Universidade do Minho

Braga, Portugal

TRABALHO PRÁTICO 3 - RELATÓRIO

Grupo 82

Redes de Computadores

Engenharia Informática 2024/25

Equipa de Trabalho:

A106932 - Luís António Peixoto Soares

A104438 - Gonçalo Filipe Duarte Barbosa

A104619 - Gonçalo da Silva Carmo

16 de Maio

Índice

1. Objetivos

Este relatório tem como objetivo aprofundar o nosso conhecimento em redes ethernet e redes wi-fi, assim como na camada de ligação lógica e no protocolo ARP(Address Resolution Protocol), através da resolução de exercícios divididos em duas partes, sendo a primeira delas focada em redes ethernet e no protocolo ARP e a segunda focada em redes wi-fi.

2. 1º Parte

2.1. Exercício 1

A topologia de rede representada na figura abaixo é constituída por: (i) uma LAN comutada que interliga os hosts Beauty, Beast e o servidor DServer (Disney Server) através de um switch (SW1) ao router de acesso Rxy; (ii) uma LAN partilhada que interliga os hosts Jasmine, Aladdin através de um hub ao router de acesso (R1); e (iii) uma rede IP ponto-a-ponto que interliga as duas LANs. Construa a topologia indicada e particularize o router Rxy com o seu número de grupo (e.g., R27 para o grupo 7 do turno PL2). De igual forma, o endereço IP do servidor DServer deve ser alterado para incluir o seu número de grupo no identificador da host interface (4º octeto), e.g. 10.0.2.27, bem como o seu endereço MAC, e.g., 00:00:00:AA:BB:27.

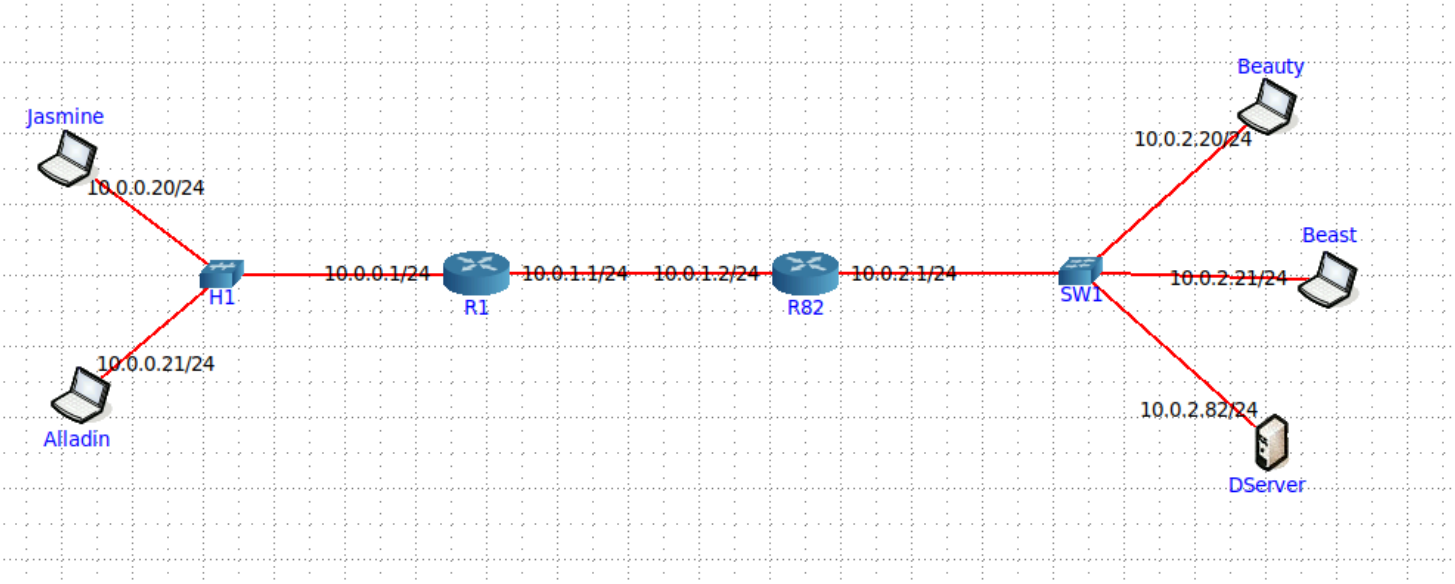


Figura 1: Topologia exercício 1

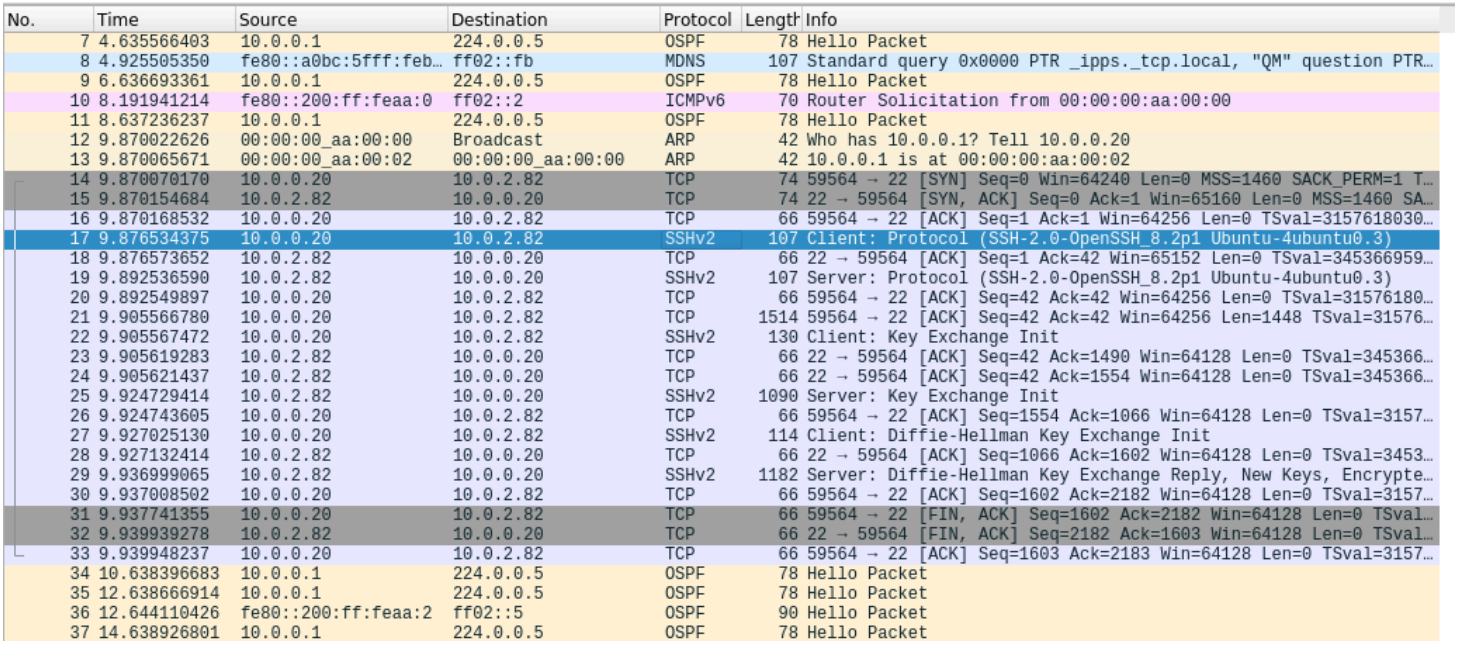
Ative a topologia de rede e ative o Wireshark na interface de saída do host Jasmine. Antes de ver a sua série favorita, a Jasmine começa por abrir um terminal e estabelecer um acesso seguro ao servidor DServer usando o comando ssh core@ 10.0.2.xy.

Pare a captura do Wireshark e analise a trama que contém os primeiros dados referentes ao tráfego ssh dirigido ao servidor.

1.1) Anote os endereços MAC de origem e MAC destino da trama capturada. Identifique a que hosts se referem. Justifique.

```
root@Jasmine:/tmp/pycore.33871/Jasmine.conf# ssh core@10.0.2.82
```

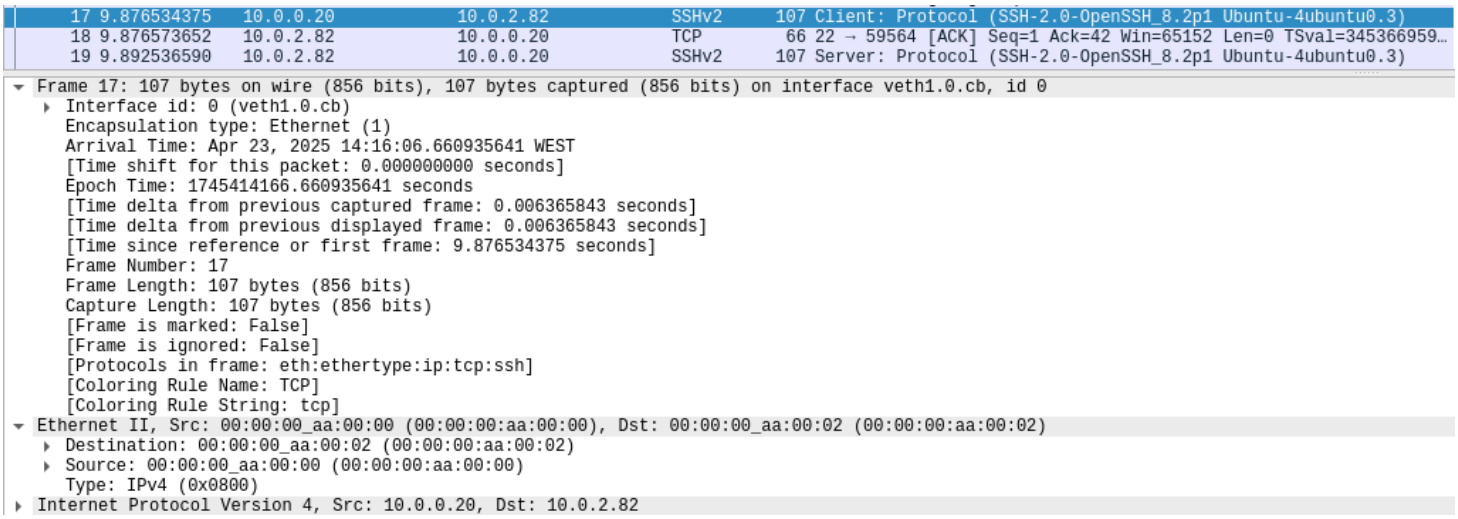
Figura 2: SSH realizado pelo host Jasmine



No.	Time	Source	Destination	Protocol	Length	Info
7	4.635566403	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
8	4.925505350	fe80::a0bc:5fff:feb…	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR…
9	6.636693361	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
10	8.191941214	fe80::200:ff:feaa:0	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:00
11	8.637236237	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
12	9.870022626	00:00:00_aa:00:00	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.20
13	9.870065671	00:00:00_aa:00:02	00:00:00_aa:00:00	ARP	42	10.0.0.1 is at 00:00:00:aa:00:02
14	9.870070170	10.0.0.20	10.0.2.82	TCP	74	59564 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T…
15	9.870154684	10.0.2.82	10.0.0.20	TCP	74	22 → 59564 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA…
16	9.870168532	10.0.0.20	10.0.2.82	TCP	66	59564 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3157618030…
17	9.876534375	10.0.0.20	10.0.2.82	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
18	9.876573652	10.0.2.82	10.0.0.20	TCP	66	22 → 59564 [ACK] Seq=1 Ack=42 Win=65152 Len=0 TSval=345366959…
19	9.892536590	10.0.2.82	10.0.0.20	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
20	9.892549897	10.0.0.20	10.0.2.82	TCP	66	59564 → 22 [ACK] Seq=42 Ack=42 Win=64256 Len=0 TSval=31576180…
21	9.905566780	10.0.0.20	10.0.2.82	TCP	1514	59564 → 22 [ACK] Seq=42 Ack=42 Win=64256 Len=1448 TSval=31576…
22	9.905567472	10.0.0.20	10.0.2.82	SSHv2	130	Client: Key Exchange Init
23	9.905619283	10.0.2.82	10.0.0.20	TCP	66	22 → 59564 [ACK] Seq=42 Ack=1490 Win=64128 Len=0 TSval=345366…
24	9.905621437	10.0.2.82	10.0.0.20	TCP	66	22 → 59564 [ACK] Seq=42 Ack=1554 Win=64128 Len=0 TSval=345366…
25	9.924729414	10.0.2.82	10.0.0.20	SSHv2	1090	Server: Key Exchange Init
26	9.924743605	10.0.0.20	10.0.2.82	TCP	66	59564 → 22 [ACK] Seq=1554 Ack=1066 Win=64128 Len=0 TSval=3157…
27	9.927025130	10.0.0.20	10.0.2.82	SSHv2	114	Client: Diffie-Hellman Key Exchange Init
28	9.927132414	10.0.2.82	10.0.0.20	TCP	66	22 → 59564 [ACK] Seq=1066 Ack=1602 Win=64128 Len=0 TSval=3453…
29	9.936999065	10.0.2.82	10.0.0.20	SSHv2	1182	Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypte…
30	9.937008502	10.0.0.20	10.0.2.82	TCP	66	59564 → 22 [ACK] Seq=1602 Ack=2182 Win=64128 Len=0 TSval=3157…
31	9.937741355	10.0.0.20	10.0.2.82	TCP	66	59564 → 22 [FIN, ACK] Seq=1602 Ack=2182 Win=64128 Len=0 TSval…
32	9.939939278	10.0.2.82	10.0.0.20	TCP	66	22 → 59564 [FIN, ACK] Seq=2182 Ack=1603 Win=64128 Len=0 TSval…
33	9.939948237	10.0.0.20	10.0.2.82	TCP	66	59564 → 22 [ACK] Seq=1603 Ack=2183 Win=64128 Len=0 TSval=3157…
34	10.638396683	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
35	12.638666914	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
36	12.644110426	fe80::200:ff:feaa:2	ff02::5	OSPF	90	Hello Packet
37	14.638926801	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet

