



UNIVERSIDADE DO MINHO
LICENCIATURA EM ENGENHARIA INFORMÁTICA

REDES DE COMPUTADORES

Trabalho Prático 3
**NÍVEL DE LIGAÇÃO LÓGICA: REDES
ETHERNET E PROTOCOLO ARP**

Mariana Gonçalves (a100662) Maya Gomes (a100822) Vicente Martins
(a100713)

May 3, 2023

Conteúdos

1	Introdução	3
2	Questões e respostas I - Captura e análise de Tramas Ethernet	3
2.1	Questão 1	5
2.2	Questão 2	5
2.3	Questão 3	5
2.4	Questão 4	6
2.5	Questão 5	7
2.6	Questão 6	7
3	Questões e respostas II - Protocolo ARP	9
3.1	Questão 1	11
3.2	Questão 2	12
3.3	Questão 3	14
3.4	Questão 4	16
3.5	Questão 5	16
3.6	Questão 6	17
4	Questões e respostas III - Domínios de colisão	19
4.1	Questão 1	19
4.2	Questão 2	21
5	Conclusão	25

1 Introdução

2 Questões e respostas I - Captura e análise de Tramas Ethernet

Pare a captura do Wireshark., e proceda da seguinte forma: Localize o estabelecimento da conexão entre o cliente e o servidor HTTP (sequência de tramas com as TCP flags TCP SYN, SYN- ACK, ACK ativas). Após a fase de estabelecimento seguro da conexão, obtenha o número de ordem da sequência de bytes capturada (coluna da esquerda na janela do Wireshark) correspondente à trama que transporta os primeiros dados aplicacionais enviados do cliente para o servidor (Application Data). Identifique também o número de ordem da trama com a resposta proveniente do servidor que contém os dados correspondentes ao acesso web realizado pelo cliente (browser). Note que os dados aplicacionais são enviados de forma segura usando o protocolo TLS (Transport Layer Security), mapeados para um segmento TCP, transportado num datagrama IP que, por sua vez, é encapsulado no campo de dados da trama Ethernet. Expanda a informação do nível da ligação de dados e observe o conteúdo da trama Ethernet (cabeçalho e dados (payload)).

Conforme sugerido, começamos a monitorizar a comunicação usando o Wireshark e abrimos o navegador para acessar o link solicitado. Durante a monitorização, não conseguimos identificar a mensagem de acesso ao servidor (HTTP GET) diretamente, pois estava criptografada usando o protocolo TLS. Optamos por selecionar um "Application Data".

Desta forma, reiniciamos a monitorização, acessamos ao site e paramos a monitorização para identificar imediatamente a mensagem desejada.

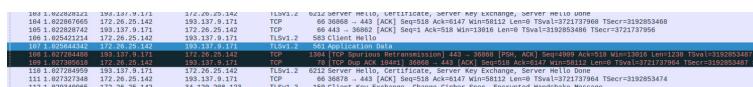


Figura 1: Trecho da captura de tráfego no acesso ao url especificado.

```

- Ethernet II, Src: LiteonTe_76:36:2f (d8:f3:bc:76:36:2f), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
  ▶ Destination: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
  ▶ Source: LiteonTe_76:36:2f (d8:f3:bc:76:36:2f)
  ▶ Type: IPv4 (0x0800)
  ▶ Internet Protocol Version 4, Src: 172.26.25.142, Dst: 193.137.9.171
  ▶ Transmission Control Protocol, Src Port: 36846, Dst Port: 443, Seq: 2763
  ▶ Transport Layer Security
    ▶ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls

0000 00 d0 03 ff 94 00 d8 f3 bc 76 36 2f 08 00 45 00 . . . . . . . . v6/..E
0010 02 23 a3 a5 40 00 40 06 04 53 ac 1a 19 8e c1 89 # . @ . S . . .
0020 09 ab 8f ee 01 bb be e4 e1 89 02 d0 1d df 80 18 . . . . . . . . .
0030 04 de 5b f1 00 00 01 01 08 0a dd d5 36 eb be 4f [ . . . . . . . . 6 - 0
0040 13 dd 17 03 03 01 ea 00 00 00 00 00 00 00 05 de . . . . . . . . .
0050 60 39 b8 8c 2e 4e 9d 8f 53 56 e2 e4 8b cf ea a9 9 . . . N . . SV . . .
0060 61 0a 4f 6e aa b0 90 7a b3 53 aa 8c 61 02 35 28 a . On . z . S . a . 5( .
0070 3f e2 9d d8 fd 5f d3 f8 a0 8f 99 59 c6 03 63 bc ? . . . _ . . Y . c .
0080 e0 3c 75 e8 b3 e5 32 53 a1 ed d9 5e 7e a1 e3 af < u . . 2S . . . ^ . .
0090 82 ae 73 58 b6 1d c8 27 8f e0 c6 41 6e 2a 84 25 . . S X . . . A n * % .
00a0 5d 47 d9 2c e9 85 14 b5 6b a8 f2 ae d7 de 9b 4f J G . . . k . . . 0
00b0 b7 22 96 bb 2b dd c1 e2 08 2a 3d 6f ec 11 d0 84 . " . + . . . T = o . .
00c0 61 10 d0 d6 14 4d c5 1b 46 66 35 c2 1a 5f c3 af a . . M . . F f 5 . .
00d0 b7 6e 21 e9 b5 54 5a cc f6 46 42 a6 f6 d0 1e b3 . n ! . T Z . . F B . .
00e0 59 03 7b f6 6c 14 65 bd 95 47 6e b5 63 54 5d f5 Y . { . l . e . . G n . c T ] .
```

Figura 2: 14 bytes de Ethernet.

```

- Internet Protocol Version 4, Src: 172.26.25.142, Dst: 193.137.9.171
  0100 . . . . = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 547
```

Figura 3: 20 bytes de IP.

```

- Transmission Control Protocol, Src Port: 36846, Dst Port: 443, Seq: 2763
  Source Port: 36846
  Destination Port: 443
  [Stream index: 0]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 495]
  Sequence Number: 2763 (relative sequence number)
  Sequence Number (raw): 3202670985
  [Next Sequence Number: 3258 (relative sequence number)]
  Acknowledgment Number: 85504 (relative ack number)
  Acknowledgment number (raw): 47193567
  1000 . . . . = Header Length: 32 bytes (8)
```

Figura 4: 32 bytes de TCP.

