que especifica?

```

▶ Frame 14: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth1.0.8b, id 0
▶ Ethernet II, Src: 00:00:00_aa:00:04 (00:00:00:aa:00:04), Dst: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
▼ 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:00:04 (00:00:00:aa:00:04)
  Sender IP address: 192.168.36.1
  Target MAC address: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
  Target IP address: 192.168.36.20
    
```

Figura 19- ARP reply.

O valor do campo *opcode* é **2** e representa “reply”, o que especifica que se trata de uma mensagem *ARP reply*. Assim, como foi referido acima, o código 1 no campo *opcode* significava que a mensagem era do tipo *ARP request*, o código 2 significa que a mensagem é do tipo *ARP reply*.

2.3.b Alínea b)

Questão: Em que posição da mensagem ARP está a resposta ao pedido ARP efetuado?

```
42 192.168.36.1 is at 00:00:00:aa:00:04
```

Figura 20

```

▶ Frame 14: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth1.0.8b, id 0
▶ Ethernet II, Src: 00:00:00_aa:00:04 (00:00:00:aa:00:04), Dst: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
▼ 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:00:04 (00:00:00:aa:00:04)
  Sender IP address: 192.168.36.1
  Target MAC address: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
  Target IP address: 192.168.36.20
    
```

Figura 21

A resposta ao pedido ARP efetuado está no campo “**Sender MAC address**” como podemos constatar pelas Fig.20 e Fig.21.

2.3.c Alínea c)

Questão: *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.*

```

rtt min/avg/max/mdev = 2,862/10,588/19,579/5,754 ms
root@PC1DepA:/tmp/pycore.44071/PC1DepA.conf# arp -a
? (192.168.36.1) at 00:00:00:aa:00:04 [ether] on eth0
root@PC1DepA:/tmp/pycore.44071/PC1DepA.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.36.20 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 fe80::200:ff:feaa:0 prefixlen 64 scopeid 0x20<link>
    inet6 2001::20 prefixlen 64 scopeid 0x0<global>
    ether 00:00:00:aa:00:00 txqueuelen 1000 (Ethernet)
    RX packets 446 bytes 38428 (38.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26 bytes 2236 (2.2 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 4 bytes 340 (340.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 340 (340.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@PC1DepA:/tmp/pycore.44071/PC1DepA.conf#

```

Figura 22

Podemos identificar, como sistema correspondente ao endereço MAC de **origem**, o **router R_A**, pois através do comando **arp** identificamos, facilmente, a correspondência entre o seu endereço IP e o seu endereço MAC e conseguimos identificar na topologia representada na figura do Exercício 2 a que sistema pertence.

Podemos ainda identificar também, como sistema correspondente ao endereço MAC de **destino**, o **Host “PC1DepA”**, através do comando `ifconfig` que mostra tanto o seu endereço IP como o seu endereço MAC.

2.3.d Alínea d)

Questão: *Justifique o modo de comunicação (unicast vs. broadcast) usado no envio da resposta ARP (ARP Reply).*

O protocolo *Address Resolution Protocol* (ARP) é utilizado para mapeamento de endereços IP para endereços físicos de rede (MAC). Quando um dispositivo necessita de enviar um pacote para outro dispositivo da sua rede, ele pode usar o ARP para descobrir qual é o endereço MAC do dispositivo de destino.

Quando um dispositivo emite uma solicitação ARP (*ARP Request*), ele envia uma mensagem de *broadcast* para todos os dispositivos da sua rede local, perguntando "quem possui o endereço IP X?". Todos os dispositivos na rede receberão então esta mensagem e o dispositivo a que corresponder o endereço IP especificado enviará então uma resposta ARP (*ARP Reply*) contendo seu endereço MAC para o dispositivo solicitante.

A resposta ARP é enviada em *unicast*, ou seja, é direcionada especificamente ao dispositivo que fez a solicitação ARP original. Isto acontece porque o dispositivo solicitante precisa do endereço MAC específico do dispositivo de destino para enviar o pacote de dados. Enviar a resposta em *broadcast* para todos os dispositivos na rede seria ineficiente e poderia causar tráfego desnecessário na rede. Portanto, a resposta ARP é sempre enviada em *unicast* para minimizar o tráfego na rede e garantir a entrega eficiente de pacotes entre os diferentes dispositivos.