Figura 3: Output do Wireshark

Como podemos ver na figura 3, a trama que contem os primeiros dados referentes ao tráfego ssh dirigido ao servidor é a trama 17, como é possível ver pelo protocolo usado, sendo ele, neste caso, o SSHv2.



```
    17 9.876534375   10.0.0.20          10.0.2.82          SSHv2     107 Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
    18 9.876573652   10.0.2.82          10.0.0.20          TCP        66 22 → 59564 [ACK] Seq=1 Ack=42 Win=65152 Len=0 TSval=345366959…
    19 9.892536590   10.0.2.82          10.0.0.20          SSHv2     107 Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
▾ Frame 17: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface veth1.0.cb, id 0
  ▸ Interface id: 0 (veth1.0.cb)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 23, 2025 14:16:06.660935641 WEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1745414166.660935641 seconds
    [Time delta from previous captured frame: 0.006365843 seconds]
    [Time delta from previous displayed frame: 0.006365843 seconds]
    [Time since reference or first frame: 9.876534375 seconds]
    Frame Number: 17
    Frame Length: 107 bytes (856 bits)
    Capture Length: 107 bytes (856 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:ssh]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
▾ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  ▸ Destination: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  ▸ Source: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.2.82
```

Figura 4: Dados da trama 17(1)

Na figura 4, podemos ver os endereços MAC de origem e destino, seno eles:

Origem: IP: 10.0.0.20 , MAC: 00:00:00_aa:00:00 (00:00:00:aa:00:00)

Destino: IP: 10.0.2.82 , MAC: 00:00:00_aa:00:02 (00:00:00:aa:00:02)

Como o host Jasmine tem endereço 10.0.0.20, significa que a trama tem origem em Jasmine, e como o servidor DServer tem endereço 10.0.2.82, significa que a trama tem como destino final o DServer.

Os endereços de origem IP e MAC, neste caso que temos o wireshark ligado na interface da Jasmine, vão corresponder ao mesmo dispositivo, logo o endereço MAC de origem 00:00:00:aa:00:00 corresponde ao host Jasmine. Já ao endereço MAC de destino vai corresponder ao próximo dispositivo no qual a trama vai passar, sendo ele o router R1, assim como é possível ver na topologia da figura 1. Como o endereço MAC de destino vai corresponder ao R1, então 00:00:00:aa:00:02 vai corresponder ao router R1, mais precisamente à interface 10.0.0.1/24 do router R1.

1.2) Qual o valor hexadecimal do campo Type contido no header da trama Ethernet? O que significa? Qual o campo do header IP que tem semântica idêntica?

```
    17 9.876534375   10.0.0.20        10.0.2.82        SSHv2     107 Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
    18 9.876573652   10.0.2.82        10.0.0.20        TCP        66 22 → 59564 [ACK] Seq=1 Ack=42 Win=65152 Len=0 TSval=345366959…
    19 9.892536590   10.0.2.82        10.0.0.20        SSHv2     107 Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
▾ Frame 17: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface veth1.0.cb, id 0
  ▸ Interface id: 0 (veth1.0.cb)
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 23, 2025 14:16:06.660935641 WEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1745414166.660935641 seconds
    [Time delta from previous captured frame: 0.006365843 seconds]
    [Time delta from previous displayed frame: 0.006365843 seconds]
    [Time since reference or first frame: 9.876534375 seconds]
    Frame Number: 17
    Frame Length: 107 bytes (856 bits)
    Capture Length: 107 bytes (856 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:ssh]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
▾ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  ▸ Destination: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  ▸ Source: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.2.82
```

Figura 5: Dados da trama 17(2)

Como podemos ver na figura 5, o valor do hexadecimal do campo Type contido no header da trama Ethernet é 0x0800 e esse valor indica que o protocolo encapsulado na camada de rede(Network layer) é o protocolo IPv4.

Type: IPv4 (0x0800)

```
    17 9.876534375   10.0.0.20        10.0.2.82        SSHv2     107 Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
    18 9.876573652   10.0.2.82        10.0.0.20        TCP        66 22 → 59564 [ACK] Seq=1 Ack=42 Win=65152 Len=0 TSval=345366959…
    19 9.892536590   10.0.2.82        10.0.0.20        SSHv2     107 Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
▸ Frame 17: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface veth1.0.cb, id 0
▾ Ethernet II, Src: 00:00:00_aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  ▸ Destination: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
  ▸ Source: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
    Type: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 10.0.0.20, Dst: 10.0.2.82
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 93
    Identification: 0xc2ac (49836)
  ▸ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x6189 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.0.0.20
    Destination: 10.0.2.82
▸ Transmission Control Protocol, Src Port: 59564, Dst Port: 22, Seq: 1, Ack: 1, Len: 41
▸ SSH Protocol
```

Figura 6: Dados da trama 17(3)

Já na figura 6, somos capazes de ver que o campo do header IP que tem semântica idêntica ao campo Type do header Ethernet é o campo Protocol, onde, neste caso, esse campo está ocupado pelo protocolo TCP, que estará encapsulado no transport layer.

Protocol: TCP (6)

1.3) Quantos bytes são usados no encapsulamento protocolar, i.e., desde o início da trama até ao início dos dados do nível aplicacional? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar.

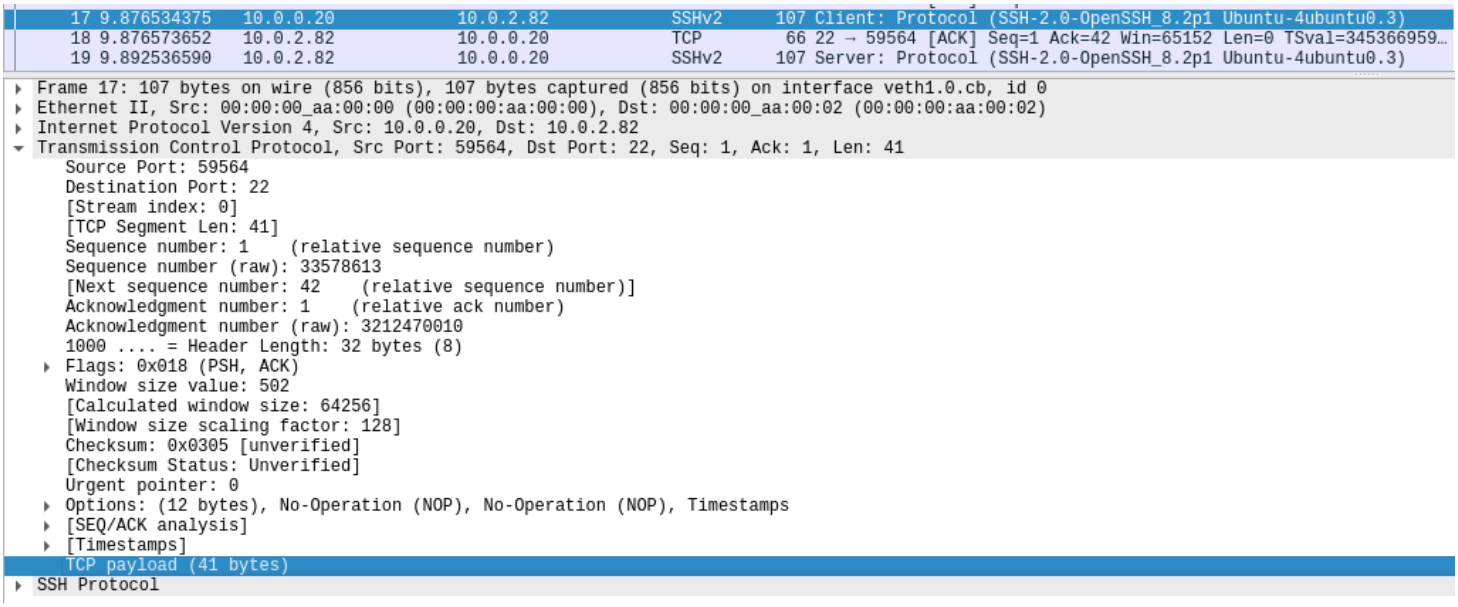


Figura 7: Dados da trama 17(4)

Como é possível ver na figura 7, a trama 17 possui 107 bytes no total e, além disso, possui um TCP payload, que são os dados do nível aplicacional, de 41 bytes. Assim, podemos concluir que são usados 107 - 41 = 66 bytes no encapsulamento protocolar.

Tamanho da trama: **107 bytes**

Payload TCP: **41 bytes**

Encapsulamento protocolar: **107 - 41 = 66 bytes**

Percentagem overhead: **(66/107)x100 = 61.68%**

São usados 66 bytes no encapsulamento protocolar, onde desses 66 bytes fazem parte os headers da link layer(ethernet), network layer(IPv4) e transport layer(TCP).

Por último, a frame tem 107 bytes de tamanho, logo a precentagem de overhead é aproximadamente 61.68%.

A seguir responda às seguintes perguntas, baseado no conteúdo de uma das tramas Ethernet que contém a resposta proveniente do servidor.

1.4) Qual é o endereço MAC da fonte? A que host e interface corresponde? Justifique.

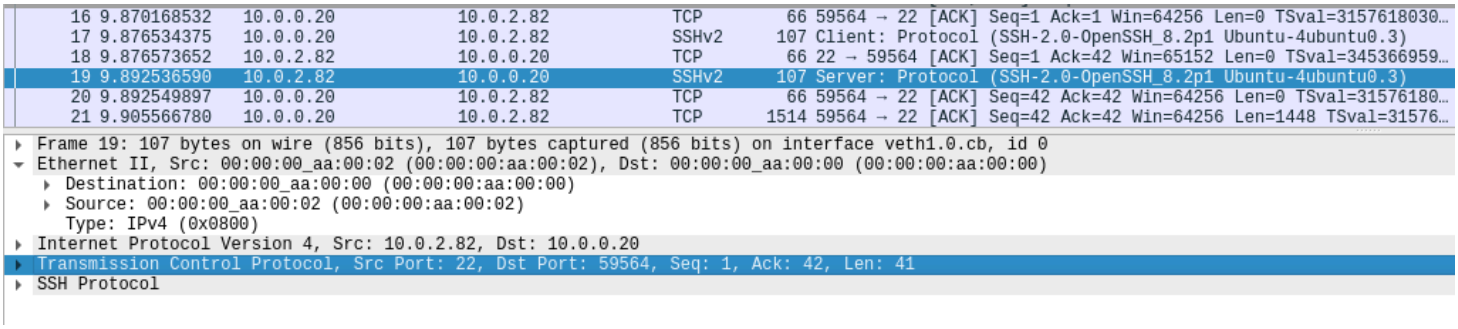


Figura 8: Dados da trama 19

Na figura 8, somos capazes de ver que a trama que representa a resposta do servidor face à trama 17 que foi enviado do host Jasmine para o servidor é a trama 19, visto a própria também possuir o protocolo SSHv2, sendo ela enviada do endereço 10.0.2.82 para o endereço 10.0.0.20, ou seja, a trama 19 foi enviada do servidor DServer. Como estamos a usar o wireshark na interface do host Jasmine, o endereço MAC da fonte vai corresponder ao último dispositivo por onde a trama passou antes de chegar a Jasmine, que neste caso corresponde ao router R1, mais precisamente à interface 10.0.0.1/24.

Concluindo, o endereço MAC 00:00:00:aa:00:02 vai corresponder ao router R1, mais precisamente à interface 10.0.0.1/24.

1.5) Qual é o endereço MAC do destino? A que host e interface corresponde?

Como referi na alínea 1.4, a trama 19, que representa a resposta do servidor, foi enviada do servidor DServer para o endereço 10.0.0.20, que corresponde ao host Jasmine que tem como endereço MAC, MAC: 00:00:00_aa:00:00 (00:00:00:aa:00:00), como é possível ver também na figura 8.

Concluindo, o endereço MAC de destino corresponde ao host Jasmine.

2.2. Exercício 2

Deverá ter a cache ARP completamente vazia antes de iniciar esta secção: reinicie a topologia, ou utilize o comando arp -d.

Comece a capturar tráfego com o Wireshark na interface dos hosts Jasmine, Aladdin, Beauty e Beast. Não sabendo que a Jasmine e a Beauty estavam a capturar tráfego, o Aladdin e o Beast fazem um acesso secreto por ssh para o servidor DServer. Efetue esse acesso e depois pare as várias capturas de tráfego.

2.1) Observe o conteúdo da tabela ARP de Aladdin com o comando arp -a. Com a ajuda do manual ARP (man arp), interprete o significado de cada uma das colunas da tabela.