Com base nas figuras acima temos que o número de ordem da sequência de *bytes* capturada é de 14(Ethernet) + 20(IP) + 32(TCP) = 66.

```

0000 00 d0 03 ff 94 00 d8 f3 bc 76 36 2f 08 00 45 00 . . . . . . . . v6/..E
0010 02 23 a3 a5 40 00 40 06 04 53 ac 1a 19 8e c1 89 # . @ . S . . .
0020 09 ab 8f ee 01 bb be e4 e1 89 02 d0 1d df 80 18 . . . . . . . . .
0030 04 de 5b f1 00 00 01 01 08 0a dd d5 36 eb be 4f [ . . . . . . . . 6 - 0
0040 13 dd 17 03 03 01 ea 00 00 00 00 00 00 00 05 de . . . . . . . . .
0050 60 39 b8 8c 2e 4e 9d 8f 53 56 e2 e4 8b cf ea a9 9 . . . N . . SV . . .
0060 61 0a 4f 6e aa b0 90 7a b3 53 aa 8c 61 02 35 28 a . On . z . S . a . 5( .
0070 3f e2 9d d8 fd 5f d3 f8 a0 8f 99 59 c6 03 63 bc ? . . . _ . . Y . c .
0080 e0 3c 75 e8 b3 e5 32 53 a1 ed d9 5e 7e a1 e3 af < u . . 2S . . . ^ . .
0090 82 ae 73 58 b6 1d c8 27 8f e0 c6 41 6e 2a 84 25 . . S X . . . A n * % .
00a0 5d 47 d9 2c e9 85 14 b5 6b a8 f2 ae d7 de 9b 4f J G . . . k . . . 0
00b0 b7 22 96 bb 2b dd c1 e2 08 2a 3d 6f ec 11 d0 84 . " . + . . . T = o . .
00c0 61 10 d0 d6 14 4d c5 1b 46 66 35 c2 1a 5f c3 af a . . M . . F f 5 . .
00d0 b7 6e 21 e9 b5 54 5a cc f6 46 42 a6 f6 d0 1e b3 . n ! . T Z . . F B . .
00e0 59 03 7b f6 6c 14 65 bd 95 47 6e b5 63 54 5d f5 Y . { . l . e . . G n . c T ] .
```

Figura 5: Bytes .

Com base na figura acima verificamos que $16*4+2 = 66$ como queríamos demonstrar.

2.1 Questão 1

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

Com base na figura 6, podemos concluir que:

- MAC de origem: **00:d0:03:ff:94:00**
- MAC de destino: **d8:f3:bc:76:36:2f**

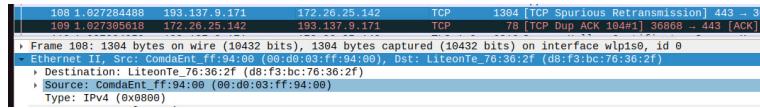


Figura 6: Informações Ethernet sobre a mensagem Application Data.

Considerando que é um endereço MAC (um identificador único associado à interface de comunicação que conecta um dispositivo à sua respetiva rede), o endereço MAC de Origem é o endereço da NIC (Controlador de Interface de Rede) associado ao dispositivo de origem, que neste caso é o nosso computador pessoal. Por outro lado, o endereço MAC de Destino é o endereço da NIC associada ao dispositivo de destino, que neste caso é o servidor da rede local.

2.2 Questão 2

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

Como podemos ver na figura 2, o valor do campo *Type* é de 0x0800 e este valor corresponde ao protocolo IP indicando que a trama em questão transporta datagramas IPv4. Este valor indica a camada para a qual o *payload* do pacote da camada Ethernet será passado(IP).

O valor hexadecimal do campo *Type* da trama *Ethernet* é 0x0800 quando se trata de uma trama IPv4. Esse valor indica que o protocolo de camada superior sendo transportado na trama *Ethernet* é o IPv4. O valor hexadecimal é uma forma numérica que representa esse protocolo de camada superior. A trama *Ethernet* é um pacote de dados que contém informações necessárias para transmitir dados através de uma rede.

2.3 Questão 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.

```

> Frame 107: 561 bytes on wire (4488 bits), 561 bytes captured (4488 bits) on interface wlp1s0, id 0
> Ethernet II, Src: LiteonTe_76:36:2f (08:f3:bc:76:36:2f), Dst: ComdAEnt_ff:94:00 (00:d0:03:ff:94:00)
> Internet Protocol Version 4, Src: 172.26.25.142, Dst: 193.137.9.171
    Identification: 0xa3a5 (41893)
    Flags: 0x40, Don't fragment
    ... 0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header checksum: 0x0453 [validation disabled]
    [header checksum status: Unverified]
    Source Address: 172.26.25.142
    Destination Address: 193.137.9.171
-> Transmission Control Protocol, Src Port: 36846, Dst Port: 443, Seq: 2763, Ack: 85504, Len: 495
    Source Port: 36846
    Destination Port: 443
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 495]
    Sequence Number: 2763 (relative sequence number)
    Sequence Number (raw): 3203570985
    [Next Sequence Number: 3258 (relative sequence number)]
    Acknowledgment Number: 85504 (relative ack number)
    Acknowledgment number (raw): 47193567
    1000 .... = Header Length: 32 bytes (8)
-> Flags: 0x018 (PSH, ACK)
    Window: 1246
    [calculated window size: 159488]
    [window size scaling factor: 128]
    Checksum: 0x5bf1 [unverified]
    [checksum status: Unverified]
    Urgent Pointer: 0
-> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    [Timestamps]
    [SEQ/ACK analysis]
    TCP payload (495 bytes)

```

Figura 7: Informação alusiva aos tamanhos necessários da mensagem *Application Data*.

Com base na figura anterior, conseguimos extrair vários valores:

- *Frame Length*: 561 Bytes;
- *IP Total Length*: 547 textitBytes;
- $IP\ Total\ Length = IP\ Header\ Length + TCP\ Header\ Length + Application = 547$ Bytes;
- $TCP\ Payload = 495\ Bytes$

Assim, sabendo o TCP *payload*, basta ir ao tamanho total da trama e subtrair esse valor e o dos cabeçalhos: *Frame Length menos o TCP Payload menos o IP Header menos o TCP Header*. Assim, temos:

$561 - 495 - 20 - 32 = 14$ Bytes. O que nos dá um *overhead* de $14/561 = 0,02495544$ (aproximadamente 2,5 % (1cd)).

2.4 Questão 4

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

Com base na figura abaixo temos que o endereço Ethernet da fonte é **00:d0:03:ff:94:00**.

Este corresponde ao servidor que é responsável por enviar a resposta ao pedido feito pelo seu computador pessoal. O endereço do servidor é o mesmo que o endereço MAC de destino na trama *Application Data*.

```

Frame 100: 1394 bytes on wire (10992 bits), 1394 bytes captured (10992 bits) on interface wlpis0, id 0
Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00), Dst: LiteonTe_76:36:2f (d8:f3:bc:76:36:2f)
  Destination: LiteonTe_76:36:2f (d8:f3:bc:76:36:2f)
    Address: LiteonTe_76:36:2f (d8:f3:bc:76:36:2f)
      .... .0 ..... .... .... = LG bit: Globally unique address (factory default)
      .... .0 ..... .... .... = IG bit: Individual address (unicast)
  Source: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
    Address: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
      .... .0 ..... .... .... = LG bit: Globally unique address (factory default)
      .... .0 ..... .... .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)

```

Figura 8: Informações Ethernet sobre da mensagem *Application Data*.

