



Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Redes de Computadores

Ano Letivo de 2024/2025

Trabalho Prático TP3 - Grupo 50

| | | |
|---------------------|---------------------|---------------------|
| João Delgado | Simão Mendes | Nelson Rocha |
| A106836 | A106928 | A106884 |

Maio, 2025

Índice

| | |
|---|-----------|
| 1. Introdução | 1 |
| 1.1 Descrição | 1 |
| 1.2 Motivação | 1 |
| 2. Parte I | 2 |
| 2.1. Problema Geral | 2 |
| 2.2. Captura e análise de Tramas Ethernet | 2 |
| 2.2.1 Questão 1 | 3 |
| 2.2.2 Questão 2 | 3 |
| 2.2.3 Questão 3 | 4 |
| 2.2.4 Questão 4 | 5 |
| 2.2.5 Questão 5 | 5 |
| 2.3. Protocolo ARP e Domínios de Colisão | 6 |
| 2.3.1 Questão 1 | 6 |
| 2.3.2 Questão 2 | 6 |
| 2.3.3 Questão 3 | 8 |
| 2.3.4 Questão 4 | 11 |
| 2.3.5 Questão 5 | 13 |
| 2.3.6 Questão 6 | 14 |
| 2.3.7 Questão 7 | 15 |
| 2.3.8 Questão 8 | 19 |
| 2.4 Serviço de NAT/PAT | 22 |
| 2.4.1 Questão 1 | 22 |
| 3. Parte II | 24 |
| 3.1 Problema Geral | 24 |
| 3.2 Acesso Rádio | 24 |
| 3.2.1 Questão 1 | 24 |
| 3.2.2 Questão 2 | 25 |
| 3.2.3 Questão 3 | 25 |
| 3.3 Scanning Passivo e Scanning Ativo | 26 |
| 3.3.1 Questão 1 | 26 |
| 3.3.2 Questão 2 | 27 |
| 3.3.3 Questão 3 | 27 |
| 3.3.4 Questão 4 | 28 |
| 3.3.5 Questão 5 | 29 |
| 3.3.6 Questão 6 | 31 |
| 3.3.7 Questão 7 | 31 |
| 3.3.8 Questão 8 | 33 |
| 3.4 Processo de Associação | 35 |
| 3.4.1 Questão 1 | 35 |
| 3.4.2 Questão 2 | 36 |
| 3.5 Transferência de Dados | 38 |
| 3.5.1 Questão 1 | 38 |
| 3.5.2 Questão 2 | 39 |
| 3.5.3 Questão 3 | 40 |

| | |
|-----------------------|----|
| 4. Conclusão | 43 |
| 5. Bibliografia | 43 |
| 6. Anexos | 44 |

Lista de Figuras

| | | |
|-----------|--|----|
| Figura 1 | Topologia <i>CORE</i> orquestrada. | 2 |
| Figura 2 | Cabeçalho Ethernet da trama SSH capturada Jasmine -> <i>DServer</i> | 3 |
| Figura 3 | Interface <i>eth0</i> de Jasmine. | 3 |
| Figura 4 | Interface <i>eth0</i> de R1. | 3 |
| Figura 5 | Cabeçalhos da trama capturada. | 4 |
| Figura 6 | Cabeçalho <i>Ethernet</i> da trama SSH capturada <i>DServer</i> -> Jasmine. | 5 |
| Figura 7 | <i>ARP request</i> do Aladdin. | 7 |
| Figura 8 | <i>ARP reply</i> do Aladdin. | 9 |
| Figura 9 | Comandos ifconfig , netstat -rn e arp -a | 10 |
| Figura 10 | Comandos em Jasmine. | 11 |
| Figura 11 | <i>Wireshark</i> em Jasmine. | 12 |
| Figura 12 | Comandos em Beauty. | 13 |
| Figura 13 | <i>Wireshark</i> em Beauty. | 14 |
| Figura 14 | Tabela Arp de Aladdin. | 14 |
| Figura 15 | Tabela Arp de Beast. | 14 |
| Figura 16 | <i>ARP Request</i> de Aladdin. | 15 |
| Figura 17 | <i>ARP Reply</i> de R1. | 16 |
| Figura 18 | Tentativa de envio de frame para <i>DServer 1</i> | 16 |
| Figura 19 | <i>ARP Request</i> entre R1 e R50. | 17 |
| Figura 20 | <i>ARP Reply</i> entre R1 e R50. | 17 |
| Figura 21 | Tentativa de envio de frame para <i>DServer 2</i> | 17 |
| Figura 22 | <i>ARP Request</i> entre R50 e <i>DServer</i> | 18 |
| Figura 23 | <i>ARP Reply</i> entre R50 e <i>DServer</i> | 18 |
| Figura 24 | Envio de <i>frame</i> de Aladdin para <i>DServer</i> | 18 |
| Figura 25 | Resposta de <i>DServer</i> para Aladdin. | 19 |
| Figura 26 | Resultado do comando ifconfig -a no terminal da Beauty | 20 |
| Figura 27 | Resultado do comando ifconfig -a no terminal do Beast | 21 |
| Figura 28 | Resultado do comando ifconfig -a no terminal do router R50 | 21 |
| Figura 29 | Resultado do comando ifconfig -a no terminal do DServer | 22 |
| Figura 30 | Trama de ordem 50 capturada. | 24 |
| Figura 31 | Trama de ordem 50 capturada, no campo 802.11 <i>radio information</i> | 24 |
| Figura 32 | Trama de ordem 50 capturada, no campo 802.11 <i>Wireless Management</i> | 25 |
| Figura 33 | Trama de ordem 50 capturada, no campo <i>Radiotap</i> | 25 |
| Figura 34 | Trama de ordem 50 capturada, no campo 802.11 <i>Wireless Management</i> | 25 |
| Figura 35 | Trama <i>beacon</i> selecionada. | 26 |
| Figura 36 | Trama <i>beacon</i> 89. | 27 |
| Figura 37 | Periodicidade e taxas de transmissão da trama <i>beacon</i> selecionada. | 29 |
| Figura 38 | Cabeçalho 802.11 <i>radio information</i> de <i>frame</i> capturado - Questão 7. | 32 |
| Figura 39 | Cabeçalho 802.11 <i>radio information</i> de <i>frame</i> capturado - Questão 8. | 33 |
| Figura 40 | <i>Supported Rates</i> e <i>HT Information</i> de <i>frame</i> capturado. | 34 |
| Figura 41 | Diagrama de Sequência do Processo de Associação. | 37 |
| Figura 42 | Aplicação de um filtro ao <i>Wireshark</i> para visualizar <i>frames</i> RTS/CTS. | 41 |