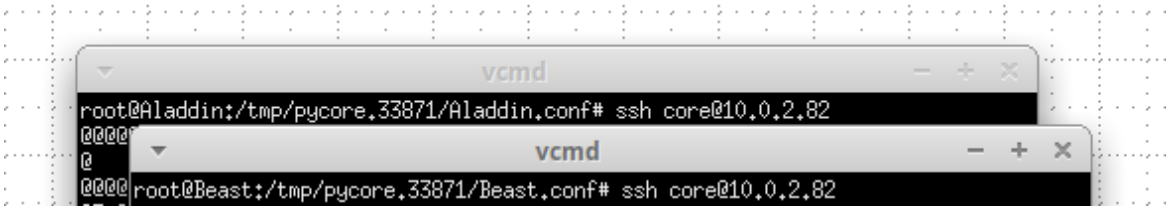


Figura 9: SSH dos hosts Aladdin e Beast para o DServer

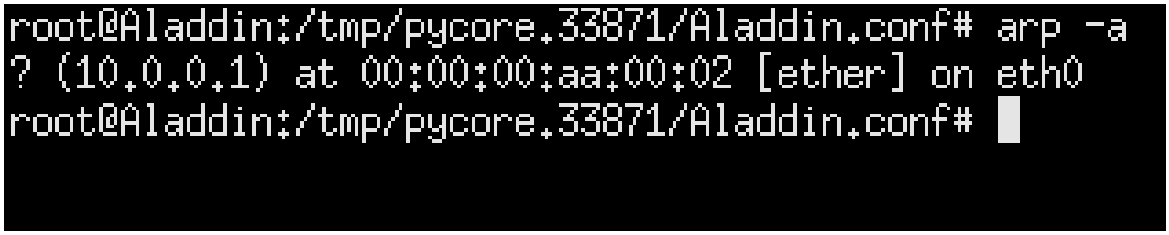


Figura 10: Tabela ARP de Aladdin

A tabela ARP(Adress Resolution Protocol), é responsável por guardar as associações entre endereços IP e endereços MAC. Ela é constituída por quatro colunas:

- IP: Endereço IP do host ao qual se associa o endereço MAC
- MAC: Endereço físico (MAC) do host identificado pelo IP
- Tipo de hardware/tecnologia (normalmente ether para Ethernet)
- Interface: Interface de rede local por onde essa correspondência foi aprendida

Na figura 10, podemos ver que na tabela ARP do host Aladdin os dados presentes em cada coluna são:

- 10.0.0.1
- 00:00:00:aa:00:02
- ether(Ethernet)
- eth0

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

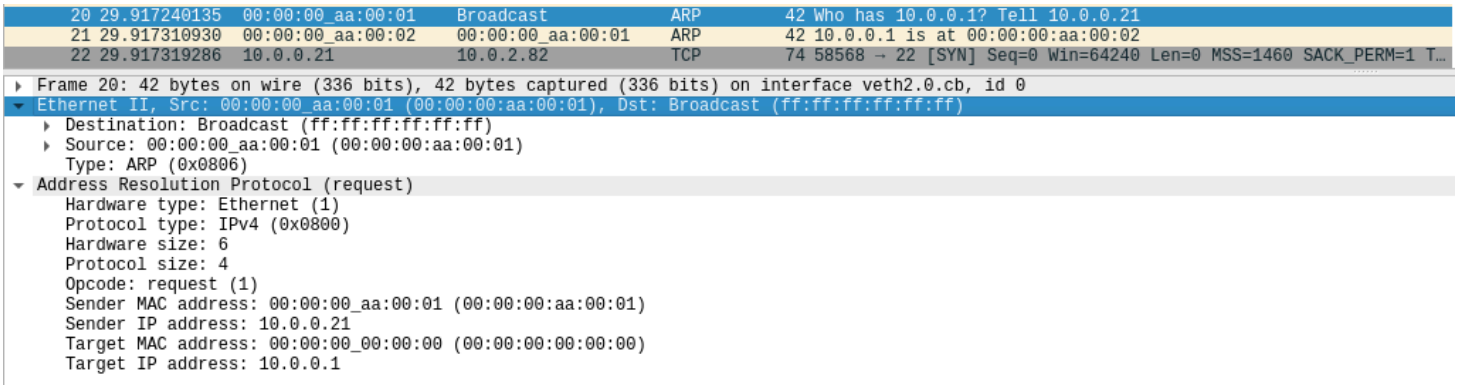


Figura 11: Dados da trama que contém a mensagem com o pedido ARP do host Aladdin

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

Origem: 00:00:00_aa:00:01 (00:00:00:aa:00:01)

Destino: Broadcast (ff:ff:ff:ff:ff:ff)

Na figura 11, o endereço usado como destino no pedido ARP é o endereço broadcast (ff:ff:ff:ff:ff:ff), pois o dispositivo que enviou o pedido não conhece o endereço de destino, então manda para todos.

b) Qual o valor hexadecimal do campo Type da trama Ethernet? O que indica?

Type: ARP (0x0806)

Na figura 11, o valor hexadecimal do campo Type da trama Ehernet representa o protocolo usado na camada superior, neste caso a network layer, ou seja, na network layer é encapsulado o protocolo ARP.

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

Na figura 11, somos capazes de ver que, na secção Adress Resolution Protocol, o campo Opcode está como 1, ou seja, Request, o que indica que se trata efetivamente de um pedido ARP. Além disso, também somos capazes de ver que na caluna Info da tabela com as diferentes tramas, que a trama 20 possui na secção Info a mensagem “Who has 10.0.0.1? Tell 10.0.0.21”, mais uma vez confirmando que se trata de um pedido ARP.

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

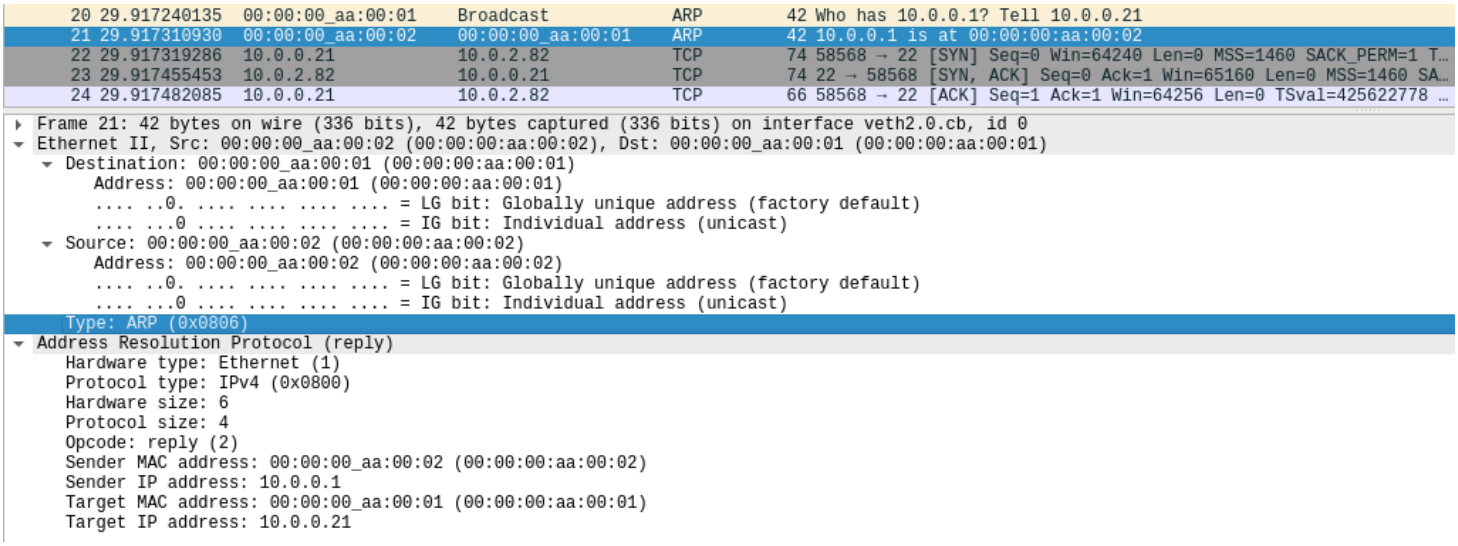


Figura 12: Dados da trama que contém a resposta ao pedido ARP enviado pelo host Aladdin

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

Como podemos ver na figura 12, o valor do campo Opcode é 2, o que significa que se trata de uma resposta(reply).

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

A resposta ao pedido ARP está nos campos:

- Sender MAC address
- Sender IP address

Como é possível ver na figura 12, na trama que estamos a analisar os valores desses dois campos são:

- Sender MAC address: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
- Sender IP address: 10.0.0.1

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 host selecionado (Aladdin).

```
root@Aladdin:/tmp/pycore.33871/Aladdin.conf# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.1                 ether   00:00:00:aa:00:02   C                     eth0
root@Aladdin:/tmp/pycore.33871/Aladdin.conf#
```

Figura 13: Output do comando arp

```
root@Aladdin:/tmp/pycore.33871/Aladdin.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.0.1        0.0.0.0         UG        0 0          0 eth0
10.0.0.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
root@Aladdin:/tmp/pycore.33871/Aladdin.conf#
```