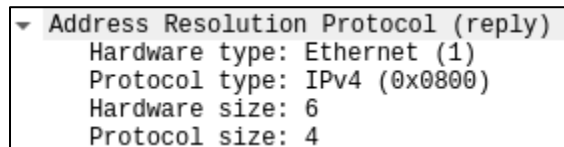
2.4 Exercício 4

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

O ping feito ao segundo PC não originou pacotes ARP, uma vez que, quando foi executado o ping pela primeira vez, ele conseguiu o endereço MAC das interfaces que necessitava para o segundo ping e não precisa de um ARP Request ou ARP Reply novamente.

2.5 Exercício 5

Questão: 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.



```
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
```

Figura 23- Mensagem ARP.

Os campos que permitem definir o tipo dos endereços das camadas de rede e de ligação lógica são: o **“Hardware Type”**, que define o tipo de tecnologia de acesso à rede, neste caso tem o valor 1 pois representa Ethernet pois estamos a trabalhar com endereços Ethernet (O Wireshark assume sempre um ambiente Ethernet); o **“Protocol Type”** que define o protocolo de rede usado pelos endereços IP, e neste caso tem o valor 0x0800 pois representa o IPv4.

Por sua vez, os campos que permitem definir o tamanho dos endereços das camadas de rede e de ligação lógica são: o **“Hardware size”**, que neste caso tem o valor 6 pois um endereço Ethernet usa 6 bytes, e o **“Protocol size”**, que neste caso tem o valor 4 pois os endereços IPv4 usam 4 bytes.

2.6 Exercício 6

Questão: 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 receção da resposta ICMP do sistema destino (represente apenas os nós intervenientes). Assuma que todas as tabelas ARP se encontram inicialmente vazias.

Na figura 24 encontra-se o diagrama temporal ilustrativo de troca de mensagens ARP e ICMP.

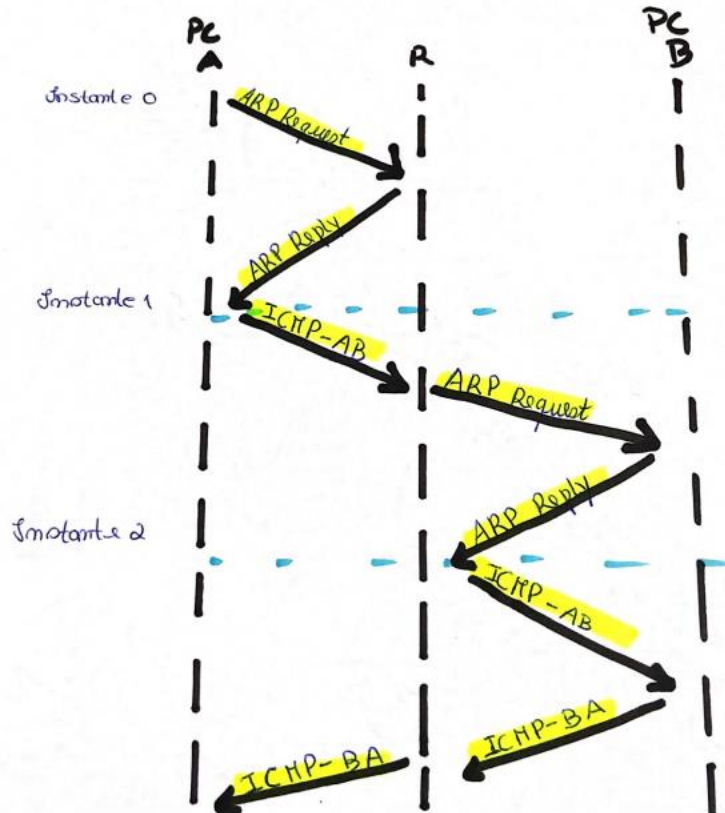


Figura 24- Esquema Temporal

Podemos ainda analisar, mais aprofundadamente, o conteúdo das mensagens ICMP. As mensagens ICMP-AB representam o *Ping Request* efetuado de A para B. Por sua vez, as mensagens ICMP-BA representam o *Ping Reply* a esse pedido. As mensagens ARP Request são enviadas sempre que a máquina de origem não conhece o endereço MAC da interface da máquina de destino e esta responde com uma mensagem ARP Reply, logo a máquina A envia mensagens ARP Request para o Router e para o destino B uma única vez antes de poder enviar o tráfego ICMP. O Router e a Máquina B respondem com ARP Reply e Ping Reply sempre que recebem o ARP Request e ARP Reply da máquina A.