Lista de Tabelas

| | | |
|----------|---|----|
| Tabela 1 | Significado dos valores do campo Type no cabeçalho Ethernet. | 8 |
| Tabela 2 | Tabela de Comutação do Switch SW1. | 20 |
| Tabela 3 | Força de sinal dos APs detectados | 32 |
| Tabela 4 | Relação modulação - débito - sensibilidade mínima | 34 |
| Tabela 5 | Tabela Resumo das Tramas. | 36 |
| Tabela 6 | Resumo da frame 1350 para análise. | 39 |
| Tabela 7 | Tabela Resumo das Tramas. | 40 |

Lista de Anexos

| | | |
|---------|---|----|
| Anexo 1 | Formato trama MAC. | 44 |
| Anexo 2 | Formato do campo de controlo de quadro nos PPDU S1G quando os subcampos de Tipo são iguais a 0 ou 2. | 44 |
| Anexo 3 | Combinações válidas de tipo e subtipo. | 45 |
| Anexo 4 | Combinações válidas de tipo e subtipo - Continuação. | 46 |
| Anexo 5 | Parâmetros MCS para 20MHz obrigatórios, $N_{SS} = 1$, $N_{ES} = 1$ | 46 |
| Anexo 6 | Símbolos usados nas tabelas de parâmetros MCS. | 47 |
| Anexo 7 | Sensibilidade mínima do nível de entrada do recetor. | 47 |

1. Introdução

1.1 Descrição

O presente trabalho prático tem como principal objetivo aprofundar os conhecimentos relativos à **camada de ligação lógica**, com especial ênfase nas tecnologias *Ethernet* e *Wi-Fi*, bem como no protocolo **ARP**. Através de atividades práticas de análise de tramas e comportamento da rede, pretende-se compreender o funcionamento das **redes locais com e sem fios**, os mecanismos de **encapsulamento** e **endereçamento**, e a comunicação entre dispositivos dentro de um mesmo domínio de *broadcast*.

1.2 Motivação

Num contexto em que as **redes locais** constituem a base da conectividade em ambientes residenciais, académicos e empresariais, é fundamental perceber os mecanismos subjacentes ao seu funcionamento eficiente e fiável. A análise prática do protocolo **ARP**, das **tramas Ethernet** e dos procedimentos de **acesso ao meio** nas redes *Wi-Fi* permite não só consolidar os conceitos teóricos lecionados nas aulas, como também desenvolver competências essenciais para a identificação e resolução de problemas de rede.

2. Parte I

2.1. Problema Geral

Para iniciar este trabalho foi necessário criar uma topologia na ferramenta de simulação de redes **CORE**. A topologia pode ser descrita do seguinte modo:

1. uma LAN comutada que interliga os *hosts* *Beauty*, *Beast* e o servidor *DServer* (*Disney Server*) através de um *switch* (SW1) ao router de acesso R50;
2. uma LAN partilhada que interliga os *hosts* *Jasmine* e *Aladdin* através de um *hub* ao *router* de acesso (R1);
3. uma rede IP ponto-a-ponto que interliga as duas LANs.

A topologia referida pode ser consultada de seguida:

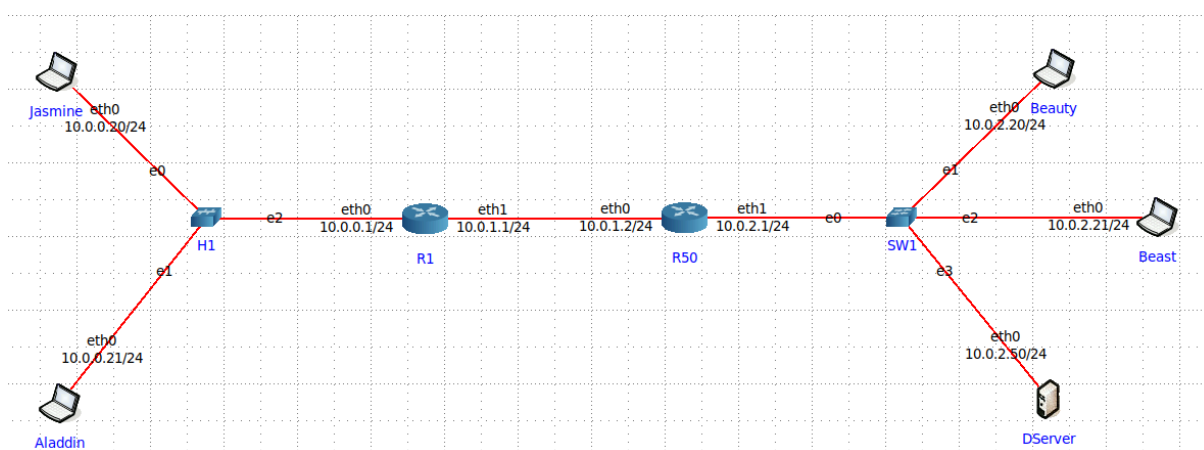


Figura 1: Topologia *CORE* orquestrada.

Na topologia foi particularizado o *router* *Rxy* para R50, sendo 50 o número do nosso grupo de trabalho, bem como o IPv4 e endereço MAC (*Medium Access Control*) do *DServer* para 10.0.2.50 e 00:00:00:AA:BB:50, respetivamente.