Figura 14: Output do comando netstat -rn

```
root@Aladdin:/tmp/pycore.33871/Aladdin.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.21  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::200:ff:feaa:1  prefixlen 64  scopeid 0x20<link>
        inet6 2001::21  prefixlen 64  scopeid 0x0<global>
        ether 00:00:00:aa:00:01  txqueuelen 1000  (Ethernet)
        RX packets 532  bytes 46853 (46.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 26  bytes 3549 (3.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 648 (648.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 648 (648.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figura 15: Output do comando ifconfig

Como a tabela ARP do Aladdin apenas possui uma entrada referente à correspondência endereço IP/endereço MAC de uma das interfaces do router R1, como é possível identificar pelo endereço IP na topologia da figura 1, isso significa que essa é a informação obtida quando o Aladdin fez o seu ARP request inicial, o que significa que para se conectar ao DServer, o Aladdin teve de passar por essa interface do router R1. Já no ARP reply, a trama enviado pelo DServer para Aladdin deve ter passado pela mesma rota que no ARP request, logo podemos concluir que a trama em causa tem endereço MAC de origem correspondente ao da interface 10.0.0.1/24 do router R1 e endereço MAC de destino correspondente ao endereço MAC do host Jasmine.

Origem: Interface 10.0.0.1/24 do Router R1

Destino: Host Aladdin

d) Discuta, justificando, o modo de comunicação (unicast vs. broadcast) usado no envio da resposta ARP (ARP Reply).

Um ARP request é enviado por broadcast(ff:ff:ff:ff:ff:ff) porque o emissor não conhece o MAC de destino. Já o ARP reply é enviado em unicast, porque agora o remetente conhece o endereço MAC de quem perguntou.

2.4) Verifique se a Jasmine teve conhecimento ou não de todo o tráfego gerado pelo acesso secreto do Aladdin? Qual será a razão para tal?

No.	Time	Source	Destination	Protocol	Length	Info
21	32.880733399	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
22	34.793564250	fe80::200:ff:feaa:2	ff02::5	OSPF	90	Hello Packet
23	34.880877834	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
24	36.883328057	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
25	38.886461681	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
26	39.922681493	00:00:00_aa:00:01	Broadcast	ARP	42	Who has 10.0.0.1? Tell 10.0.0.21
27	39.922726326	00:00:00_aa:00:02	00:00:00_aa:00:01	ARP	42	10.0.0.1 is at 00:00:00:aa:00:02
28	39.922736724	10.0.0.21	10.0.2.82	TCP	74	58568 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T…
29	39.922871014	10.0.2.82	10.0.0.21	TCP	74	22 → 58568 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA…
30	39.922897875	10.0.0.21	10.0.2.82	TCP	66	58568 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=425622778 …
31	39.924776689	10.0.0.21	10.0.2.82	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
32	39.924802427	10.0.2.82	10.0.0.21	TCP	66	22 → 58568 [ACK] Seq=1 Ack=42 Win=65152 Len=0 TSval=231318058…
33	39.971495617	10.0.2.82	10.0.0.21	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
34	39.971519530	10.0.0.21	10.0.2.82	TCP	66	58568 → 22 [ACK] Seq=42 Ack=42 Win=64256 Len=0 TSval=42562282…
35	39.971875188	10.0.0.21	10.0.2.82	TCP	1514	58568 → 22 [ACK] Seq=42 Ack=42 Win=64256 Len=1448 TSval=42562…
36	39.971877373	10.0.0.21	10.0.2.82	SSHv2	130	Client: Key Exchange Init
37	39.971910627	10.0.2.82	10.0.0.21	TCP	66	22 → 58568 [ACK] Seq=42 Ack=1490 Win=63744 Len=0 TSval=231318…
38	39.971922420	10.0.2.82	10.0.0.21	TCP	66	22 → 58568 [ACK] Seq=42 Ack=1554 Win=63744 Len=0 TSval=231318…
39	39.973190188	10.0.2.82	10.0.0.21	SSHv2	1090	Server: Key Exchange Init
40	39.973202292	10.0.0.21	10.0.2.82	TCP	66	58568 → 22 [ACK] Seq=1554 Ack=1066 Win=64128 Len=0 TSval=4256…
41	39.975921023	10.0.0.21	10.0.2.82	SSHv2	114	Client: Diffie-Hellman Key Exchange Init
42	39.975959024	10.0.2.82	10.0.0.21	TCP	66	22 → 58568 [ACK] Seq=1066 Ack=1602 Win=63744 Len=0 TSval=2313…
43	39.985087559	10.0.2.82	10.0.0.21	SSHv2	1182	Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypte…
44	39.985105117	10.0.0.21	10.0.2.82	TCP	66	58568 → 22 [ACK] Seq=1602 Ack=2182 Win=64128 Len=0 TSval=4256…
45	39.986109785	10.0.0.21	10.0.2.82	TCP	66	58568 → 22 [FIN, ACK] Seq=1602 Ack=2182 Win=64128 Len=0 TSval…
46	39.988568216	10.0.2.82	10.0.0.21	TCP	66	22 → 58568 [FIN, ACK] Seq=2182 Ack=1603 Win=64128 Len=0 TSval…
47	39.988585703	10.0.0.21	10.0.2.82	TCP	66	58568 → 22 [ACK] Seq=1603 Ack=2183 Win=64128 Len=0 TSval=4256…
48	40.887217592	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet
49	42.888403747	10.0.0.1	224.0.0.5	OSPF	78	Hello Packet

Figura 16: Output do wireshark no host Jasmine

Na topologia que nos foi apresentada no início da Parte 1 e que também é possível ver na figura 1, é possível observar que os hosts Jasmine e Aladdin estão ligados por um hub. Os hubs são dispositivos de interligação que operam a nível físico, i.e., repetem o sinal que chega através de uma porta de entrada para todas as outras portas, ou seja, eles replicam qualquer trama recebida por uma porta para todas as outras portas.

Como os hosts Jasmine e Aladdin estão ligados por um hub, quando Aladdin faz a ligação ssh para o servidor DServer, Jasmine também vai poder receber o trágego gerado por esse acesso, assim como capturá-lo por wireshark, como se pode observar na figura 16, que mostra as tramas ssh capturadas no wireshark pelo host Jasmine.

2.5) De igual modo, verifique se a Beauty teve conhecimento ou não de todo o tráfego gerado pelo acesso secreto do Beast? Qual será a razão para tal?

Ao contrário dos hosts Jasmine e Aladdin que estão ligados por um hub, os hosts Beauty e Beast estão ligados por um switch, onde diferente do anterior, um switch só envia tramas unicast à porta associada ao endereço MAC de destino, com base na sua tabela de comutação.