2.5 Questão 5

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

Com base na figura 8 o endereço MAC do destino é **d8:f3:bc:76:36:2f** correspondendo ao nosso computador pessoal.

2.6 Questão 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.

```

108 1.027284480 193.137.9.171 172.26.25.142 TCP 1384 [TCP Spurious Retransmission] 443 → 3686
109 1.027285018 172.26.25.142 193.137.9.171 TCP 78 [TCP Dup ACK 104#1] 36868 → 443 [ACK] 5e
110 1.027285095 193.137.9.171 172.26.25.142 TLSV1.2 6212 Server Hello, Certificate, Server Key Exchange, Session Ticket, Application Data
Frame 108: 1394 bytes on wire (10932 bits), 1394 bytes captured (10932 bits) on interface wlpis0, id 0
> Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00), Dst: LiteonTe_76:36:2f (d8:f3:bc:76:36:2f)
> Internet Protocol Version 4, Src: 193.137.9.171, Dst: 172.26.25.142
> Transmission Control Protocol, Src Port: 443, Dst Port: 36868, Seq: 4999, Ack: 518, Len: 1238

```

Figura 9: Informações sobre os vários protocolos contidos na trama recebida.

Com base na figura acima, identificamos 3 protocolos: *Ethernet*, *IPv4*, *TCP*. As figuras abaixo justificam os vários protocolos contidos na trama recebida.

Figura 10: Campo referente aos dados do Ethernet.

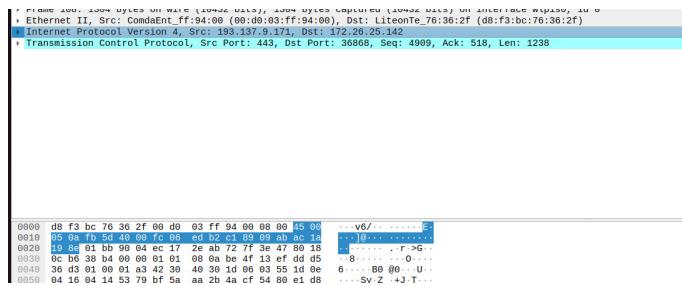


Figura 11: Campo referente aos dados do IPv4.

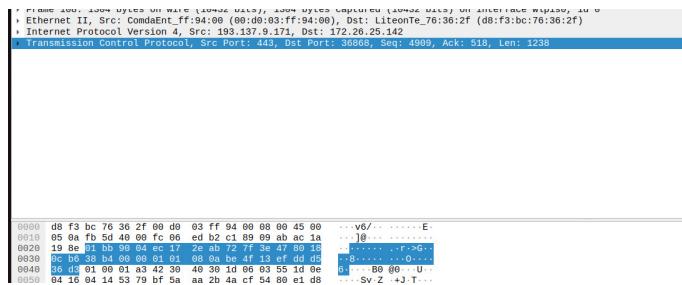


Figura 12: Campo referente aos dados do TCP.

3 Questões e respostas II - Protocolo ARP

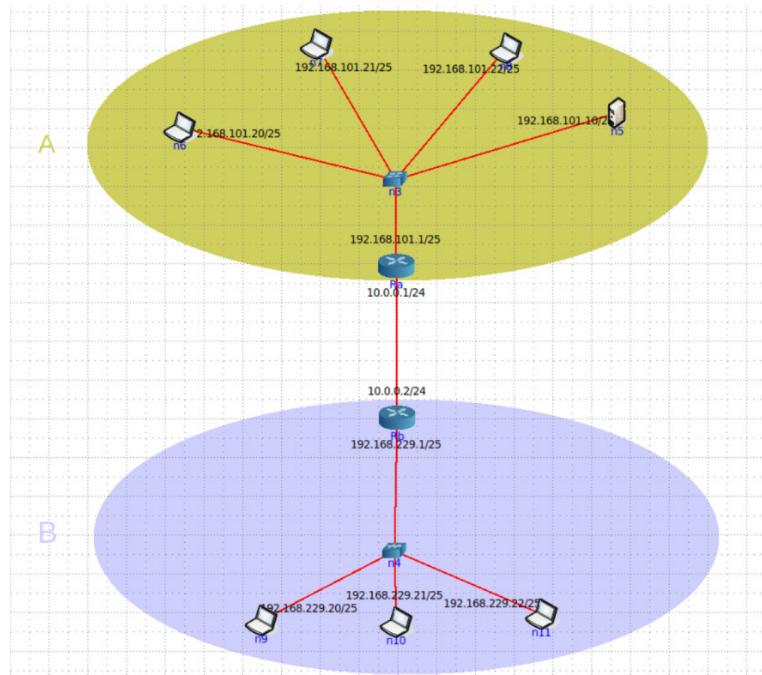


Figura 13: Topologia CORE criada com base no enunciado.

Selecione um PC de um dos departamentos à sua escolha e inicie a captura de tráfego com o Wireshark do CORE. A partir desse sistema efetue ping para dois PCs localizados na outra rede (departamento). Pare a captura de tráfego no Wireshark e localize o tráfego ARP, usando o filtro arp.

```

root@n7:/tmp/pycore.41631/n7.conf# ping 192.168.229.20
PING 192.168.229.20 (192.168.229.20) 56(84) bytes of data.
64 bytes from 192.168.229.20: icmp_seq=1 ttl=62 time=0.147 ms
64 bytes from 192.168.229.20: icmp_seq=2 ttl=62 time=0.152 ms
64 bytes from 192.168.229.20: icmp_seq=3 ttl=62 time=0.103 ms
64 bytes from 192.168.229.20: icmp_seq=4 ttl=62 time=0.084 ms
64 bytes from 192.168.229.20: icmp_seq=5 ttl=62 time=0.125 ms
64 bytes from 192.168.229.20: icmp_seq=6 ttl=62 time=0.128 ms
64 bytes from 192.168.229.20: icmp_seq=7 ttl=62 time=0.117 ms
64 bytes from 192.168.229.20: icmp_seq=8 ttl=62 time=0.144 ms
64 bytes from 192.168.229.20: icmp_seq=9 ttl=62 time=0.095 ms
64 bytes from 192.168.229.20: icmp_seq=10 ttl=62 time=0.624 ms
...
-- 192.168.229.20 ping statistics --
10 packets transmitted, 10 received, 0% packet loss, time 8194ms
rtt min/avg/max/mdev = 0.084/0.171/0.624/0.152 ms
root@n7:/tmp/pycore.41631/n7.conf# ping 192.168.229.21
PING 192.168.229.21 (192.168.229.21) 56(84) bytes of data.
64 bytes from 192.168.229.21: icmp_seq=1 ttl=62 time=0.297 ms
64 bytes from 192.168.229.21: icmp_seq=2 ttl=62 time=0.107 ms
64 bytes from 192.168.229.21: icmp_seq=3 ttl=62 time=0.196 ms
64 bytes from 192.168.229.21: icmp_seq=4 ttl=62 time=0.121 ms
64 bytes from 192.168.229.21: icmp_seq=5 ttl=62 time=0.117 ms
64 bytes from 192.168.229.21: icmp_seq=6 ttl=62 time=0.111 ms
64 bytes from 192.168.229.21: icmp_seq=7 ttl=62 time=0.072 ms
64 bytes from 192.168.229.21: icmp_seq=8 ttl=62 time=0.069 ms
64 bytes from 192.168.229.21: icmp_seq=9 ttl=62 time=0.118 ms
...
-- 192.168.229.21 ping statistics --
9 packets transmitted, 9 received, 0% packet loss, time 8195ms
rtt min/avg/max/mdev = 0.069/0.134/0.297/0.067 ms
root@n7:/tmp/pycore.41631/n7.conf#