É com esta topologia, que segue toda a Parte I deste trabalho, que tem em ênfase o funcionamento dos *MAC addresses*, do protocolo *Ethernet* e do *ARP*.

2.2. Captura e análise de Tramas Ethernet

Após ativarmos a topologia de rede na ferramenta **CORE**, iniciámos o *Wireshark* na interface de saída do *host* *Jasmine*. De seguida, a partir do terminal da *Jasmine*, estabelecemos uma ligação segura ao servidor *DServer* através do seguinte comando:

```
1 ssh core@10.0.2.50
```

bash

Durante esta ligação, capturamos o tráfego SSH gerado. Para responder às primeiras três questões da secção, interrompemos a captura e analisamos a trama que contém os primeiros dados enviados em direção ao servidor. Para as duas restantes questões, incidimos no conteúdo de uma das tramas *Ethernet* que contém a resposta proveniente do *DServer*.

2.2.1 Questão 1

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

Resposta: Após realizar a inspeção da primeira trama SSH gerado entre a Jasmine e o *DServer* encontramos os seguintes endereços MAC de origem e destino:

```
▼ 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)
```

Figura 2: Cabeçalho Ethernet da trama SSH capturada Jasmine -> *DServer*.

Portanto:

- **MAC Origem:** 00:00:00:aa:00:00.
- **MAC Destino:** 00:00:00:aa:00:02.

Com o auxílio da opção do simulador *CORE* para ver a interface de cada dispositivo conectado à rede e com o comando UNIX:

```
1 ip link show <interface>
```

bash

Pode-se verificar o endereço MAC associado a cada interface de um dispositivo. A primeira, e mais óbvia, a verificar foi a interface *eth0* de Jasmine:

```
root@Jasmine:/tmp/pycore.42165/Jasmine.conf# ip link show eth0
26: eth0@if27: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
mode DEFAULT group default qlen 1000
    link/ether 00:00:00:aa:00:00 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

Figura 3: Interface *eth0* de Jasmine.

O que coincidiu com o endereço MAC de origem da trama capturado. Mais surpreendente, talvez, foi constatar que o endereço MAC de destino não pertence ao *DServer*, mas sim à interface *eth0* do *router* R1, o qual se encontra diretamente ligado à rede local da Jasmine:

```
root@R1:/tmp/pycore.42165/R1.conf# ip link show eth0
30: eth0@if31: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
mode DEFAULT group default qlen 1000
    link/ether 00:00:00:aa:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

Figura 4: Interface *eth0* de R1.

Esta situação justifica-se pelo facto de o *DServer* não pertencer à mesma sub-rede IP da Jasmine. Como tal, ao nível da camada de rede, a Jasmine reconhece que a trama tem como destino uma rede remota e, por isso, entrega-o ao seu *default gateway*, neste caso o *router* R1. A camada de ligação lógica, por sua vez, encapsula o datagrama IP numa trama *Ethernet* cujo destino é o endereço MAC do *router*, e não do *DServer*. Só posteriormente, ao nível do *router*, é feita uma nova alteração do endereço MAC de destino e reencaminhamento em direcção ao *DServer*.

Portanto, de forma geral, pode-se concluir que a camada de ligação lógica (*layer 2*) opera apenas dentro dos limites de uma sub-rede, sendo responsável pela comunicação entre nós diretamente ligados à mesma rede física.

2.2.2 Questão 2

Problema: 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?

Resposta: Conforme observado na Figura 2, o valor hexadecimal do campo *Type* presente no cabeçalho da trama *Ethernet* é 0x0800. Este valor indica que o conteúdo encapsulado na trama corresponde a um datagrama do protocolo **IPv4**, de acordo com os valores definidos na especificação *EtherType*, que pode ser consultada em [1].

Este campo tem como função indicar qual o protocolo da camada superior que está a ser transportado pela trama *Ethernet*. Neste caso, 0x0800 identifica o protocolo **Internet Protocol Version 4**.

No cabeçalho IP, o campo com semântica equivalente é o campo **Protocol**, que indica qual o protocolo da camada de transporte. Conclui-se, portanto, que o campo *Type* da trama *Ethernet* e o campo *Protocol* do cabeçalho IP têm funções análogas, pois ambos indicam qual o protocolo da camada superior que deve ser usado para interpretar os dados transportados.

2.2.3 Questão 3

Problema: 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.

Resposta: Foi analisada uma versão completa da trama capturada inicialmente:

```
▶ Frame 9: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface veth1.0.11, 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: 10.0.2.50
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 93
  Identification: 0x0f04 (3844)
  ▶ Flags: 0x4000, Don't fragment
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0x1552 [validation disabled]
  [Header checksum status: Unverified]
  Source: 10.0.0.20
  Destination: 10.0.2.50
▼ Transmission Control Protocol, Src Port: 39416, Dst Port: 22, Seq: 1, Ack: 1, Len: 41
  Source Port: 39416
  Destination Port: 22
  [Stream index: 0]
  [TCP Segment Len: 41]
  Sequence number: 1 (relative sequence number)
  Sequence number (raw): 4050581782
  [Next sequence number: 42 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  Acknowledgment number (raw): 2629482338
  1000 .... = Header Length: 32 bytes (8)
```

Figura 5: Cabeçalhos da trama capturada.

A análise da mesma revelou os seguintes tamanhos de cabeçalho, por camada:

- **Cabeçalho IPv4:** 20 *bytes*.
- **Cabeçalho TCP:** 32 *bytes*.
- **Tamanho total trama:** 107 *bytes*.

Como o *Wireshark* não representa o tamanho do cabeçalho *Ethernet* foi necessário consultar o livro de referência para a unidade curricular [2], no qual encontramos que o tamanho adicionado pelo cabeçalho *Ethernet* é de 14 *bytes*.