Como os hosts Beauty e Beast estão ligados por um switch, quando o Beast faz a ligação ssh ao servidor DServer, Beauty não vai receber o tráfego gerado por esse acesso, visto que a trama vai ser enviada apenas para o DServer.

2.6) Consulte a tabela ARP do Aladdin e do Beast. Que principal diferença entre as tabelas obtidas e que impacto tem no funcionamento da rede?

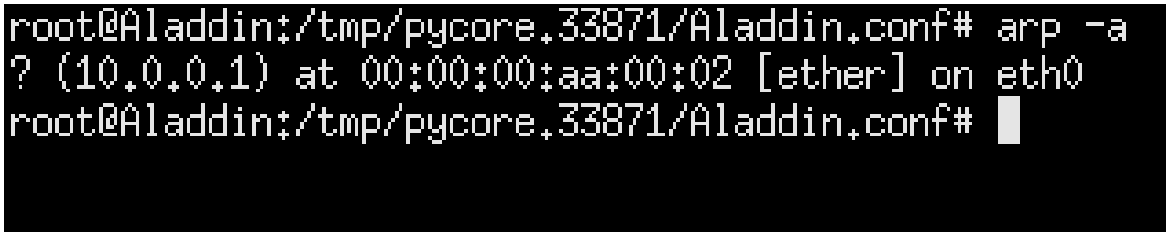


Figura 17: Tabela ARP de Aladdin

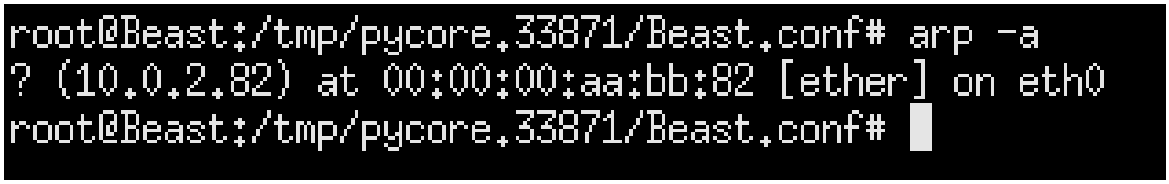


Figura 18: Tabela ARP de Beast

Como podemos ver na figura 17, a tabela ARP do host Aladdin possui uma entrada que faz o endereço IP 10.0.0.1 corresponder ao endereço MAC 00:00:00:aa:00:02, endereço que corresponde a uma das interfaces do router R1 obtida durante o ARP request de Aladdin para fazer um acesso ssh ao servidor DServer.

Já na figura 18, temos a tabela ARP do host Beast que também possui uma entrada que faz corresponder o endereço IP 10.0.2.82 ao endereço MAC 00:00:00:aa:bb:82, endereço que corresponde ao servidor DServer também obtido num ARP rquest realizado por Beast.

A diferença entre as duas tabelas é que na tabela de Aladdin está uma correspondência referente a um dispositivo pela qual a trama vai passar antes de efetivamente chegar ao DServer. Já na tabela do Beast, a entrada que nele existe refere-se ao próprio DServer, o que significa que a trama foi diretamente do Beast para o DServer sem passar por outros dispositivos, à exceção do switch SW1 que apenas transmitiu a trama para a porta correspondente.

2.7) Esboce um diagrama em que ilustre claramente, e de forma cronológica, todo o tráfego layer 2 (tramas) entre o Aladdin e os hosts com os quais comunica, até à receção do primeiro pacote que contém dados do acesso remoto.

O host Aladdin(10.0.2.x) quer comunicar como o DServer(10.0.0.x), e como o DServer está numa rede diferente da dele, ele tem de passar primeiro pelo router R1. Para isso acontecer, ele precisa saber do endereço MAC do R1.

- 1. ARP request enviado pelo Aladdin em Broadcast a perguntar o endereço MAC de R1(10.0.0.1)
- 2. ARP reply enviado por R1 para Aladdin a comunicar o seu endereço MAC

Após estes dois primeiros passos, Aladdin passa a saber o MAC de R1 e já pode se comunicar como o DServer através do R1.

- 3. TCP SYN enviado de Aladdin para DServer para iniciar ligação TCP

- 4. TCP SYN-ACK enviado de DServer para Aladdin a responder que aceita a ligação TCP
- 5. TCP ACK enviado do Aladdin para o DServer a confirmar a receção do TCP SYN-ACK
- 6. Primeiro pacote SSH com dados enviado do DServer para o Aladdin a enviar o primeiro pacote real da comunicação contendo dados SSH

ARP request -> ARP reply -> TCP SYN -> TCP SYN-ACK -> TCP ACK -> SSH Data

2.8) Construa manualmente a tabela de comutação completa do switch da casa da Beauty e do Beast, (SW1) atribuindo números de porta à sua escolha.

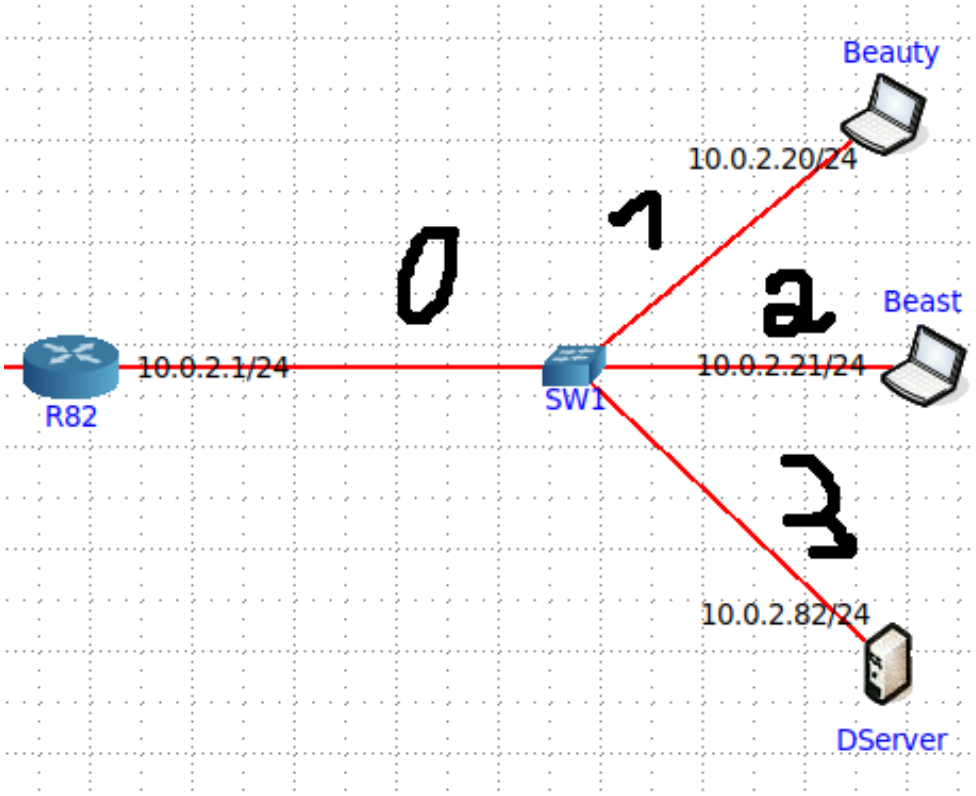


Figura 19: Ilustração dos ports

Como os endereços MAC dos hosts Beast, Beauty e da interface com endereço IP 10.0.2.1/24 do router R82, assim como os hosts Jasmine e Aladdin estavam definidos como auto-assign, atribuimo-lhes endereços MAC específicos, durante a realização deste exercício, para a criação da tabela de comutação de SW1.

- Endereço MAC router R82 na interface 10.0.2.1/24 : 00:00:00:00:00:01
- Endereço MAC do host Beast: 00:00:00:00:00:02
- Endereço MAC do host Beauty: 00:00:00:00:00:03
- Endereço MAC do host Jasmine: 00:00:00:00:00:04
- Endereço MAC do host Aladdin: 00:00:00:00:00:05

Além disso, como já tinhamos definido antes, atribuimos o endereço MAC 00:00:00:AA:BB:82 ao servidor DServer.

MAP adress	**Port**
00:00:00:00:00:01	0
00:00:00:00:00:02	2
00:00:00:00:00:03	1
00:00:00:00:00:04	0
00:00:00:00:00:05	0
00:00:00:AA:BB:82	3

Tabela 1: Tabela de comutação do switch SW1

2.3. Exercício 3

3.1) Como proteção, a Jasmine e o Aladdin, juntamente com a Beauty e o Beast, decidiram conectar R1 e Rxy a uma rede de um ISP com endereços IP públicos, mantendo todo o endereçamento privado das suas LANs. Sabe-se que o ISP não encaminha tráfego para redes privadas, portanto, R1 e Rxy não conseguem encaminhar tráfico para endereços privados remotos, i.e., não fisicamente adjacentes.

Discuta que solução implementaria em R1 e em Rxy de modo a manter todas as funcionalidades anteriormente existentes (conectividade IP, acesso ssh ao servidor, etc.).

De modo a permitir que R1 e R82 mantenham todas as funcionalidades anteriores, ou seja, sejam capazes de manter conetividade entre as duas redes privadas, é possível utilizar a técnica NAT entre R1 e a rede privada contendo os hosts Aladdin e Jasmine, e também entre R82 e os hosts Beauty e Beast e o servidor DServer.

NAT é uma técnica que permite alterar o endereço IP de origem ou destino enquanto eles passam por um router. Através desta técnica seria possível traduzir os endereços IP privados da redes privadas em endereços públicos ao passar num router ou vice-versa, o que permitiria a conetividade entre uma rede pública e privada através desse router. Dentro do âmbito da NAT existe a NAT estática e dinâmica, onde, neste caso, teríamos de usar NAT estática, pois a NAT dinâmica não permitiria conexão iniciada de uma rede pública para uma rede privada, já que na NAT dinâmica a tradução de endereços privados para públicos acontece quando uma conexão é iniciada numa rede privada, o que não permitiria começar a conexão a partir de uma rede pública.

Como pretendemos que seja possível iniciar a conexão de qualquer uma das redes privadas da nossa topologia, não vamos poder usar NAT estática, já que ao passar pela rede de ISP pública para outra rede privada estaríamos a iniciar a conexão numa rede pública. Por isso, para solucionar o nosso problema iremos usar NAT estática alocando estaticamente um endereço público para cada endereço privado, que ficaria então visível para outros dispositivos.

Concluindo, iremos usar a NAT estática em cada um dos routers(R1 e R82) de modo a permitir que haja conexão entre as redes privadas e a rede do ISP com endereços públicos, o que nos permite manter todas as funcionalidades anteriormente existentes.