```

Figura 14: Comandos -ping do PC n7 para os PCs n9 e n10.

No.	Time	Source	Destination	Protocol	Length	Info
5	3.093636197	00:00:00_aa:00:05	Broadcast	ARP	42	Who has 192.168.101.1? Tell 192.168.101.21
6	3.093977785	00:00:00_aa:00:00	00:00:00_aa:00:05	ARP	42	192.168.101.1 is at 00:00:00_aa:00:05
23	8.191248994	00:00:00_aa:00:00	00:00:00_aa:00:05	ARP	42	Who has 192.168.101.21? Tell 192.168.101.1
25	8.191270635	00:00:00_aa:00:05	00:00:00_aa:00:00	ARP	42	192.168.101.21 is at 00:00:00_aa:00:05

Figura 15: Wireshark do n9 com filtro arp.

No.	Time	Source	Destination	Protocol	Length	Info
16	7.745766578	00:00:00_aa:00:00	00:00:00_aa:00:05	ARP	42	Who has 192.168.101.21? Tell 192.168.101.1
17	7.745699569	00:00:00_aa:00:05	00:00:00_aa:00:00	ARP	42	Who has 192.168.101.1? Tell 192.168.101.21
18	7.745744892	00:00:00_aa:00:05	00:00:00_aa:00:00	ARP	42	192.168.101.21 is at 00:00:00_aa:00:05
19	7.745765992	00:00:00_aa:00:00	00:00:00_aa:00:05	ARP	42	192.168.101.1 is at 00:00:00_aa:00:05

Figura 16: Wireshark do n10 com filtro arp.

3.1 Questão 1

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

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

Primeiramente fez-se ping do n7 par o n6 como segue na figura abaixo.

```
root@n7:/tmp/pycore.41631/n7.conf# ping 192.168.101.20
PING 192.168.101.20 (192.168.101.20) 56(84) bytes of data.
64 bytes from 192.168.101.20: icmp_seq=1 ttl=64 time=0.111 ms
64 bytes from 192.168.101.20: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 192.168.101.20: icmp_seq=3 ttl=64 time=0.064 ms
64 bytes from 192.168.101.20: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 192.168.101.20: icmp_seq=5 ttl=64 time=0.063 ms
64 bytes from 192.168.101.20: icmp_seq=6 ttl=64 time=0.039 ms
64 bytes from 192.168.101.20: icmp_seq=7 ttl=64 time=0.068 ms
64 bytes from 192.168.101.20: icmp_seq=8 ttl=64 time=0.066 ms
64 bytes from 192.168.101.20: icmp_seq=9 ttl=64 time=0.081 ms
64 bytes from 192.168.101.20: icmp_seq=10 ttl=64 time=0.063 ms
^C
```

Figura 17: Comando ping.

De seguida realizou-se o comando arp -a como segue na figura abaixo. A esquerda da tabela aparece o símbolo "?", pois não está nenhum nome de host associado.

```
root@n7:/tmp/pycore.41631/n7.conf# arp -a
? (192.168.101.1) at 00:00:00:aa:00:00 [ether] on eth0
? (192.168.101.20) at 00:00:00:aa:00:04 [ether] on eth0
root@n7:/tmp/pycore.41631/n7.conf# arp -n
```

Figura 18: Tabela resultante do comando arp -a .

De seguida para conseguir-se uma tabela mais explicita realizou-se o comando arp -n como segue na próxima figura.

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.101.1	ether	00:00:00:aa:00:00	C		eth0
192.168.101.20	ether	00:00:00:aa:00:04	C		eth0

Figura 19: Tabela resultante do comando arp -n.

O significado das respetivas colunas é:

- *Address*: o endereço IP do *host* remoto na rede local;
- *HWtype*: o tipo de hardware usado para a conexão de rede, como *Ethernet* ou *Wi-Fi*;

- *HWaddress* : o endereço físico (MAC) do *host* remoto na rede local;
 - *Flags Mask*: informa que a conexão foi estabelecida com sucesso (C);
 - *Iface*: a interface de rede em que o *host* remoto está conectado ao sistema local.
- b. *Indique, justificando, qual o equipamento da intranet em causa que poderá apresentar a maior tabela ARP em termos de número de entradas.*

A tabela ARP é usada pelos dispositivos numa rede para mapear endereços IP em endereços MAC correspondentes. Sendo assim, cada entrada na tabela ARP corresponde a um par de endereços IP e MAC.

O equipamento com maior tabela ARP em termos de número de entradas é o *router*. Isso acontece porque o *router* geralmente é o ponto central de uma rede e é responsável por encaminhar o tráfego entre diferentes redes. Com isso, o *router* necessita manter uma tabela ARP para cada rede conectada a ele.

3.2 Questão 2

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

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

De forma a observar o pedido ARP efetuamos um *ping* a partir do Portátil n7 para o n9. Sabemos que se trata do *request*, pois o campo "Address Resolution Protocol" tem a palavra "*request*". Obtendo no Wireshark as seguintes entradas:

No.	Time	Source	Destination	Protocol	Length	Info
5	3.093036107	00:00:00:aa:00:05	Broadcast	ARP	42	42 Who has 192.168.101.1? Tell 192.168.101.21
6	3.093077785	00:00:00:aa:00:00	00:00:00:aa:00:05	ARP	42	192.168.101.1 is at 00:00:00:aa:00:00
23	8.191248994	00:00:00:aa:00:00	00:00:00:aa:00:05	ARP	42	Who has 192.168.101.21? Tell 192.168.101.1
25	8.191270635	00:00:00:aa:00:05	00:00:00:aa:00:00	ARP	42	192.168.101.21 is at 00:00:00:aa:00:05

```

> Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth7.0.3d, id 0
> Ethernet II, Src: 00:00:00:aa:00:05 (00:00:00:aa:00:05), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: 00:00:00:aa:00:05 (00:00:00:aa:00:05)
  > Type: ARP (0x0806)
  > Address Resolution Protocol (request)

```

Figura 20: Tabela ARP.