3- Domínios de colisão

Considere a topologia de rede definida anteriormente.

3.1 Exercício 1

Questão: 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

De forma a efetuarmos os testes pretendidos, executamos o comando **tcpdump** em 2 máquinas de cada departamento e executamos o comando **ping** de outra máquina também de cada departamento para uma das máquinas com o comando **tcpdump** no respetivo departamento.

Podemos observar abaixo os resultados obtidos:

Departamento A (SWITCH):

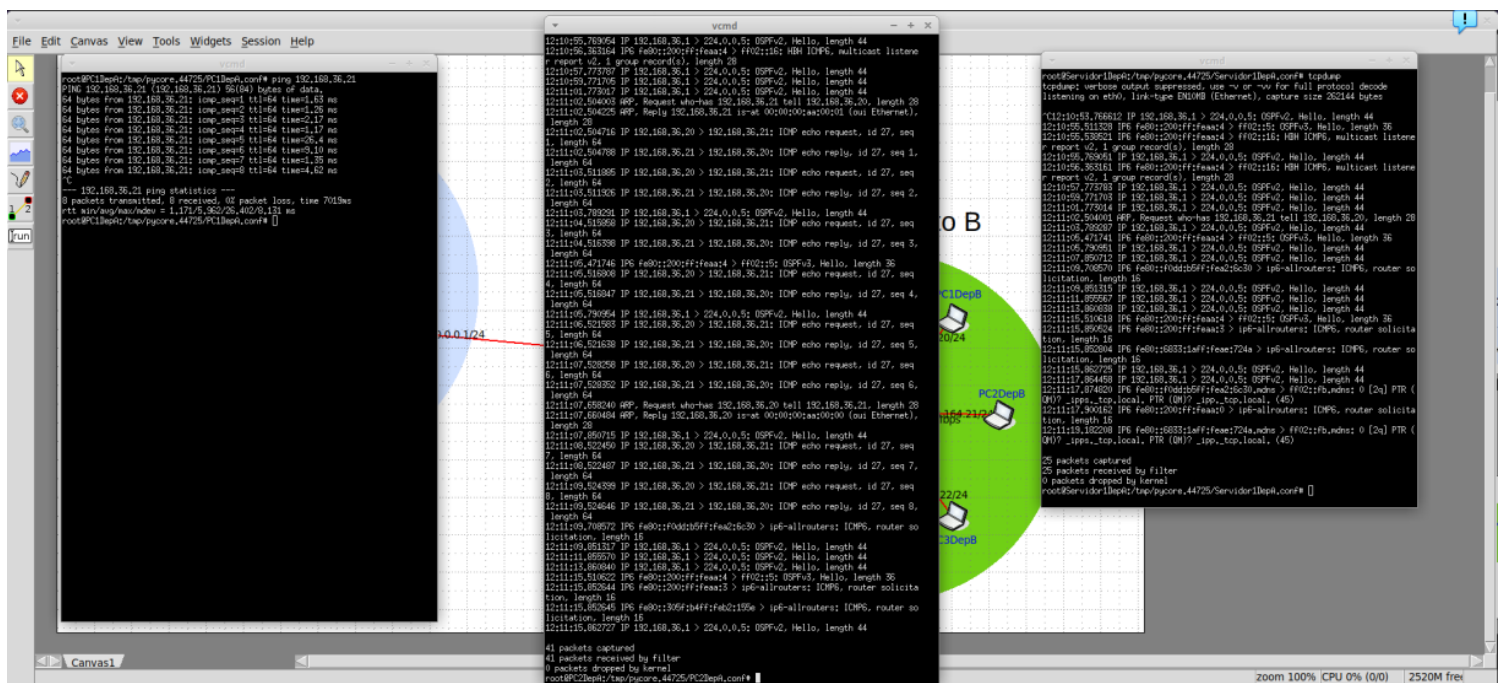


Figura 25 – Análise do tráfego no Departamento A

Departamento B (HUB):