Somando os três cabeçalhos, obtemos um total de $14 + 20 + 32 = 66$ *bytes* de *overhead* antes do início dos dados da camada de aplicação.

Sabendo que o tamanho total da trama capturada é de 107 bytes, podemos calcular a sobrecarga introduzida pela pilha protocolar pela expressão:

$$\frac{66}{107} \cdot 100 \approx 61.68\%$$

Conclui-se, assim, que aproximadamente **61.68%** da trama corresponde a sobrecarga associada ao encapsulamento dos dados através das várias camadas da pilha protocolar.

2.2.4 Questão 4

Problema: Qual é o endereço MAC da fonte? A que *host* e interface corresponde? Justifique.

Resposta: À semelhança da primeira questão analisou-se o primeira trama SSH, só que desta vez da resposta proveniente do servidor. A trama pode ser vista de seguida:

```
▼ Ethernet II, Src: 00:00:00_aa:00:02 (00:00:00:aa:00:02), Dst: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
  ▶ Destination: 00:00:00_aa:00:00 (00:00:00:aa:00:00)
  ▶ Source: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
    Type: IPv4 (0x0800)
```

Figura 6: Cabeçalho *Ethernet* da trama SSH capturada *DServer* -> Jasmine.

Assim sendo,

- **MAC Origem:** 00:00:00:aa:00:02.

Como já sabemos da questão 1, este endereço MAC pertence ao dispositivo R1, e corresponde à interface eth0, o que pode ser constatado pela figura 4 anteriormente apresentada. O importante a saber de momento, é o porquê do endereço MAC de origem ser de R1 e não de *DServer*, afinal a trama é proveniente do mesmo.

Para responder a esta questão, é essencial ter novamente em consideração a área de atuação da camada de ligação lógica, completamente local, que trata da comunicação entre dispositivos diretamente conectados. Mesmo que o servidor (*DServer*) tenha gerado a resposta, a trama é encaminhada através da rede, sendo encaminhada por R1, o *router* que conecta a rede do servidor à rede de Jasmine. Quando a trama é encaminhada, a **camada de Data Link no R1** substitui o endereço MAC de origem pelo do próprio *router*, visto que ele é o responsável por encaminhar a resposta para o destino final. Além disso, o *Wireshark*, quando aplicado na interface de Jasmine, só permite capturar as tramas que circulam dentro da rede local em que Jasmine se encontra e que, de alguma forma, interagem com o dispositivo. Isso significa que o *Wireshark* pode ver tramas enviadas para ou recebidas por Jasmine, mas **não pode capturar tramas que não transitam pela sua rede local** ou que estão fora do alcance da sua interface de rede.

Portanto, o **endereço MAC de origem** 00:00:00:aa:00:02 corresponde ao R1, na interface , que foi responsável por encaminhar a resposta de *DServer* para o *host* Jasmine. Este comportamento deve-se ao facto de que a camada de *Data Link* funciona localmente e utiliza o endereço MAC da interface de saída do *router* para enviar tramas entre redes diferentes.

2.2.5 Questão 5

Problema: Qual é o endereço MAC do destino? A que *host* e interface corresponde?

Resposta: Invocando novamente a figura 6, é possível verificar que:

- **MAC Destino:** 00:00:00:aa:00:00.

Portanto, atendendo aos dados obtidos durante a resolução da questão 1, sabe-se que este endereço MAC corresponde ao *host* Jasmine na interface eth0, disposta na Figura 3.

2.3. Protocolo ARP e Domínios de Colisão

Antes de iniciar esta parte, garantimos que a cache ARP estava completamente limpa em todos os *hosts*, reiniciando a topologia e recorrendo ao comando:

```
1 arp -d
```

bash

De seguida, iniciámos a captura de tráfego com o *Wireshark* nas interfaces de rede dos *hosts* Jasmine, Aladdin, Beauty e Beast.

Sem que os *hosts* Aladdin e Beast soubessem que estavam a ser monitorizados, procedemos à realização de uma ligação SSH (*Secure Shell*) “secreta” a partir desses dois *hosts* para o servidor *DServer*.

Após estabelecida a ligação e capturado o tráfego inicial da comunicação, interrompemos todas as capturas nos quatro *hosts*.

2.3.1 Questão 1

Problema: 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.

Resposta: Após executar o comando `arp -a` na máquina *Aladdin*, obteve-se a seguinte linha na tabela ARP:

```
1 ? (10.0.0.1) at 00:00:00:aa:00:02 [ether] on eth0
```

bash

Esta entrada contém a seguinte informação:

- **? (10.0.0.1):** O símbolo ? indica que o nome do *host* associado ao IP 10.0.0.1 não foi resolvido (por **DNS** ou ficheiros locais como o `/etc/hosts`). O IP 10.0.0.1 corresponde, neste caso, ao *gateway* da rede, mais concretamente à interface do *router R1*.
- **at 00 : 00 : 00 : aa : 00 : 02:** Indica o endereço físico (MAC) associado ao IP 10.0.0.1. Este MAC foi obtido através do protocolo ARP quando a máquina *Aladdin* tentou comunicar com um IP fora da sua sub-rede, tendo como próximo salto o *router*.
- **[ether]:** Especifica que esta entrada ARP corresponde a uma rede *Ethernet*.
- **on eth0:** Mostra que esta informação é válida para a interface de rede eth0, que é a utilizada pela máquina *Aladdin*.

Em resumo, esta linha da tabela ARP mostra que, ao tentar comunicar com um destino fora da rede local (como o **DServer** em 10.0.2.50), o *host Aladdin* utilizou o protocolo ARP para descobrir o endereço MAC do seu *gateway* (10.0.0.1), permitindo assim o envio correto dos pacotes IP via camada de ligação.

2.3.2 Questão 2

Problema: 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?
- **b.** Qual o valor hexadecimal do campo *Type* da trama *Ethernet*? O que indica?