Tal como conseguimos verificar na figura acima, os endereços são:

- Origem: 00:00:00:aa:00:05
- Destino: ff:ff:ff:ff:ff:ff

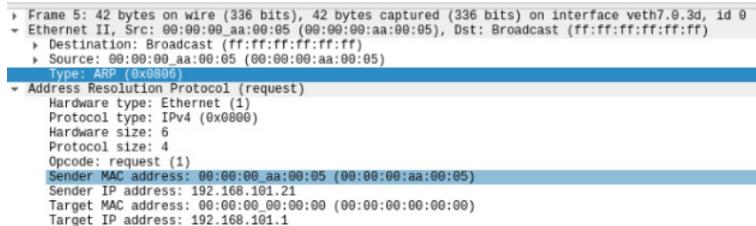
O endereço de destino é este, pois como o a tabela ARP estava vazia, o Portátil n7 não conhecia o IP fornecido, ou seja, necessitava de enviar a todos os nós da rede local a quem pertencia o IP dado.

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

Com base na figura anterior temos que o valor hexadecimal do campo Tipo é **0x0806**.

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

Primeiramente, uma forma de saber-se que se trata de um pedido ARP é a coluna *Protocol* da tabela Wireshark, pois como na figura acima essa possui a palavra "ARP" como conteúdo.



The screenshot shows a Wireshark capture of an ARP request frame. The frame details pane shows the following information:

- Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth7.0.3d, id 0
- Ethernet II, Src: 00:00:00:aa:00:05 (00:00:00:aa:00:05), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Source: 00:00:00:aa:00:05 (00:00:00:aa:00:05)
- 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:00:05 (00:00:00:aa:00:05)
- Sender IP address: 192.168.101.21
- Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
- Target IP address: 192.168.101.1

Figura 21: Detalhes da tabela ARP.

Conseguimos também notar o campo "Address Resolution Protocol" e o o *Opcode* igual a 1 afirmando que se trata de um *request*.

Para além disso, podemos concluir que trata-se de um pedido ARP, pois os endereços contidos na mensagem são endereços MAC.

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

O host de origem pergunta a todos os nós da sua rede se o seu IP é igual ao que está à procura. Em caso afirmativo o host quer obter como resposta o seu endereço MAC.

3.3 Questão 3

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

No.	Time	Source	Destination	Protocol	Length	Info
5	3.093036107	00:00:00_aa:00:05	Broadcast	ARP	42	Who has 192.168.101.1? Tell 192.168.101.21
6	3.093077785	00:00:00_aa:00:00	00:00:00_aa:00:05	ARP	42	192.168.101.1 is at 00:00:00_aa:00:00
23	8.191248994	00:00:00_aa:00:00	00:00:00_aa:00:05	ARP	42	Who has 192.168.101.21? Tell 192.168.101.1
25	8.191270635	00:00:00_aa:00:05	00:00:00_aa:00:00	ARP	42	192.168.101.21 is at 00:00:00_aa:00:05

Frame 6: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth7.0.3d, id 0
Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00_aa:00:05 (00:00:00:aa:00:05)
> Destination: 00:00:00_aa:00:05 (00:00:00:aa:00:05)
> Source: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
Version: 2 (0x0002)
Type: 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: 192.168.101.1
Target MAC address: 00:00:00_aa:00:05 (00:00:00:aa:00:05)
Target IP address: 192.168.101.21

Figura 22: Tabela ARP.

- a. Qual o valor do campo ARP opcode? O que especifica?

O valor do campo "ARP opcode" é 2, e especifica que é uma mensagem de resposta, isto é, do tipo *reply*.

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

A resposta ao pedido ARP está no campo Sender MAC address, sendo neste caso **00:00:00_aa:00:00**.

- 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.

```

root@n7:/tmp/pycore_41631/n7.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.101.21 netmask 255.255.255.128 broadcast 0.0.0.0
                inet6 fe80::2001:fffea%eth0 brd fe80::ff:feff%eth0 mngtmpd
        ether 00:00:00:aa:00:05 txqueuelen 1000 (Ethernet)
        RX packets 3413 bytes 278333 (276.3 kB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 73 bytes 6310 (6.3 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 60 bytes 5180 (5.1 kB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 60 bytes 5180 (5.1 kB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Figura 23: Comando ifconfig.

Desta forma, com base na figura acima conseguimos ver que o *ether* é *00:00:00:aa:00:05* sendo esse o endereço MAC de origem.

Destination	Gateway	Gw mask	Flags	MSS	Window	irtt	Iface
0.0.0.0	192.168.101.1	0.0.0.0	UG	0	0	0	eth0
192.168.101.0	0.0.0.0	255.255.255.128	U	0	0	0	eth0

Figura 24: Comando netstat -rn .

Address	Hwdtype	Hlladdress	Flags	Mask	Iface
192.168.101.1	ether	00:00:00:aa:00:00	C		eth0
192.168.101.22	ether	00:00:00:aa:00:06	C		eth0
192.168.101.20	ether	00:00:00:aa:00:04	C		eth0

Figura 25: Comando arp.

Com base na figura a cima e na coluna "HWaddress" temos que o endereço MAC de destino é *00:00:00:aa:00:00* (router Ra).

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

A resposta ARP é enviada como um *unicast* contrariamente ao pedido que é *broadcast*. Desta forma, com o *unicast* a resposta é enviada apenas para o dispositivo que fez a solicitação, sendo mais eficiente e económico em termos de recursos de rede.

Sendo que o ARP reply é uma resposta direta à solicitação ARP, o envio de uma mensagem *broadcast* seria desnecessária e geraria tráfego desnecessário na rede. Com o *broadcast* a mensagem é enviada para todos os dispositivos na rede, sendo que cada dispositivo tem de processar a mensagem ARP.

3.4 Questão 4

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

O *ping* feito ao segundo PC não originou pacote ARP, pois esse situa-se noutra rede. A mensagem chega ao router Ra mas não o atravessa, pois para a frente a rede é diferente.

De uma forma geral, quando um PC não pertence à mesma rede do PC que está tentando realizar o *ping*, é possível que não ocorra nenhuma comunicação ARP. Isso ocorre, pois para que ocorra uma comunicação ARP é necessário que os dispositivos estejam na mesma rede e sejam capazes de comunicar diretamente uns com os outros. Quando o PC que está tentando realizar o *ping* e o PC de destino está em redes diferentes, o PC de origem precisa de enviar o pacote para o *router*, e não para o endereço MAC do PC de destino, para que esse possa ser encaminhado para a rede de destino.

3.5 Questão 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.