3. 2º Parte

A Jasmine, como não gosta de ver os cabos da rede Ethernet espalhados pelo palácio, convenceu o Aladdin a substituir a infraestrutura Ethernet por uma rede sem fios. O Aladdin decidiu então comprar equipamento Wi-Fi e fazer uma captura de tráfego para perceber melhor o funcionamento da rede. Descarregue da plataforma de ensino a captura WLAN-traffic-20250407.pcapng.zip e abra o ficheiro .pcapng no Wireshark.

Não se esqueça que deve ser incluída evidência prática que sustente a resposta às questões

3.1. Exercício 1

Como pode ser observado, a sequência de bytes capturada inclui meta-informação do nível físico (radiotap header, radio information) obtida do firmware da interface Wi-Fi, para além dos bytes correspondentes a tramas 802.11.

Selecione a trama de ordem xy correspondente ao seu identificador de grupo (Turno-Grupo, e.g., 27).

1.1) Identifique em que frequência do espectro está a operar a rede sem fios, e o canal que corresponde a essa frequência.

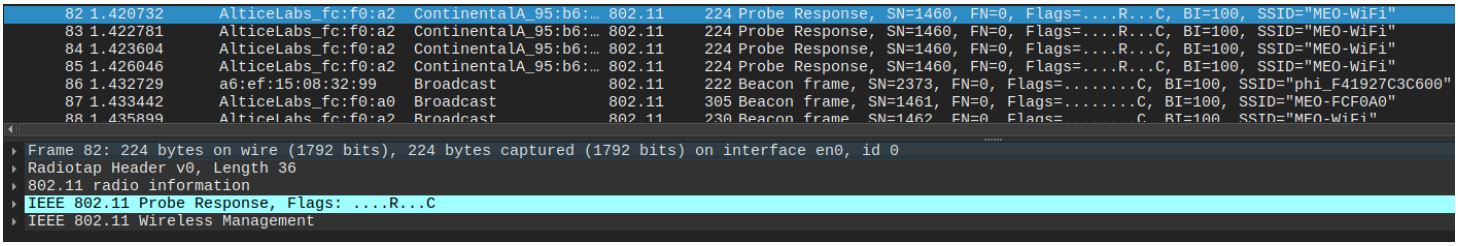


Figura 20: Trama 82 da captura WLAN-traffic-20250407.pcapng.gz

Como fazemos parte do PL82, usámos a trama 82.

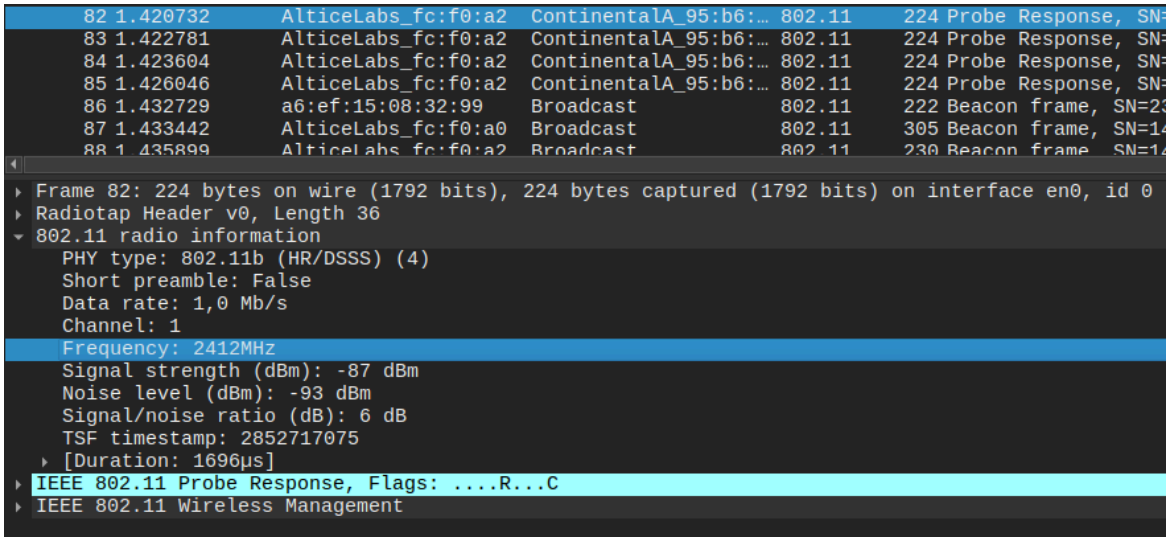


Figura 21: Dados da trama 82(1)

Como é possível ver na figura 21, a rede sem fios está a operar na frequência 2412 MHz, o que corresponde ao canal 1 da banda 2.4 GHz.

Frequência: 2412 MHz

Canal: 1

1.2) Identifique a versão da norma IEEE 802.11 que está a ser usada.

Como podemos observar na figura 21, no subcampo PHY type do campo 802.11 radio information, a versão da norma que está a ser usada é IEEE 802.11b.

PHY type: 802.11b (HR/DSSS) (4)

1.3) Qual a taxa de transmissão a que foi enviada a trama escolhida? Será que essa taxa de transmissão corresponde à máxima que a interface Wi-Fi pode operar? Justifique.

Na figura 21, no subcampo Data rate do campo 802.11 radio information é possível ver que a taxa de transmissão da trama 82 é 1,0 Mb/s. No entanto, a taxa de transmissão máxima que as redes IEEE 802.11b utilizam é de 11 Mb/s.

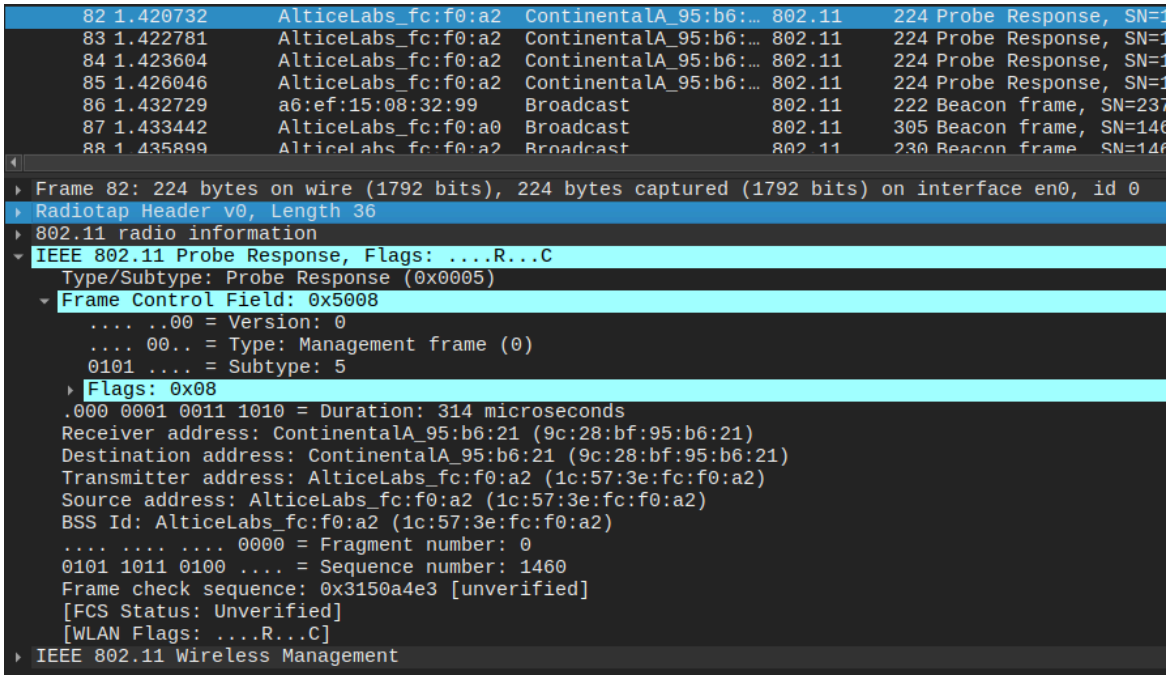


Figura 22: Dados da trama 82(2)

Na figura 22, no campo Type do Frame Control Field é possível ver que a trama 82 trata-se de uma trama de gestão. A diferença entre a velocidade de transmissão em que a trama 82 foi enviada e taxa de transmissão máxima a que a interface wi-fi pode operar deve-se ao facto que as tramas de gestão costumam ter taxas de transmissão menores.

3.2. Exercício 2

Como referido, as tramas beacon permitem efetuar scanning passivo em redes IEEE 802.11 (Wi-Fi). Para a captura de tramas disponibilizada, e considerando xy o seu nº de TurnoGrupo (PLxy), responda às seguintes questões:

2.4) Selecione uma trama beacon cuja ordem (ou terminação) corresponda ao seu ID de grupo. Esta trama pertence a que tipo de tramas 802.11? Identifique o valor dos identificadores de tipo e de subtipo da trama. Em que parte concreta do cabeçalho da trama estão especificados (ver Anexo I)?

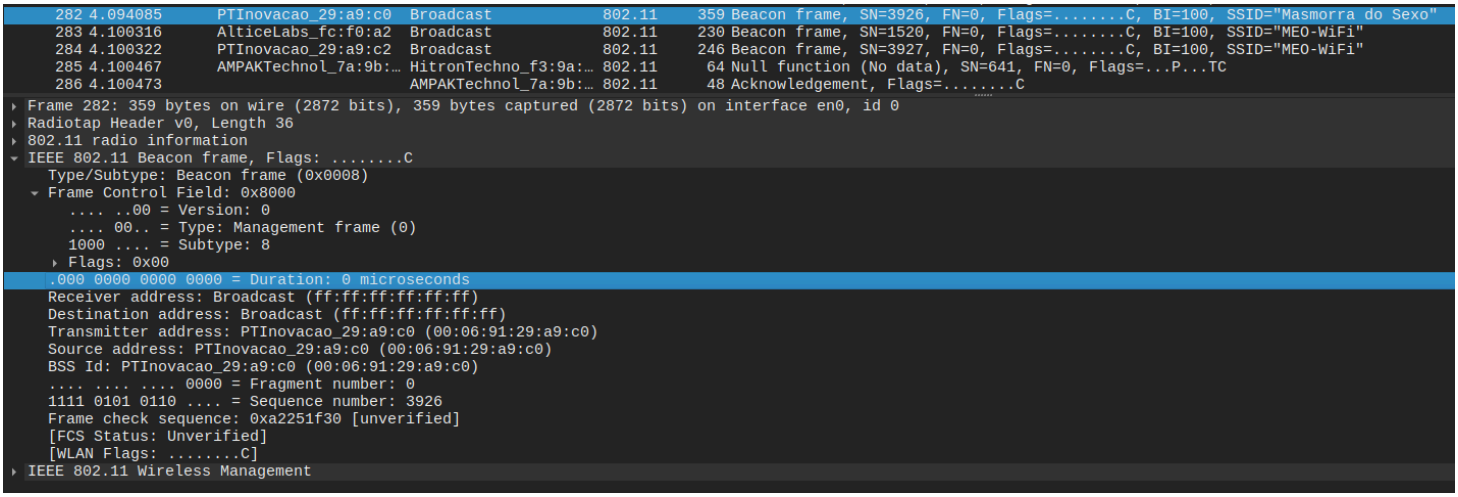


Figura 23: Dados trama 282(1)

Como é possível ver na figura 23, escolhemos a trama 282 que termina com o nosso ID de grupo(82), além disso, ainda na figura 23, podemos ver que se trata de uma trama de gestão e mais especificamente a uma trama Beacon, através dos campos **Type** e **Subtype** do **Frame Control Field**, que está situado no campo **IEEE 802.11 Beacon frame, Flags: ....C**.

.... 00.. = Type: Management frame (0)

1000 .... = Subtype: 8

No campo **Subtype**, o número 8 refere-se a tramas do tipo **Beacon**, e no campo **Type** o número 0 refere-se a uma trama de gestão.

2.5) Verifique se está a ser usado o método de deteção de erros (CRC). Justifique. (Poderá ter de ativar a verificação no Wireshark, em Edit -> Preferences -> Protocols -> IPv4 -> "Validate Checksum if Possible").

Na figura 23, é possível ver que o campo FCS status está como unverified, o que significa que não está a usar o método de deteção de erros (CRC), uma vez que FCS(Frame Check Sequence) é um campo relativo a verificação de erros que usa um algotitmo de CRC(Cyclic Redundancy Check) para detetar se ocorreram erros durante a transmissão da trama. Assim, se o campo FCS está como unverified, significa que não foi feita essa verificação, logo não foi usado o método de deteção de erros (CRC).

2.6) Justifique o porquê de ser necessário usar deteção de erros em redes sem fios.

Existe a necessidade de utilizar métodos de deteção de erros, pois, em redes sem fios, existe uma maior interferência e atenuação do sinal, o que torna este meio mais suscetível a erros, fazendo com que não haja garantia da entrega de tramas sem a presença de erros, ao contrário dos cabos ethernet que são mais estáveis. Devido a estes problemas,

usa-se deteção de erros, como o método CRC, de forma a descobrir tramas corrompidas e evitar o envio de dados inválidos.

2.7) Uma trama beacon anuncia o intervalo entre beacons às várias taxas de transmissão (B) que o AP suporta, assim como várias taxas de transmissão adicionais (extended supported rates). Indique qual a periodicidade e as taxas de transmissão suportadas pelo AP da trama beacon selecionada.

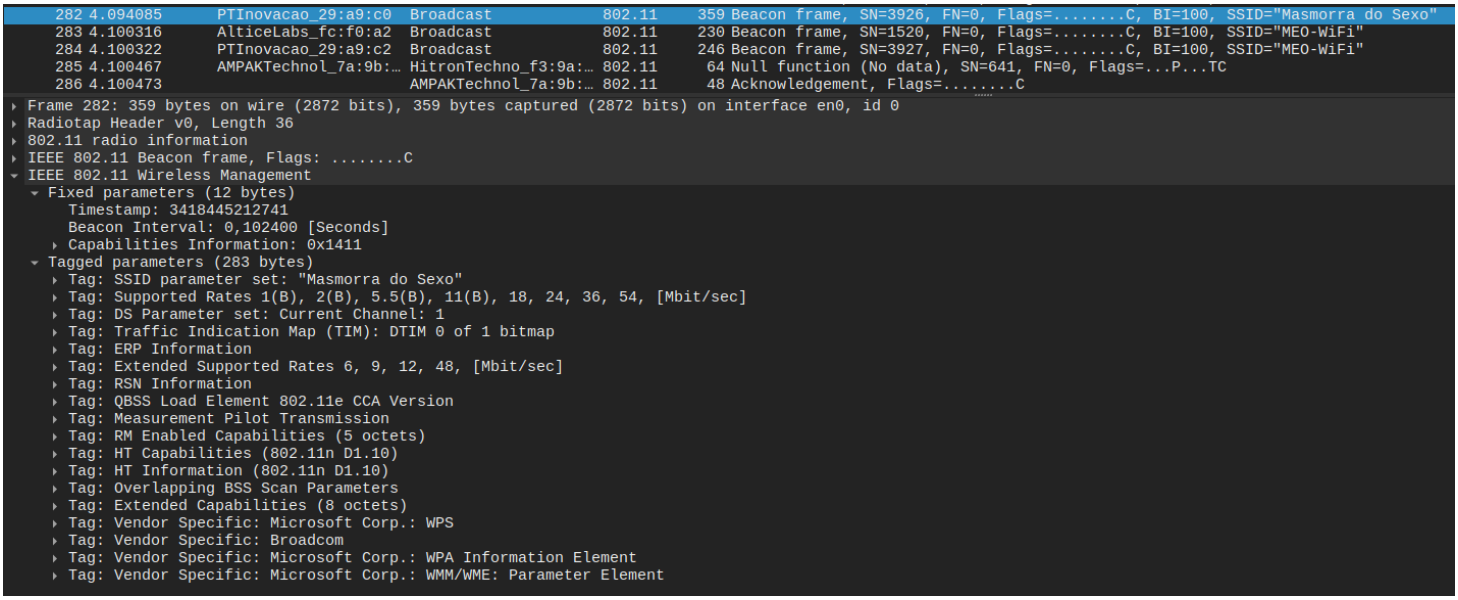


Figura 24: Dados trama 282(2)

Como é possível ver na figura 24, no subcampo **Beacon Interval** pertencente a **Fixed parameters** do campo **IEEE 802.11 Wireless Management** é apresentado a periodicidade da trama beacon selecionada, que, neste caso, é **0,102400** segundos.

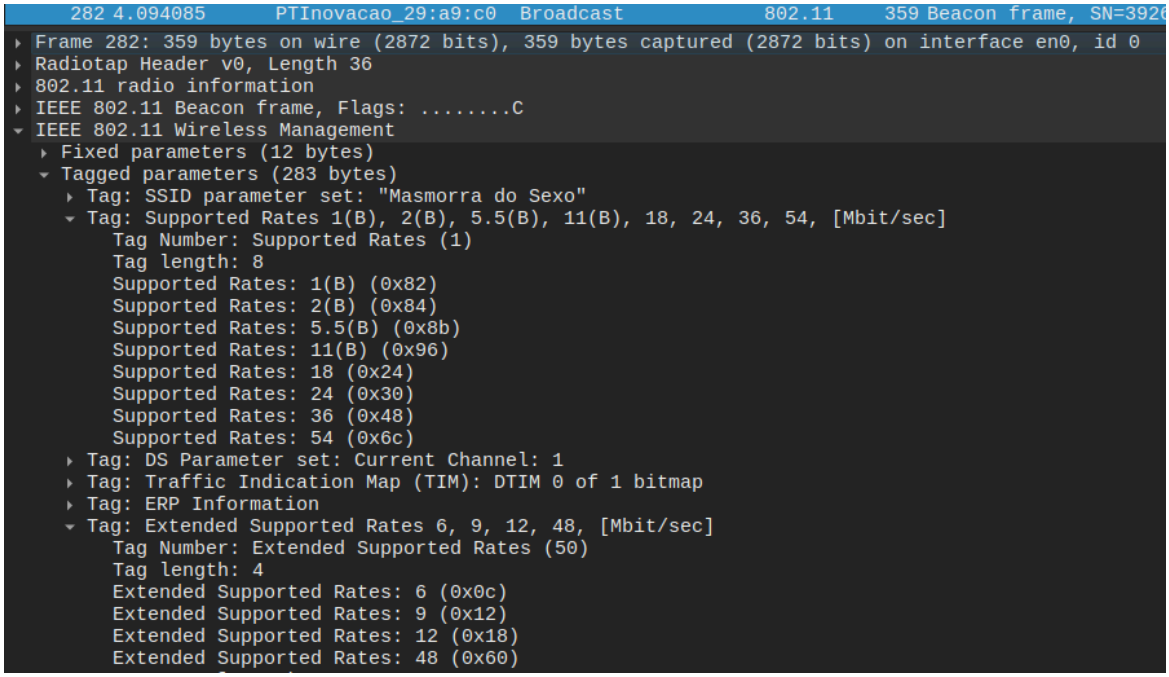


Figura 25: Dados trama 282(3)

Já na imagem 25, somos capazes de ver as taxas de transmissão suportadas pelo AP da trama selecionada no subcampo **Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54, [Mbit/sec]** que pertence ao campo **Tagged parameters** situado em **IEEE 802.11 Wireless Management**.

2.8) Identifique e liste os SSIDs dos APs que estão a operar na vizinhança da STA de captura. Explicite o modo como obteve essa informação (por exemplo, se usou algum filtro para o efeito).