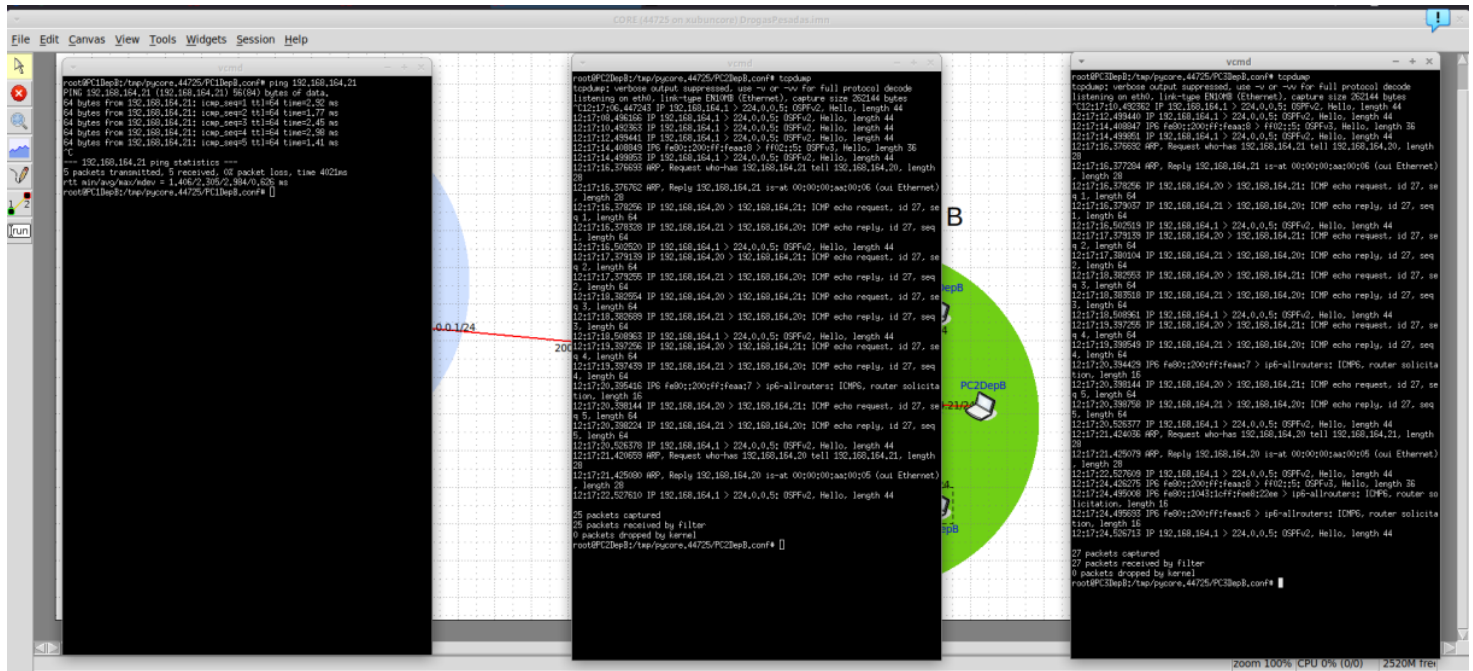


Figura 26 – Análise do tráfego no Departamento B

Tal como esperado podemos verificar diferença no tráfego do Departamento A e do Departamento B, tornando-se evidente nas mensagens ICMP.

No caso da LAN comutada (SWITCH) podemos verificar que o Servidor1DepA não recebe qualquer mensagem relativa à comunicação entre o PC1DepA e o PC2DepA, enquanto na LAN partilhada (HUB), apesar do PC3DepB não fazer parte da comunicação de PC1DepB e PC2DepB também recebe as mensagens ICMP destinadas ao PC2DepB.

Assim, podemos facilmente identificar as diferenças entre uma LAN comutada e uma LAN partilhada, podendo ainda identificar as diferenças de um hub e de um switch.