```
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: 192.168.101.1
  Target MAC address: 00:00:00_aa:00:05 (00:00:00:aa:00:05)
  Target IP address: 192.168.101.21
```

Figura 26: Comando arp.

Com base na figura acima, o campo "*Hardware type*"(Tipo de endereço de hardware) especifica o formato dos endereços das camadas de ligação lógica, neste caso é 1, isto é, o *Ethernet*, o campo "*Protocol type*"informa-nos que se trata de IPv4 (0x0800). Relativamente ao tamanho dos endereços das camadas de rede e de ligação lógica que se pretendem mapear o campo "*Hardware size*"indica-nos um tamanho de 6 *bytes* e o campo "*Protocol size*"indica-nos um tamanho de 4 *bytes*.

Desta forma, no caso do campo "*Hardware Size*", o valor 6 é geralmente utilizado para indicar que os endereços MAC são representados em 6 *bytes* (48 *bits*), o que é o tamanho padrão dos endereços MAC utilizados em redes Ethernet. Já no caso do campo "*Protocol Size*", o valor 4 é utilizado para indicar que os endereços IP são representados em 4 *bytes* (32 *bits*), que é o tamanho padrão dos endereços IPv4.

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

- 1. O PC Origem n7 envia uma mensagem ARP broadcast para descobrir o endereço MAC do Router Ra: ARP Request:
 - Source MAC: MAC do PC Origem n7
 - Source IP: 192.168.101.21
 - Destination MAC: FF:FF:FF:FF:FF:FF
 - Destination IP: 192.168.101.1
- 2. O Router Ra responde à mensagem ARP do PC Origem n7 com o seu endereço MAC: ARP Reply:
 - Source MAC: MAC do Router Ra
 - Source IP: 192.168.101.1
 - Destination MAC: MAC do PC Origem n7
 - Destination IP: 192.168.101.21
- 3. O PC Origem n7 envia um pacote ICMP echo request para o Router Ra: ICMP Echo Request:
 - Source MAC: MAC do PC Origem n7
 - Source IP: 192.168.101.21
 - Destination MAC: MAC do Router Ra
 - Destination IP: 192.168.229.20
- 4. O Router Ra consulta a sua tabela de roteamento e encaminha o pacote para o Router Rb: (Encaminhamento realizado através da interface ligada ao Router Rb)
 - MAC do Router Ra -> MAC do Router Rb
- 5. O Router Rb recebe o pacote e consulta a sua tabela de ARP para determinar o endereço MAC do destino: ARP Request:
 - Source MAC: MAC do Router Rb
 - Source IP: 192.168.229.1
 - Destination MAC: FF:FF:FF:FF:FF:FF
 - Destination IP: 192.168.229.20

- 6. O Hub recebe a mensagem ARP broadcast e a reenvia para todos os PCs.
- 7. O PC Destino responde ao ARP Request do Router Rb com o seu endereço MAC: ARP Reply:
 - Source MAC: MAC do PC Destino
 - Source IP: 192.168.229.20
 - Destination MAC: MAC do Router Rb
 - Destination IP: 192.168.229.1
- 8. O Router Rb recebe o ARP Reply do PC Destino e encaminha o pacote ICMP echo request: (Encaminhamento realizado através da interface ligada ao Router Ra)
 - MAC do Router Rb -> MAC do Router Ra
- 9. O Router Ra recebe o pacote ICMP echo request e consulta a sua tabela de roteamento.
- 10. O Router Ra encaminha o pacote ICMP echo request para o PC Destino.

4 Questões e respostas III - Domínios de colisão

4.1 Questão 1

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. 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 concluir?

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.

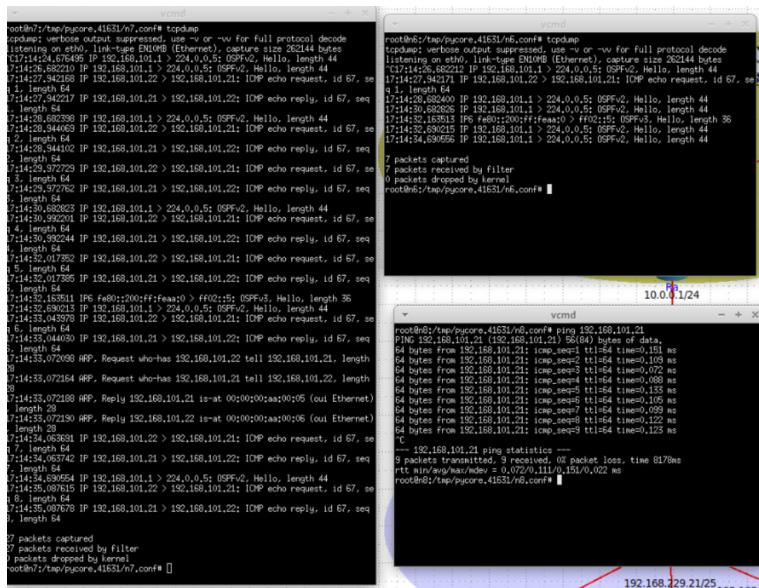


Figura 27: Tcpdump em LAN comutada.

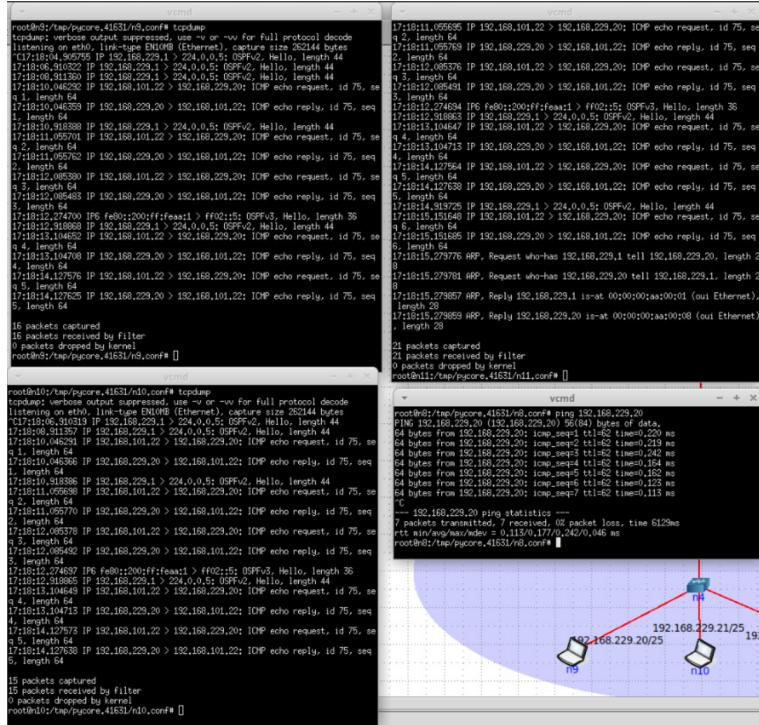


Figura 28: Tcpcdump em LAN partilhada.

No departamento A, que utiliza uma LAN comutada com *switches*, podemos observar que o tráfego é muito mais eficiente e organizado. Cada dispositivo na rede está conectado diretamente a um *switch*, o que permite que as informações sejam enviadas apenas para os dispositivos relevantes na rede, reduzindo a quantidade de tráfego desnecessário. Quando é gerado tráfego intra-departamento, como através do comando *ping*, podemos observar que o tráfego flui diretamente do dispositivo de origem para o dispositivo de destino, sem percorrer todos os dispositivos da rede.

No departamento B, que utiliza uma LAN partilhada com *hubs*, podemos observar que o tráfego não é tão eficiente. Cada dispositivo na rede está conectado diretamente a um *hub*, o que significa que todas as informações são enviadas para todos os dispositivos na rede, mesmo que não sejam relevantes. Quando é gerado tráfego intra-departamento, como através do comando *ping*, podemos observar que o tráfego flui para todos os dispositivos na rede antes de chegar ao dispositivo de destino.

Podemos identificar algumas diferenças importantes entre o *hub* e o *switch*: o *hub* possui apenas um domínio de colisão, enquanto o *switch* tem diferentes portas com diferentes domínios de colisão. O *hub* não pode armazenar endereços MAC, enquanto o *switch* pode.

Com base nesses resultados, podemos concluir que a utilização de *switches* é muito mais eficiente do que a utilização de *hubs* no contexto de controlar ou dividir domínios de colisão. Os *switches* permitem que o tráfego seja direcionado apenas para os dis-

sitivos relevantes na rede, enquanto os *hubs* enviam todas as informações para todos os dispositivos na rede.

4.2 Questão 2

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

Para construir a tabela de endereços MAC do departamento A, é necessário estabelecer uma troca de pacotes entre os dispositivos desse departamento. Para isso, pode-se realizar um *ping* do n7 para o n6 e, em seguida, do *router* Ra para o Servidor n5. Seguindo esse processo, temos o seguinte fluxo:

- Com o primeiro *ping*, o n7 envia um pacote para o n6, iniciando a comunicação.
- O *switch* recebe o pacote na interface 1 e adiciona o endereço MAC de origem (n7) à sua tabela de endereços MAC. O *switch* recebe o pacote na interface 1 e adiciona o endereço MAC de origem (n7) à sua tabela de endereços MAC.
- Como o endereço MAC do n6 não está disponível na tabela de endereços MAC do *switch*, o pacote é encaminhado para todas as interfaces ativas, exceto a 1.
- O n6 recebe o pacote e responde à n7. O *switch* recebe esse pacote na interface 3 e adiciona o endereço MAC de origem (n6) à sua tabela de endereços MAC.
- Agora, n7 e n6 podem se comunicar entre si, mas o Servidor n5 não pode ver os pacotes transmitidos entre eles. Quando um dispositivo se comunicar com o Servidor n5, o seu endereço MAC também será atualizado na tabela de endereços MAC do *switch*. Então, ao realizar o segundo *ping*, inicia-se essa comunicação.
- O *switch* recebe o pacote do *router* Ra pela interface 4 e adiciona o endereço desse dispositivo à sua tabela de endereços MAC. Novamente, o endereço MAC do Servidor n5 não está na tabela do *switch*, então o pacote é enviado para todos os outros dispositivos, exceto o *router* Ra. O Servidor n5 recebe o pacote e responde pela interface 2 ao *router*. O *switch* guarda o endereço MAC do servidor em sua tabela. A partir desse momento, todos os dispositivos podem se comunicar entre si.
- O processo é também realizado com o n8.