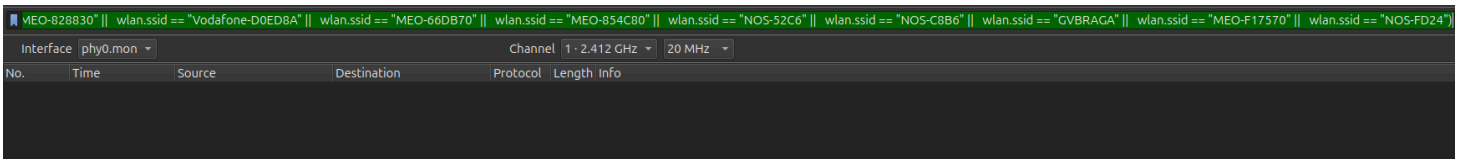


Figura 26: Output do filtro utilizado

Como é possível ver na figura 26, fizemos um filtro que apenas mostra tramas beacon, visto que elas são enviadas pelos APs para anunciar o seu SSID e outras informações. Além disso, conforme íamos encontrando SSIDs íamos adicionando-os ao filtro de modo a não aparecerem, o que nos permitiu descobrir todos os SSIDs da captura disponibilizada conforme adicionávamos os SSIDs ao filtro.

Mais abaixo, é possível ver tanto o filtro utilizado, assim como os SSIDs encontrados.

Filtro wireshark:

(wlan.fc.type_subtype == 8) &&

!(wlan.ssid == “phi_F41927C3C600” ||

wlan.ssid == “MEO-WiFi” ||

wlan.ssid == “FlyingNet” ||

wlan.ssid == “MEO-9BF2A0” ||

wlan.ssid == “NOS-26F6” ||

wlan.ssid == “Masmorra do Sexo” ||

wlan.ssid == “GVBRAGA_EXT” ||

wlan.ssid == “MEO-FCF0A0” ||

wlan.ssid == “GVBRAGA_quarto” ||

wlan.ssid == “NOS-9946_EXT” ||

wlan.ssid == “MEO-828830” ||

wlan.ssid == “Vodafone-D0ED8A” ||

wlan.ssid == “MEO-66DB70” ||

wlan.ssid == “MEO-854C80” ||

wlan.ssid == “NOS-52C6” ||

wlan.ssid == “NOS-C8B6” ||

wlan.ssid == “GVBRAGA” ||

wlan.ssid == “MEO-F17570” ||

wlan.ssid == “NOS-FD24”)

Lista de SSIDs:

“phi_F41927C3C600”

“MEO-WiFi”

“FlyingNet”

“MEO-9BF2A0”

“NOS-26F6”

“Masmorra do Sexo”

“GVBRAGA_EXT”

“MEO-FCF0A0”

“GVBRAGA_quarto”

“NOS-9946_EXT”

"MEO-828830"

"Vodafone-D0ED8A"

"MEO-66DB70"

"MEO-854C80"

"NOS-52C6"

"NOS-C8B6"

"GVBRAGA"

"MEO-F17570"

"NOS-FD24"

2.9) Estabeleça um filtro Wireshark apropriado que lhe permita visualizar todas as tramas probing request e probing response, simultaneamente.

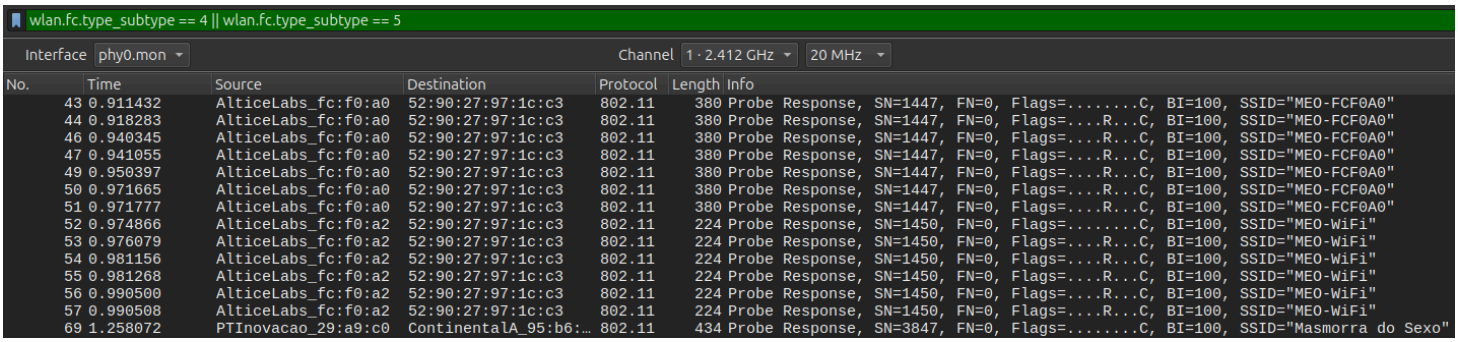


Figura 27: Output do filtro utilizado

Filtro wireshark: **wlan.fc.type_subtype == 4 || wlan.fc.type_subtype == 5**

2.10) Assuma que a STA de captura consegue-se associar a qualquer AP na vizinhança. Dadas as tramas recebidas através do scanning ativo e passivo, observe os valores da força do sinal (Signal Strength) nas meta-informações de nível físico e indique a qual AP a STA de captura se deve associar para obter a melhor qualidade de ligação possível.

Indique como chegou a esta resposta.

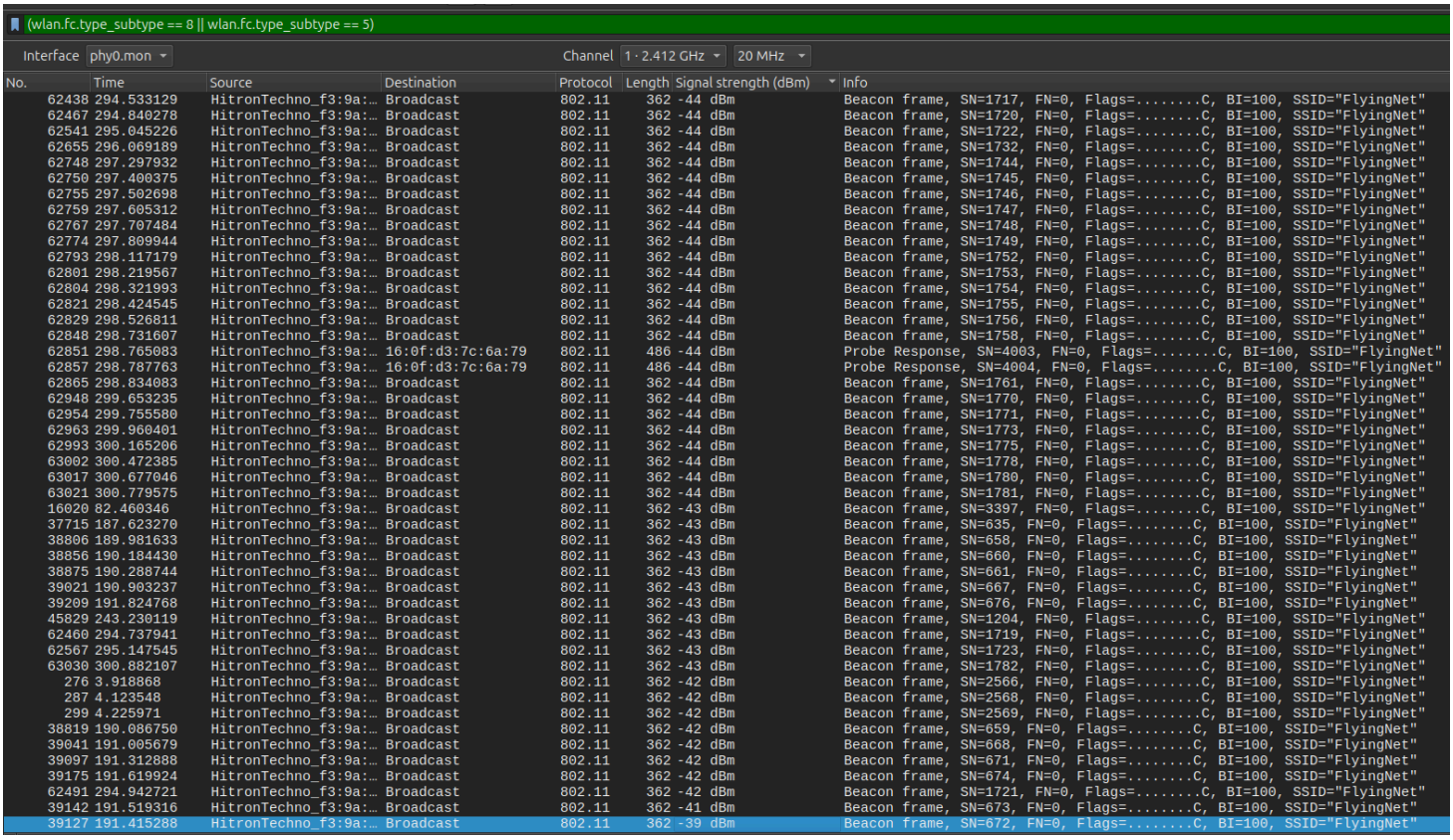


Figura 28: Output do filtro utilizado mais a ordenação por potência de sinal

As tramas recebidas pela STA ao efetuar scanning ativo e passivo são do tipo beacon e prove response, o que nos levou a fazer um filtro de modo a que só elas aparecessem, visto que são as únicas tramas que nos interessam. Por fim, ordenamos as tramas resultantes da aplicação do filtro por ordem do campo **Signal strength(dBm)** para determinar aquela com a melhor potência de sinal, que, neste caso, seria aquela com a unidade dBm mais próxima de 0, como é possível ver na figura 28.

Filtro utilizado : **(wlan.fc.type_subtype == 8 || wlan.fc.type_subtype == 5)**

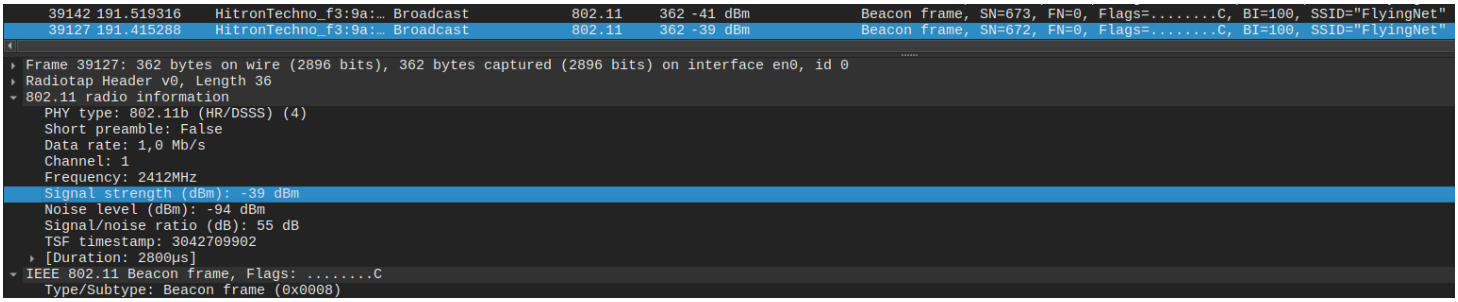


Figura 29: Trama que possui menor potência de sinal

Na figura 29, podemos ver que a trama com melhor potência de sinal é a trama 39127 com -39 dBm, o que significa que, tendo em conta que se trata de uma trama beacon, a AP que enviou a trama 39127 é aquela que a STA de captura se deve associar de modo a obter a melhor qualidade de ligação possível.

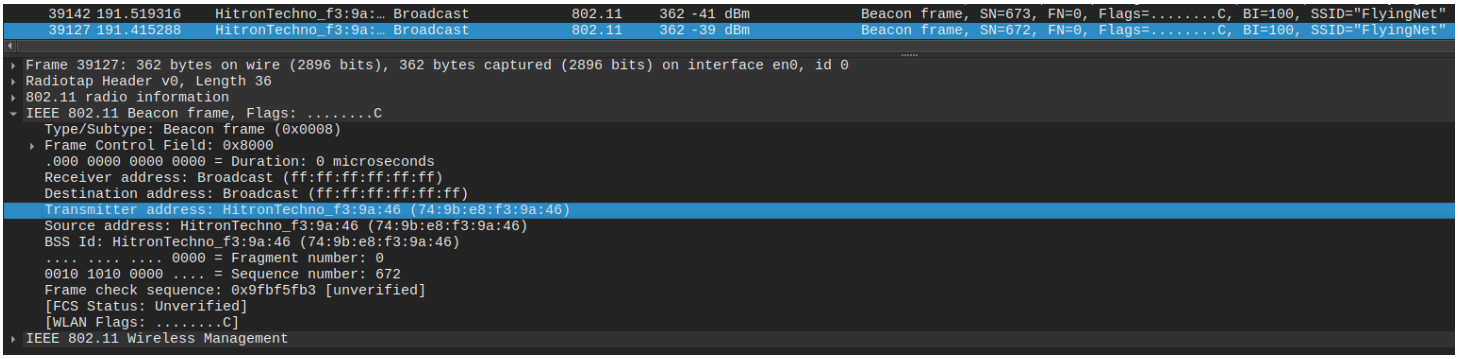


Figura 30: Dados adicionais da trama anterior

Já na figura 30, somos capazes de ver que a AP que enviou a trama 39127 tem o endereço MAC **HitronTechno_f3:9a:46 (74:9b:e8:f3:9a:46)**, como é possível ver no subcampo **Source address** pertencente ao campo **IEEE 802.11 Beacon frame: ....C**, ou seja, a STA de captura deve-se conectar à AP de endereço MAC **HitronTechno_f3:9a:46 (74:9b:e8:f3:9a:46)**.

2.11) Os valores de taxa de transmissão do Wi-Fi estão diretamente associados à qualidade da receção do sinal. Considerando os valores de sensibilidade mínima (Minimum Sensivity) e taxa de transmissão (Data Rate) que constam nas tabelas de referência (ver Anexo II), e a força do sinal recebido nas tramas do AP identificado na resposta anterior, estime o débito que a STA obterá nessa ligação.

Como vimos na alínea anterior, a trama 39127 tem uma força d sinal de -39 dBm, o que é maior que todos os valores de sensibilidade mínima presentes no Anexo II, o que nos permite concluir que o débito que a STA obterá na ligação terá de ser no mínimos igual ao maior Data Rate presente no Anexo II. Por fim, como neste trabalho prático considera-se que os dispositivos IEEE 802.11n utilizam um intervalo de guarda (GI) padrão de 800 ns, podemos estimar que o débito da ligação será no mínimo 65 Mb/s.

3.3. Exercício 3

3.12) Identifique uma sequência de tramas que corresponda a um processo de associação realizado com sucesso entre a STA e o AP, incluindo a fase de autenticação

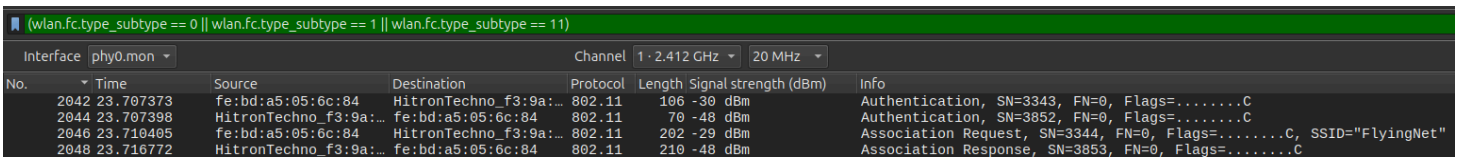


Figura 31: Sequência de tramas que corresponde a um processo de associação realizado com sucesso entre a STA e o AP

Como é possível observar na figura 31, identificamos uma sequência de tramas que corresponde a um processo de associação realizado com sucesso entre a STA e o AP, incluindo a fase de autenticação.

3.13) Efetue um diagrama que ilustre a sequência de todas as tramas trocadas no processo.

1: Authentication (trama enviada pelo STA para o AP)

2: Association request (trama enviado pelo STA para o AP, contendo o pedido de associação)

3: Association response(trama enviado pelo AP para o STA, contendo a resposta ao pedido de associação)

Authentication -> Association request -> Association response

3.4. Exercício 4

4.14) Estabeleça um filtro apropriado e selecione uma trama de dados (Data ou QoS Data), cujo número de ordem inclua o seu identificador de grupo (terminação xy, ou y caso não exista xy). Sabendo que o campo Frame Control contido no cabeçalho das tramas 802.11 permite especificar a direccionalidade das tramas, o que pode concluir face à direccionalidade dessa trama, será local à WLAN?