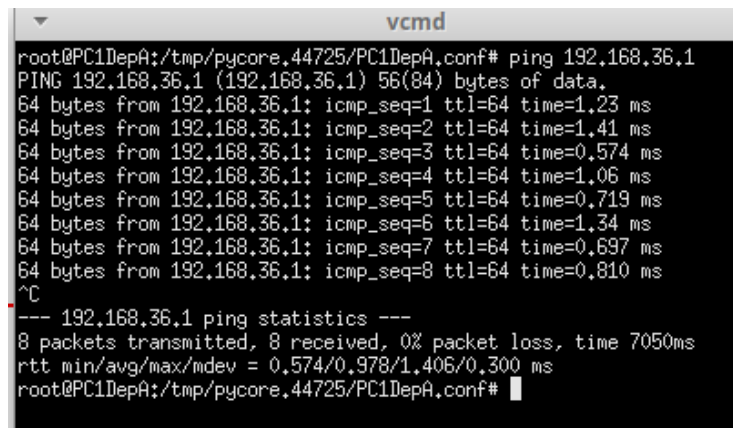
Um hub é um dispositivo que repete o sinal que chega através de uma porta de entrada para todas as outras portas, ou seja, difunde o sinal por todas as interfaces a que está conectado.

Um switch comuta as tramas que recebe para interface destino apropriada apoiando-se na sua tabela de switching, transmitindo apenas para todas as interfaces no caso de não possuir o endereço pretendido.

3.2 Exercício 2

Questão: *Construa manualmente a tabela de comutação do switch do Departamento A, atribuindo números de porta à sua escolha.*

De maneira a ser possível a construção da tabela de comutação do *switch* do Departamento A é preciso, primeiramente identificar os endereços *MAC* dos dispositivos conectados ao *switch*.

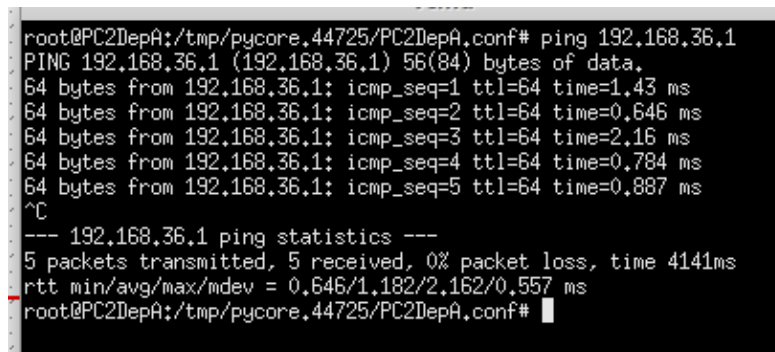


```

vcmd
root@PC1DepA:/tmp/pycore.44725/PC1DepA.conf# ping 192.168.36.1
PING 192.168.36.1 (192.168.36.1) 56(84) bytes of data.
64 bytes from 192.168.36.1: icmp_seq=1 ttl=64 time=1.23 ms
64 bytes from 192.168.36.1: icmp_seq=2 ttl=64 time=1.41 ms
64 bytes from 192.168.36.1: icmp_seq=3 ttl=64 time=0.574 ms
64 bytes from 192.168.36.1: icmp_seq=4 ttl=64 time=1.06 ms
64 bytes from 192.168.36.1: icmp_seq=5 ttl=64 time=0.719 ms
64 bytes from 192.168.36.1: icmp_seq=6 ttl=64 time=1.34 ms
64 bytes from 192.168.36.1: icmp_seq=7 ttl=64 time=0.637 ms
64 bytes from 192.168.36.1: icmp_seq=8 ttl=64 time=0.810 ms
^C
--- 192.168.36.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7050ms
rtt min/avg/max/mdev = 0.574/0.978/1.406/0.300 ms
root@PC1DepA:/tmp/pycore.44725/PC1DepA.conf#

```

Figura 27 - Resultado comando ping PC1DepA



```

root@PC2DepA:/tmp/pycore.44725/PC2DepA.conf# ping 192.168.36.1
PING 192.168.36.1 (192.168.36.1) 56(84) bytes of data.
64 bytes from 192.168.36.1: icmp_seq=1 ttl=64 time=1.43 ms
64 bytes from 192.168.36.1: icmp_seq=2 ttl=64 time=0.646 ms
64 bytes from 192.168.36.1: icmp_seq=3 ttl=64 time=2.16 ms
64 bytes from 192.168.36.1: icmp_seq=4 ttl=64 time=0.784 ms
64 bytes from 192.168.36.1: icmp_seq=5 ttl=64 time=0.887 ms
^C
--- 192.168.36.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4141ms
rtt min/avg/max/mdev = 0.646/1.182/2.162/0.557 ms
root@PC2DepA:/tmp/pycore.44725/PC2DepA.conf#