```

root@n7:/tmp/pycore.41631/n7.conf# ping 192.168.101.20
PING 192.168.101.20 (192.168.101.20) 56(84) bytes of data.
64 bytes from 192.168.101.20: icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from 192.168.101.20: icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from 192.168.101.20: icmp_seq=3 ttl=64 time=0.067 ms
64 bytes from 192.168.101.20: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 192.168.101.20: icmp_seq=5 ttl=64 time=0.101 ms
64 bytes from 192.168.101.20: icmp_seq=6 ttl=64 time=0.063 ms
64 bytes from 192.168.101.20: icmp_seq=7 ttl=64 time=0.080 ms
64 bytes from 192.168.101.20: icmp_seq=8 ttl=64 time=0.068 ms
64 bytes from 192.168.101.20: icmp_seq=9 ttl=64 time=0.067 ms
^C
--- 192.168.101.20 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8180ms
rtt min/avg/max/mdev = 0.046/0.069/0.101/0.013 ms
root@n7:/tmp/pycore.41631/n7.conf#

```

Figura 29: Comando -ping de n7 para n6.

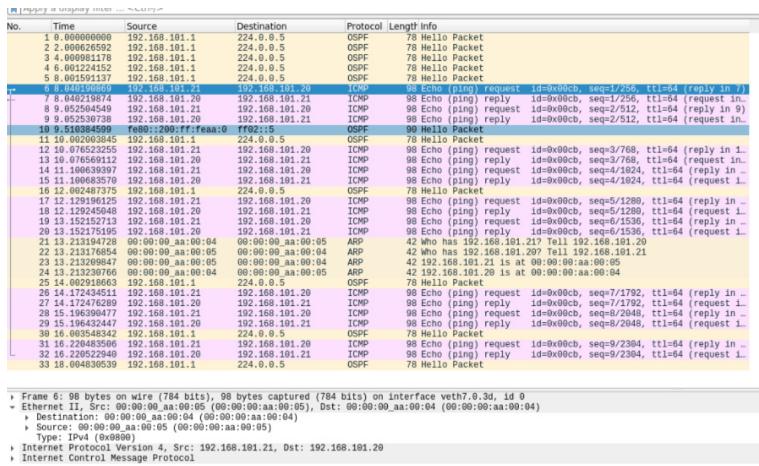


Figura 30: Captura do tráfego de n7 para n6.