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

Resposta: A partir das capturas de tráfego realizadas nos *hosts* pedidos, foi possível observar algumas tramas *Ethernet* contendo um *ARP request*. Abaixo apresenta-se a análise detalhada desta trama, de acordo com os pontos solicitados.

Para fornecer contexto, apresentamos a seguir uma captura de ecrã dos detalhes do *ARP request* do *host* Aladdin, capturada utilizando o *Wireshark*:

```

▼ Frame 27: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth2.0.8a, id 0
  ▼ Interface id: 0 (veth2.0.8a)
    Interface name: veth2.0.8a
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 14, 2025 11:40:44.617609048 WEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1744627244.617609048 seconds
    [Time delta from previous captured frame: 1.866712617 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 29.897479063 seconds]
    Frame Number: 27
    Frame Length: 42 bytes (336 bits)
    Capture Length: 42 bytes (336 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
  ▼ Ethernet II, Src: 00:00:00_aa:00:01 (00:00:00:aa:00:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    ▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      Address: Broadcast (ff:ff:ff:ff:ff:ff)
      ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
      ....1. .... = IG bit: Group address (multicast/broadcast)
    ▼ Source: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
      Address: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
      ....0. .... = LG bit: Globally unique address (factory default)
      ....0. .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
  ▼ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
    Sender IP address: 10.0.0.21
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.0.1

```

Figura 7: *ARP request* do Aladdin.

a. Valor hexadecimal dos endereços MAC origem e destino. Interpretação do endereço de destino utilizado:

Endereço MAC de origem: 00:00:00:aa:00:01 .

Este é o endereço MAC da interface de rede do *host* Aladdin, responsável pelo envio da trama.

Endereço MAC de destino: ff:ff:ff:ff:ff:ff .

Este endereço é conhecido como *broadcast Ethernet*. É utilizado quando o emissor da mensagem não conhece ainda o endereço MAC correspondente ao endereço IP de destino (neste caso, o IP 10.0.0.1, o *gateway* da rede local).

Justificação do uso de *broadcast*:

No contexto de um pedido ARP (*ARP Request*), o emissor precisa descobrir qual é o endereço MAC associado a um determinado endereço IP. Como essa informação ainda não é conhecida, a mensagem é enviada para todos os dispositivos da sub-rede. Apenas o *host* com o IP correspondente responderá ao pedido. Este comportamento é consistente com o funcionamento esperado do protocolo ARP.

b. Valor hexadecimal do campo *Type* da trama *Ethernet* e sua interpretação:

Valor do campo *Type*: 0x0806 .

Este campo da trama *Ethernet* indica qual é o protocolo encapsulado no *payload* útil da trama. O valor hexadecimal 0x0806 corresponde ao protocolo ARP (**A**ddress **R**esolution **P**rotocol) [1].

| Valor <i>Type</i> | Protocolo |
|-------------------|-----------|
| 0x0800 | IPv4 |
| 0x0806 | ARP |
| 0x86DD | IPv6 |

Tabela 1: Significado dos valores do campo *Type* no cabeçalho *Ethernet*.

Assim, este valor confirma que a trama transporta uma mensagem **ARP**.

c. Identificação da mensagem como um pedido ARP (*ARP Request*):

A análise da estrutura da mensagem ARP permite confirmar, de duas formas distintas, que se trata de um pedido ARP.

Para além de ao analisar a mensagem ARP dizer que se trata de um *ARP request* (*A*ddress *R*esolution *P*rotocol (request)), existem mais duas possíveis maneiras de identificar que se trata, efetivamente de um pedido ARP:

- **Campo “*Opcode*” da mensagem ARP:**
 - O campo *Opcode* tem o valor 1, que representa um *ARP Request*, ou seja, um pedido de resolução de endereço.
- **Endereço MAC de destino na trama *Ethernet*:**
 - O endereço MAC de destino da trama é ff:ff:ff:ff:ff:ff, ou seja, um *broadcast Ethernet*. Este tipo de endereço é característico dos pedidos ARP, pois o emissor ainda não conhece o MAC de destino e precisa que todos os dispositivos da rede analisem a mensagem.

Conclusão: A análise das tramas *Ethernet* capturada permite concluir, com base no endereço de destino, no campo *Type* e no conteúdo da mensagem ARP, que se trata efetivamente de um *ARP Request*. Este pedido foi emitido pelo *host* Aladdin com o objetivo de obter o endereço MAC correspondente ao IP 10.0.0.1.

2.3.3 Questão 3

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

- a. Qual o valor do campo ARP *opcode*? O que especifica?
- b. Em que campo da mensagem ARP está a resposta ao pedido ARP efetuado?
- 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).

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

Resposta:

Novamente, para contexto antes de responder às perguntas, está abaixo uma captura de ecrã de um ARP *reply* capturado através do *Wireshark* no *host* Aladdin.

```

▼ Frame 28: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface veth2.0.8a, id 0
  ▼ Interface id: 0 (veth2.0.8a)
    Interface name: veth2.0.8a
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 14, 2025 11:40:44.617656044 WEST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1744627244.617656044 seconds
    [Time delta from previous captured frame: 0.000046996 seconds]
    [Time delta from previous displayed frame: 0.000046996 seconds]
    [Time since reference or first frame: 29.897526059 seconds]
    Frame Number: 28
    Frame Length: 42 bytes (336 bits)
    Capture Length: 42 bytes (336 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
  ▼ Ethernet II, Src: 00:00:00_aa:00:02 (00:00:00:aa:00:02), Dst: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
    ▶ Destination: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
    ▶ Source: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
    Type: ARP (0x0806)
  ▼ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 00:00:00_aa:00:02 (00:00:00:aa:00:02)
    Sender IP address: 10.0.0.1
    Target MAC address: 00:00:00_aa:00:01 (00:00:00:aa:00:01)
    Target IP address: 10.0.0.21

```

Figura 8: ARP *reply* do Aladdin.

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

O valor do campo ARP *opcode* é 2, que indica que esta mensagem é uma ARP *reply* (resposta ARP).