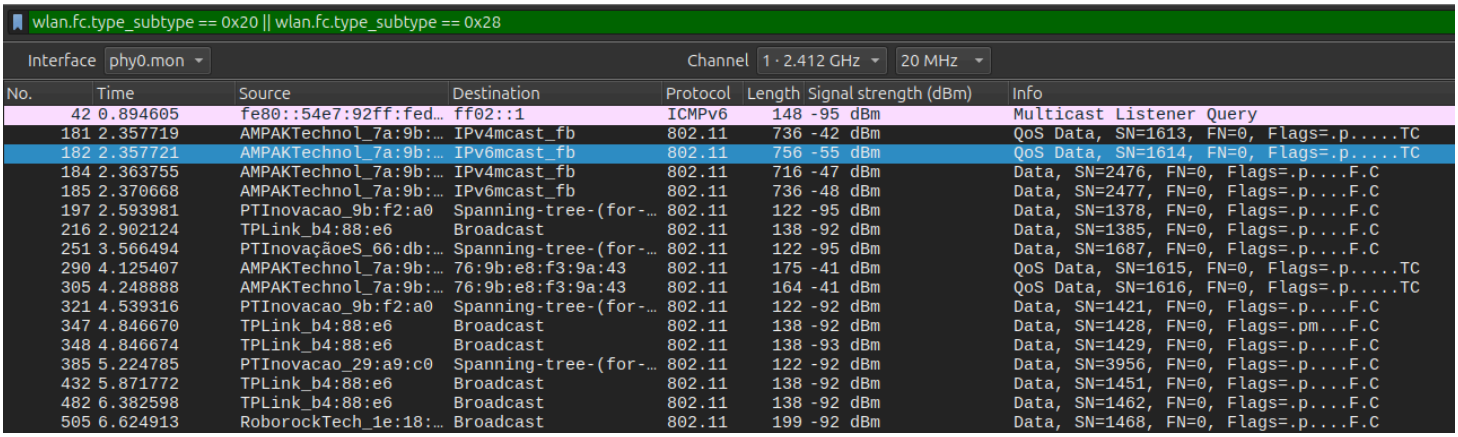


Figura 32: Output do filtro utilizado

Como é possível ver na figura 32, estabelecemos um filtro no wireshark de forma a que apenas aparecessem tramas de dados do tipo Data e QoS Data.

Filtro utilizado: **wlan.fc.type_subtype == 0x20 || wlan.fc.type_subtype == 0x28**

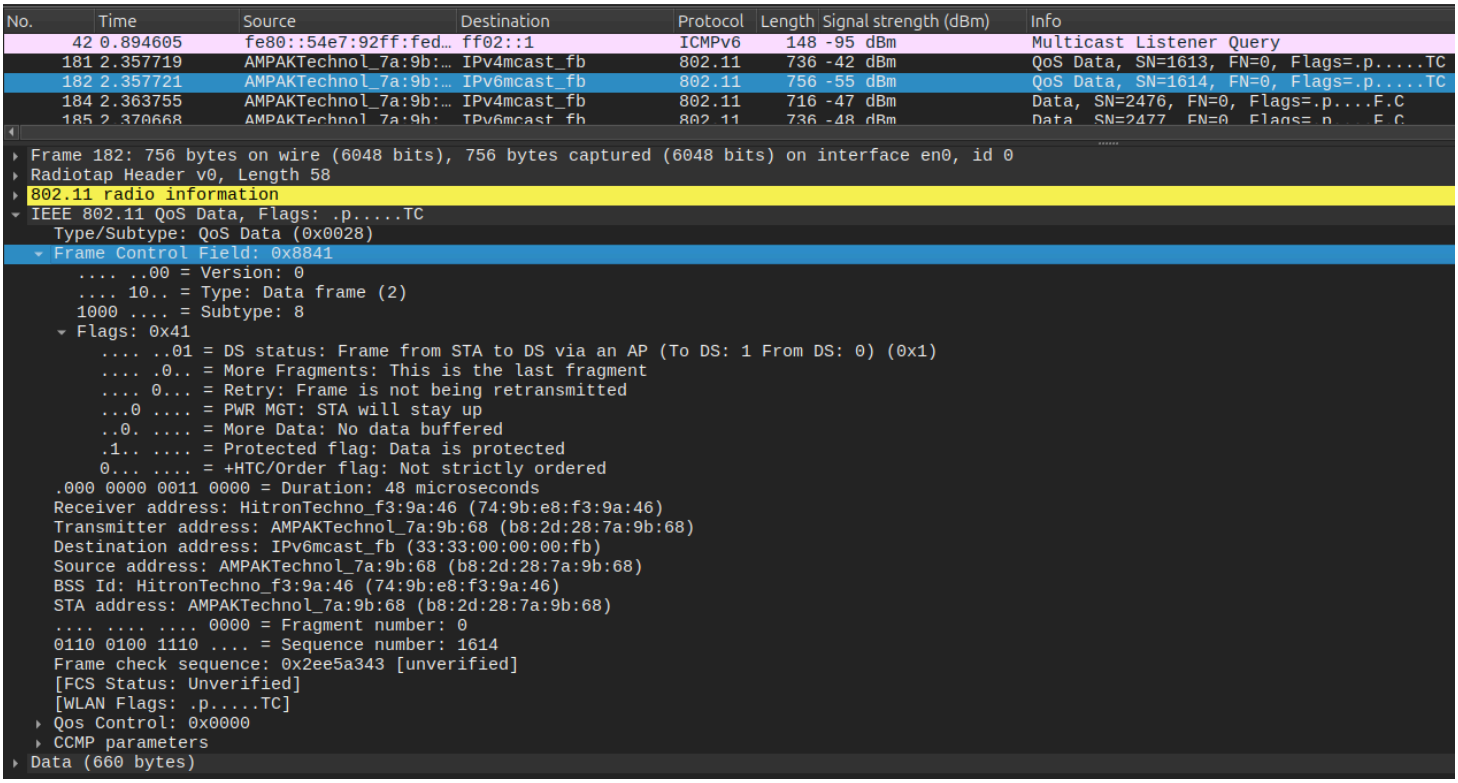


Figura 33: Dados da trama 182

Já na figura 33, é possível observar a trama escolhida, que, neste caso, é a trama 182, visto que os últimos dois dígitos são iguais ao nosso identificador de grupo (PL82).

Além disso, no campo **DS status** , que é o campo que nos permite identificar a direcionalidade da trama, é possível confirmar que o conteúdo do mesmo é **Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x1)**. Como o **To DS** tem valor 1 e o **From DS** tem valor 0, a trama está a ser enviada da STA para um AP.

Por fim, como a transmissão da trama limita-se a ao STA e a um AP exclusivamente, podemos concluir que ela é local à WLAN, visto que ela não está a ser encaminhada entre diferentes APs.

4.15) Para a trama de dados selecionada, transcreva os endereços MAC em uso, identificando quais os endereços correspondentes à estação sem fios (STA), ao AP e ao router de acesso ao sistema de distribuição (DS)?

Usano a trama apresentada na figura 33, fomos capazes de transcrever os seguintes endereços MAC:

- Receiver address: HitronTechno_f3:9a:46 (74:9b:e8:f3:9a:46)
- Transmitter address: AMPAKTechnol_7a:9b:68 (b8:2d:28:7a:9b:68)
- Destination address: IPv6mcast_fb (33:33:00:00:00:fb)
- Source address: AMPAKTechnol_7a:9b:68 (b8:2d:28:7a:9b:68)
- BSS Id: HitronTechno_f3:9a:46 (74:9b:e8:f3:9a:46)

- STA address: AMPAKTechnol_7a:9b:68 (b8:2d:28:7a:9b:68)

Como é possível ver na figura 33, os endereços MAC do AP e do STA são, respetivamente, **HitronTechno_f3:9a:46 (74:9b:e8:f3:9a:46)** e **AMPAKTechnol_7a:9b:68 (b8:2d:28:7a:9b:68)**, como consta nos campos **BSS Id** e **STA address**. Além disso, como o endereço MAC do AP corresponde ao **receiver address** e o endereço MAC da STA corresponde ao **Source address** e ao **Transmitter address**, sobra apenas o **Destination address** que irá corresponder ao router de acesso ao sistema de distribuição (DS), logo o endereço do router de acesso ao DS é **IPv6mcast_fb (33:33:00:00:00:fb)**.

4.16) O uso de tramas Request To Send e Clear To Send, apesar de opcional, é comum para efetuar "pré-reserva" do acesso ao meio quando se pretende enviar tramas de dados, com o intuito de reduzir o número de colisões resultante maioritariamente de STAs escondidas. Para o envio de dados selecionado acima, verifique se está a ser usada a opção RTS/CTS na troca de dados entre a STA e o AP/Router da WLAN, identificando a direccionalidade das tramas e os sistemas envolvidos.

Dê um exemplo de uma transferência de dados em que é usada a opção RTC/CTS e um outro em que não é usada.

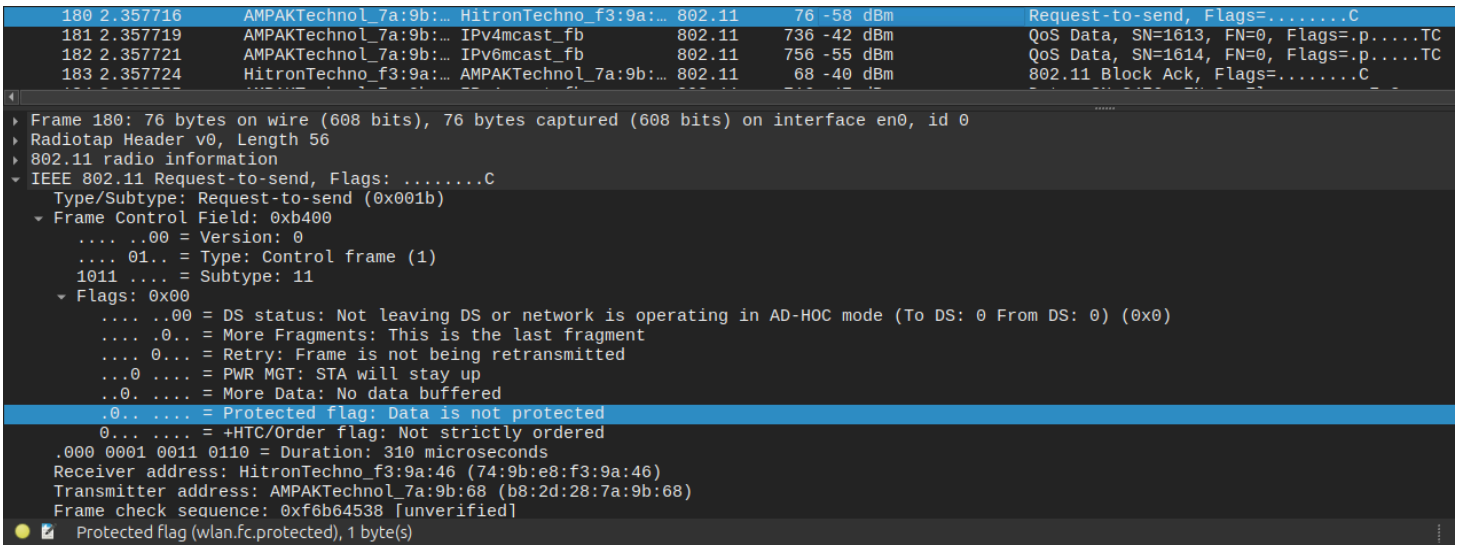


Figura 34: Trama Request to send (RTS)

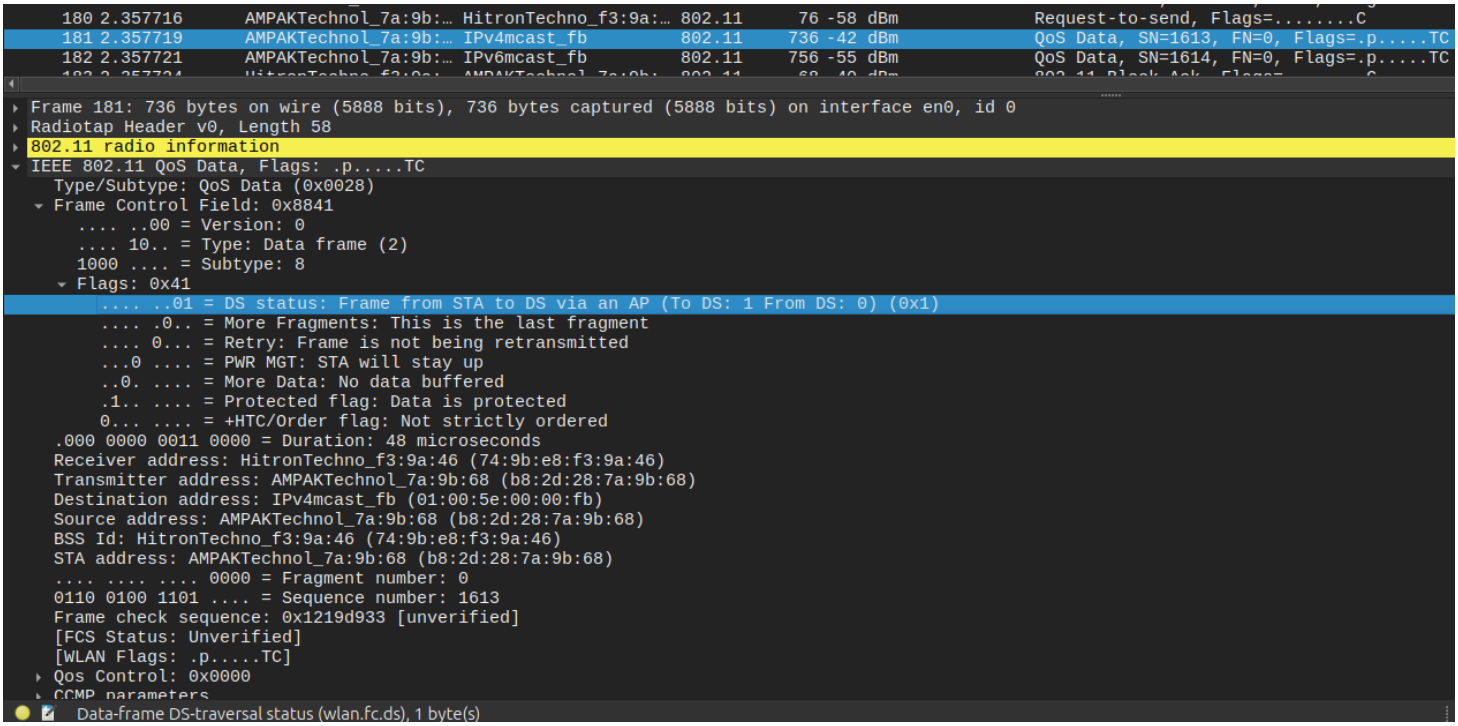


Figura 35: Dados da trama 181

Como é possível ver na figura 34, a trama 180 é uma trama Request to send (RTS) que aparece logo antes das tramas 181 e 182 que enviam dados da STA para o AP, pois tanto na figura 33 como na 35, onde são apresentados os dados das tramas 182 e 182, respetivamente, somos capazes de ver que o campo **To DS** tem valor 1 e o campo **From DS** tem valor 0, o que indica que ambas as tramas estão a ser enviadas da STA para um AP.

A existência de uma trama Request to send (RTS) indica que foi efetuada uma "pré--reserva" de acesso ao AP para o qual as tramas 181 e 182 estão a ser enviadas.

Como já tínhamos visto tanto a trama 182 como a 182 são tramas que são transmitidas da STA diretamente para o AP. Já a trama 180 que se trata de uma trama de controlo do tipo request to send, ela é enviada da STA para o AP para perguntar se pode enviar dados.

O RTS/CTS é usado em situações onde existe um risco elevado de colisões, tramas de grande tamanho e também em tramas encriptadas, e é possível de ver no wireshark através da presença de tramas do tipo Request to send (RTS) e Clear to send (CTS). Já em tramas menores e mais simples, ou quando não existe tráfego para esse determinado AP, o STA envia os dados diretamente sem fazer nenhuma "pré-reserva", e, consequentemente, sem usar a opção RTS/CTS.

4. Conclusão

Neste trabalho, fomos capazes de aprofundar e aplicar o nosso conhecimento em redes ethernet e em redes wi-fi, assim como na camada de ligação lógica, no protocolo ARP e também no protocolo IEEE 802.11.

A execução e realização dos exercícios, no decorrer da realização da parte 1 do projeto, permitiu-nos explorar e abordar mais detalhadamente a composição de tramas ethernet e dos seus diversos campos, assim como o funcionamento das redes locais e dos dispositivos que as interligam, tais como os hubs, dispositivos de nível físico, os switches, dispositivos de nível de ligação lógica, e os routers, dispositivos de nível de rede, além também do funcionamento dos endereços MAC.

Já na parte 2 do projeto, fomos capazes de aprender mais sobre os vários aspetos do protocolo IEEE 802.11 tais como o formato das tramas e alguns dos seus tipos e subtipos mais comuns, como a tramas de gestão, controlo e de dados, além também do processo que um dispositivo(STA) passa para poder se conectar a uma rede wi-fi, desde a realização de scanning passivo e ativo de modo a escolher um ponto de acesso(AP) ao qual se conectar, para a autenticação e associação da STA ao AP, até à transmissão de dados entre ambos e também formas de evitar colisões nessas mesmas transmissões.

Em suma, a partir da realização deste trabalho fomos capazes de aprofundar os nossos conhecimentos no funcionamento das redes locais, mais precisamente no domínio da ethernet, assim como também nas redes sem fio, como é o caso das redes wi-fi, o que nos permitiu ter uma pequena ideia sobre como funciona a área de redes de computadores na realidade.