```

root@Ra:/tmp/pycore_41631/Ra.conf# ping 192.168.101.10
PING 192.168.101.10 (192.168.101.10) 56(84) bytes of data.
64 bytes from 192.168.101.10: icmp_seq=1 ttl=64 time=0.061 ms
64 bytes from 192.168.101.10: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.101.10: icmp_seq=3 ttl=64 time=0.079 ms
64 bytes from 192.168.101.10: icmp_seq=4 ttl=64 time=0.079 ms
64 bytes from 192.168.101.10: icmp_seq=5 ttl=64 time=0.120 ms
64 bytes from 192.168.101.10: icmp_seq=6 ttl=64 time=0.081 ms
64 bytes from 192.168.101.10: icmp_seq=7 ttl=64 time=0.102 ms
64 bytes from 192.168.101.10: icmp_seq=8 ttl=64 time=0.047 ms
64 bytes from 192.168.101.10: icmp_seq=9 ttl=64 time=0.073 ms
64 bytes from 192.168.101.10: icmp_seq=10 ttl=64 time=0.045 ms
64 bytes from 192.168.101.10: icmp_seq=11 ttl=64 time=0.076 ms
64 bytes from 192.168.101.10: icmp_seq=12 ttl=64 time=0.069 ms
64 bytes from 192.168.101.10: icmp_seq=13 ttl=64 time=0.087 ms
64 bytes from 192.168.101.10: icmp_seq=14 ttl=64 time=0.080 ms
^C
--- 192.168.101.10 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13315ms
rtt min/avg/max/mdev = 0.045/0.076/0.120/0.018 ms
root@Ra:/tmp/pycore_41631/Ra.conf#

```

Figura 31: Comando -ping de Ra para n5.

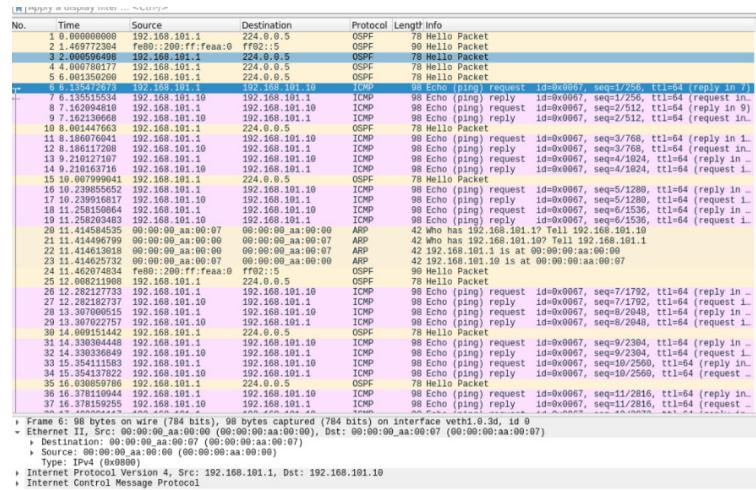


Figura 32: Câptura do tráfego de Ra para n5.

```

root@n8:/tmp/pycore.41631/n8.conf# ping 192.168.101.21
PING 192.168.101.21 (192.168.101.21) 56(84) bytes of data.
64 bytes from 192.168.101.21: icmp_seq=1 ttl=64 time=0.091 ms
64 bytes from 192.168.101.21: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 192.168.101.21: icmp_seq=3 ttl=64 time=0.064 ms
64 bytes from 192.168.101.21: icmp_seq=4 ttl=64 time=0.095 ms
64 bytes from 192.168.101.21: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 192.168.101.21: icmp_seq=6 ttl=64 time=0.068 ms
64 bytes from 192.168.101.21: icmp_seq=7 ttl=64 time=0.068 ms
64 bytes from 192.168.101.21: icmp_seq=8 ttl=64 time=0.075 ms
64 bytes from 192.168.101.21: icmp_seq=9 ttl=64 time=0.066 ms
64 bytes from 192.168.101.21: icmp_seq=10 ttl=64 time=0.083 ms
64 bytes from 192.168.101.21: icmp_seq=11 ttl=64 time=0.091 ms
64 bytes from 192.168.101.21: icmp_seq=12 ttl=64 time=0.044 ms
```
--- 192.168.101.21 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11289ms
rtt min/avg/max/mdev = 0.044/0.072/0.095/0.015 ms
root@n8:/tmp/pycore.41631/n8.conf#

```

Figura 33: Comando -ping de n8 para n7.

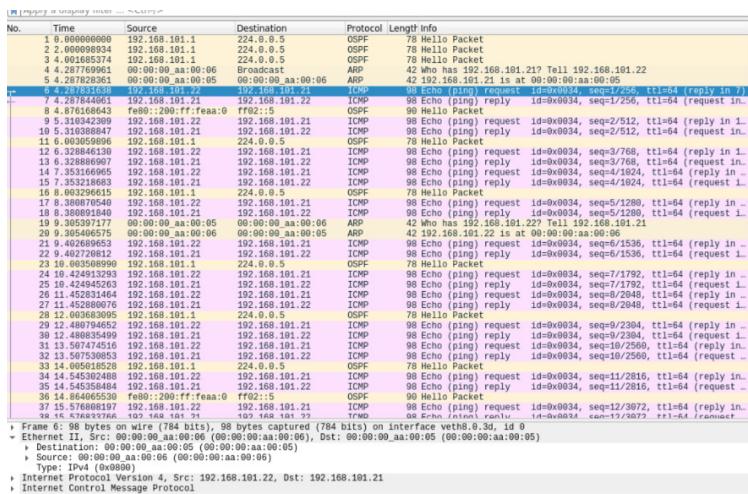


Figura 34: Câptura do tráfego de n8 para n7.

Assim sendo, olhando para os campos de “*Source*” e “*Destination*”, de cada imagem acima, obtemos a tabela resultante de endereços MAC formada pelo *switch*:

| Device | Interface | MAC Address       |
|--------|-----------|-------------------|
| n7     | 1         | 00:00:00:aa:00:05 |
| n5     | 2         | 00:00:00:aa:00:07 |
| n6     | 3         | 00:00:00:aa:00:04 |
| Ra     | 4         | 00:00:00:aa:00:00 |
| n8     | 5         | 00:00:00:aa:00:06 |

Tabela 1: Tabela final de endereços MAC do Departamento A.

## 5 Conclusão

Na primeira parte do trabalho, analisamos o tráfego de tramas *Ethernet* para entender como os endereços MAC são relacionados aos sistemas físicos e o conceito de encapsulamento protocolar. Na segunda parte, estudamos o protocolo ARP e os diversos conceitos relacionados, como por exemplo, a análise de mensagens de pedidos e resposta. Na terceira parte, exploramos a questão dos domínios de colisão e como os *switches* são importantes para solucionar essas colisões, especialmente através da construção de tabelas de endereços MAC.

Em geral, o trabalho correu bem e conseguimos responder a todas as perguntas, o que nos permitiu consolidar e aprofundar os diversos conceitos.