- 1 - ARP *request* (pedido).
- 2 - ARP *reply* (resposta).

Portanto, o dispositivo que envia o ARP *reply* está a informar qual é o seu endereço MAC associado ao endereço IP solicitado no ARP *request*.

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

No contexto do ARP, a “resposta” diz respeito principalmente ao endereço MAC que corresponde ao IP consultado. Logo, a informação que efetivamente satisfaz o pedido (“Quem tem IP X?”) encontra-se no campo **Sender MAC address** (juntamente com o **Sender IP address**, que confirma o IP respondido).

Por outras palavras, ao recebermos um ARP *reply*, sabemos que o IP requisitado (no campo **Sender IP address**) está associado ao **Sender MAC address**. Esses campos são:

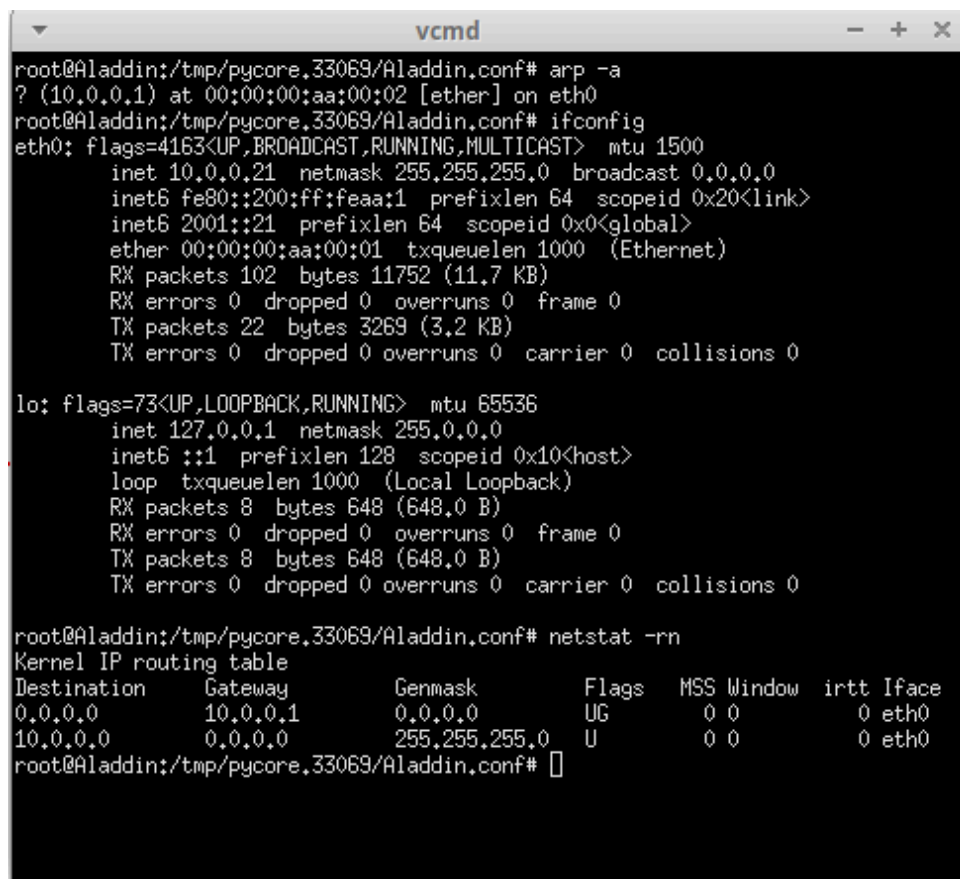
- **Sender MAC address:** 00:00:00:aa:00:02 (exemplo do valor visto na Figura 8).
- **Sender IP address:** 10.0.0.1 .

Estes dois campos em conjunto respondem de forma completa ao pedido ARP inicial, fornecendo o MAC vinculado ao IP consultado.

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).

Primeiramente, serão apresentados os resultados obtidos no terminal de Aladdin, sendo que, posteriormente, discutiremos as conclusões que foram retiradas das imagens.

Assim sendo, segue-se a seguinte imagem:



```
root@Aladdin:/tmp/pycore.33069/Aladdin.conf# arp -a
? (10.0.0.1) at 00:00:00:aa:00:02 [ether] on eth0
root@Aladdin:/tmp/pycore.33069/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 102 bytes 11752 (11.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22 bytes 3269 (3.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 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

root@Aladdin:/tmp/pycore.33069/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.33069/Aladdin.conf#
```

Figura 9: Comandos `ifconfig`, `netstat -rn` e `arp -a`.

Com base nos comandos executados no *host* Aladdin, identificamos que o seu endereço MAC é 00:00:00:aa:00:01, associado ao IP 10.0.0.21 na interface eth0. Este endereço MAC corresponde, portanto, ao próprio sistema Aladdin, sendo utilizado como origem ou destino em tramas locais.

Além disso, a tabela ARP (obtida com `arp -a`) mostra uma única entrada: o **Gateway** padrão da rede, com o endereço MAC 00:00:00:aa:00:02 e IP 10.0.0.1. Este *Gateway* é o *router* responsável por encaminhar o tráfego para redes externas, conforme confirmado pela tabela de rotas (`netstat -rn`), que indica que todo o tráfego não local (0.0.0.0) é direcionado para 10.0.0.1.

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

Quando um dispositivo na rede precisa descobrir o endereço MAC associado a um determinado endereço IP, ele envia uma solicitação ARP (ARP *Request*) em modo **broadcast**, com o endereço de

destino configurado como `ff:ff:ff:ff:ff:ff`. Este mecanismo garante que **todos os dispositivos na rede local recebam o pedido**. No entanto, quando o dispositivo que possui o IP solicitado responde, este fá-lo diretamente para o solicitante, utilizando o modo **unicast**. Isto significa que a resposta é enviada exclusivamente para o endereço MAC do dispositivo que originou a solicitação, evitando tráfego desnecessário na rede.