```

Figura 28 - Resultado comando ping PC2DepA

```

root@PC3DepA:/tmp/pycore.44725/PC3DepA.conf# ping 192.168.36.1
PING 192.168.36.1 (192.168.36.1) 56(84) bytes of data:
64 bytes from 192.168.36.1: icmp_seq=1 ttl=64 time=1.69 ms
64 bytes from 192.168.36.1: icmp_seq=2 ttl=64 time=0.934 ms
64 bytes from 192.168.36.1: icmp_seq=3 ttl=64 time=1.77 ms
64 bytes from 192.168.36.1: icmp_seq=4 ttl=64 time=0.806 ms
64 bytes from 192.168.36.1: icmp_seq=5 ttl=64 time=0.360 ms
^C
--- 192.168.36.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4084ms
rtt min/avg/max/mdev = 0.360/1.110/1.766/0.538 ms
root@PC3DepA:/tmp/pycore.44725/PC3DepA.conf#

```

Figura 29 - Resultado comando ping PC3DepA

```

root@Servidor1DepA:/tmp/pycore.44725/Servidor1DepA.conf# ping 192.168.36.1
PING 192.168.36.1 (192.168.36.1) 56(84) bytes of data:
64 bytes from 192.168.36.1: icmp_seq=1 ttl=64 time=1.13 ms
64 bytes from 192.168.36.1: icmp_seq=2 ttl=64 time=3.72 ms
64 bytes from 192.168.36.1: icmp_seq=3 ttl=64 time=4.65 ms
64 bytes from 192.168.36.1: icmp_seq=4 ttl=64 time=51.6 ms
^C
--- 192.168.36.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 1.127/15.274/51.604/21.014 ms
root@Servidor1DepA:/tmp/pycore.44725/Servidor1DepA.conf#

```

Figura 30 - Resultado comando ping ServidorqDepA

```

root@RA:/tmp/pycore.44725/RA.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.36.1 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 fe80::200:ff:feaa:4 prefixlen 64 scopeid 0x20<link>
    inet6 2001::1 prefixlen 64 scopeid 0x0<global>
    ether 00:00:00:aa:00:04 txqueuelen 1000 (Ethernet)
    RX packets 129 bytes 12572 (12,5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 235 bytes 19198 (19,1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Figura 26- Output do comando ifconfig -a

```

root@RA:/tmp/pycore.44725/RA.conf# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.36.22            ether   00:00:00:aa:00:02    C                     eth0
192.168.36.21            ether   00:00:00:aa:00:01    C                     eth0
192.168.36.10            ether   00:00:00:aa:00:03    C                     eth0
192.168.36.20            ether   00:00:00:aa:00:00    C                     eth0
root@RA:/tmp/pycore.44725/RA.conf#

```

Figura 32-Output do comando arp

Tendo isto, podemos então associar o endereço MAC correspondente a cada dispositivo:

- rA-> 00:00:00:aa:00:04
- PC1Dep1A-> 00:00:00:aa:00:00
- PC2Dep1A->00:00:00:aa:00:01
- PC3Dep1A-> 00:00:00:aa:00:02
- Servidor1DepA->00:00:00:aa:00:03

Podemos então definir a tabela de comutação pretendida:

<i>Mac Address</i>	<i>Interface</i>
00:00:00:aa:00:04	<i>eth1</i>
00:00:00:aa:00:00	<i>eth2</i>
00:00:00:aa:00:01	<i>eth3</i>
00:00:00:aa:00:02	<i>eth4</i>
00:00:00:aa:00:03	<i>eth5</i>

Conclusões

Com este trabalho, foi possível a consolidação de alguns temas lecionados na unidade curricular de Redes de Computadores.

Em particular, a realização deste trabalho prático permitiu que o grupo de trabalho tivesse a oportunidade de aprofundar conhecimentos relativos à camada de ligação lógica, sobretudo, na realização de problemas relativos ao estudo de endereços MAC, funcionamento do protocolo Ethernet e do protocolo ARP.