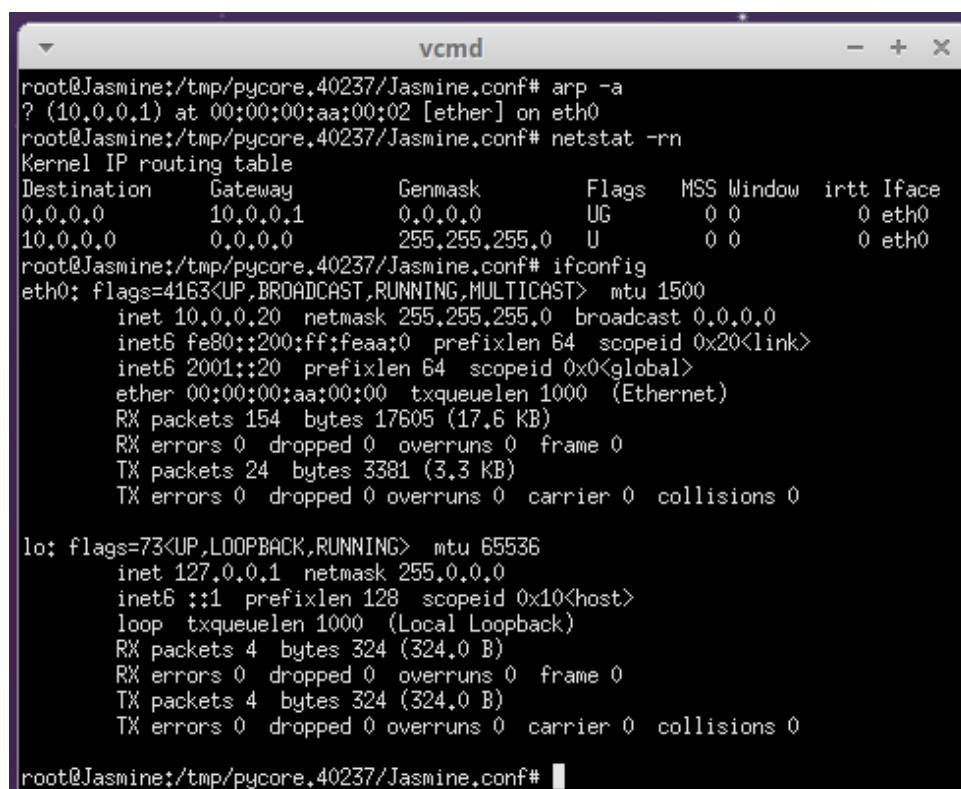
O uso do modo **unicast** para respostas ARP é uma escolha inteligente do protocolo ARP para otimizar o tráfego na rede. Se as respostas ARP fossem enviadas em modo **broadcast**, todos os dispositivos na rede precisariam de processar o *frame*, mesmo que não fossem os destinatários. Isto **consumiria largura de banda desnecessária** e aumentaria a carga de processamento em dispositivos que não têm interesse na comunicação. Ao direcionar a resposta apenas ao solicitante, o protocolo ARP minimiza o tráfego e melhora a eficiência geral da rede.

Assim sendo, tendo em conta o **ARP Reply** apresentado anteriormente, conclui-se que este exemplifica perfeitamente o funcionamento padrão do protocolo ARP, onde as respostas são enviadas em modo **unicast** para garantir eficiência e reduzir o tráfego desnecessário na rede. O endereço MAC de destino (`00:00:00:aa:00:01`) e os campos específicos da resposta ARP, como o *Opcode Reply* e os endereços de destino, confirmam que a comunicação é direcionada e exclusiva, caracterizando-a como **unicast**. Este comportamento é **essencial para manter a rede organizada e eficiente**, evitando sobrecarga e garantindo que as comunicações sejam rápidas e diretas.

2.3.4 Questão 4

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

Resposta:



```
root@Jasmine:/tmp/pycore.40237/Jasmine.conf# arp -a
? (10.0.0.1) at 00:00:00:aa:00:02 [ether] on eth0
root@Jasmine:/tmp/pycore.40237/Jasmine.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@Jasmine:/tmp/pycore.40237/Jasmine.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.20 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 fe80::200:ff:feaa:0 prefixlen 64 scopeid 0x20<link>
    inet6 2001::20 prefixlen 64 scopeid 0x0<global>
    ether 00:00:00:aa:00:00 txqueuelen 1000 (Ethernet)
    RX packets 154 bytes 17605 (17.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 3381 (3.3 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 4 bytes 324 (324.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 324 (324.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@Jasmine:/tmp/pycore.40237/Jasmine.conf#
```

Figura 10: Comandos em Jasmine.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|------------------------|-------------|----------|--------|--|
| 1 | 0.000000 | fe80::18da:40ff:fea... | ff02::1 | OSPF | 263 | Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR ... |
| 2 | 1.345590270 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 3 | 1.359995136 | fe80::200:ff:feaa:2 | ff02::5 | OSPF | 90 | Hello Packet |
| 4 | 1.759922918 | fe80::6054:37ff:fe4... | ff02::fb | MDNS | 263 | Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR ... |
| 5 | 3.345719843 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 6 | 3.448665933 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 7 | 7.349681939 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 8 | 9.349947103 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 9 | 11.312837569 | fe80::200:ff:feaa:2 | ff02::5 | OSPF | 90 | Hello Packet |
| 10 | 11.350003235 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 11 | 12.538989180 | fe80::200:ff:feaa:1 | ff02::2 | ICMPv6 | 70 | Router Solicitation from 00:00:00:aa:00:01 |
| 12 | 13.350169153 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 13 | 14.331103046 | fe80::200:ff:feaa:0 | ff02::2 | ICMPv6 | 70 | Router Solicitation from 00:00:00:aa:00:00 |
| 14 | 14.843118999 | fe80::18da:40ff:fea... | ff02::2 | ICMPv6 | 70 | Router Solicitation from 62:54:37:44:ed:8a |
| 15 | 15.350716765 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 16 | 16.007575314 | fe80::18da:40ff:fea... | ff02::fb | MDNS | 263 | Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR ... |
| 17 | 16.891293474 | fe80::6054:37ff:fe4... | ff02::2 | ICMPv6 | 70 | Router Solicitation from 62:54:37:44:ed:8a |
| 18 | 17.350861051 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 19 | 17.761968361 | fe80::6054:37ff:fe4... | ff02::fb | MDNS | 263 | Standard query 0x0000 PTR _nfs._tcp.local, "QM" question PTR ... |
| 20 | 19.350923603 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 21 | 21.289898342 | fe80::200:ff:feaa:2 | ff02::5 | OSPF | 90 | Hello Packet |
| 22 | 21.351114192 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 23 | 23.351359990 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 24 | 25.351523993 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 25 | 27.351766579 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 26 | 29.352202065 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 27 | 31.301694637 | fe80::200:ff:feaa:2 | ff02::5 | OSPF | 90 | Hello Packet |
| 28 | 31.352366638 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |
| 29 | 31.359075743 | fe80::200:ff:feaa:2 | ff02::16 | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 30 | 32.123389853 | fe80::200:ff:feaa:2 | ff02::16 | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 31 | 33.353141994 | 10.0.0.1 | 224.0.0.5 | OSPF | 78 | Hello Packet |

Figura 11: Wireshark em Jasmine.

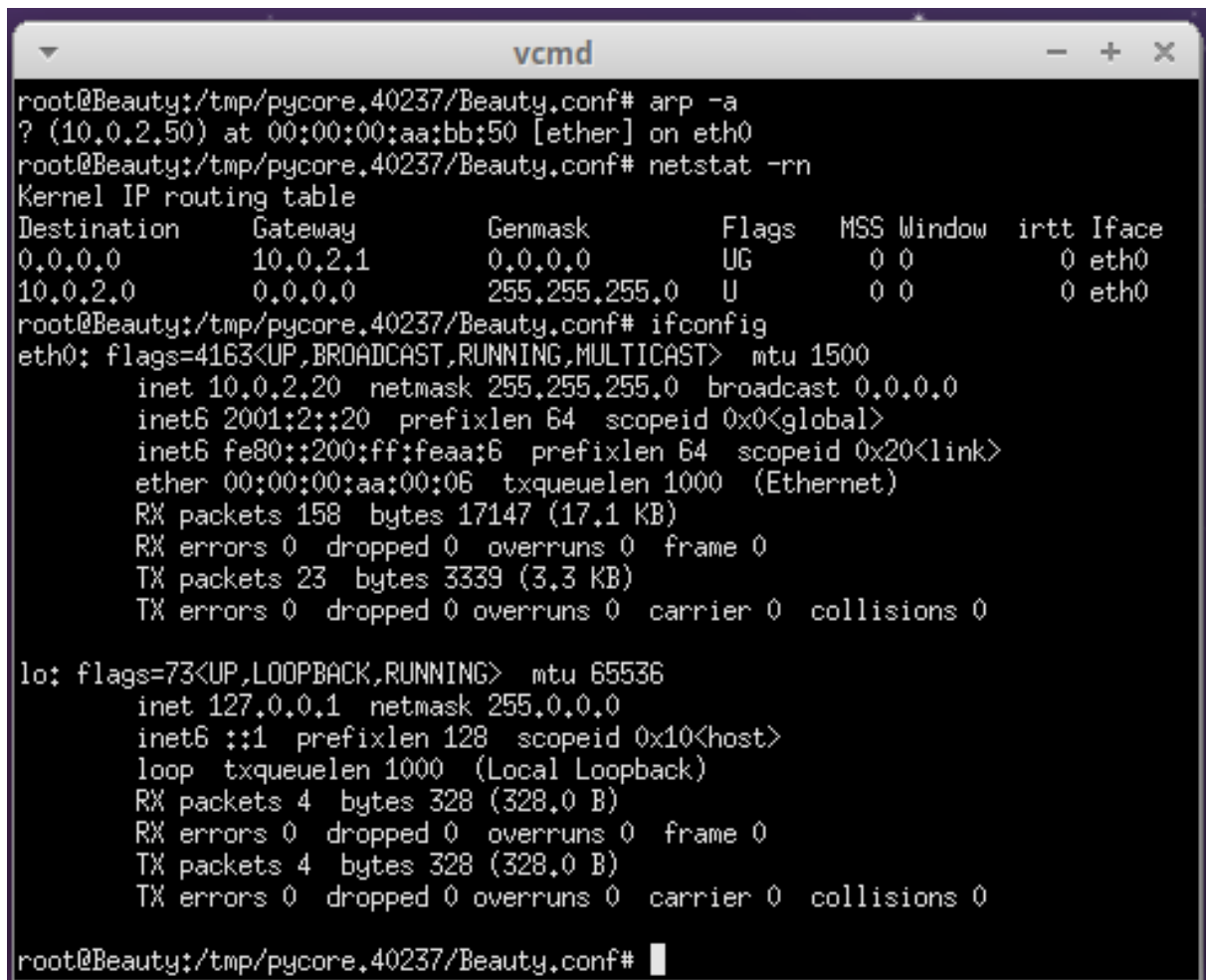
A partir da análise das capturas de rede e dos comandos executados em Jasmine, conclui-se que Jasmine não teve conhecimento do tráfego secreto gerado por Aladdin. A razão para tal deve-se ao facto de o tráfego em questão não ter sido dirigido para ela, nem ter sido transmitido em **broadcast** numa forma que a envolvesse.

A tabela ARP de Jasmine apenas mostra o *Gateway* (10.0.0.1), sem registos de outros dispositivos com os quais Aladdin possa ter comunicado de forma reservada. Além disso, as capturas de rede em Jasmine revelam apenas tráfego genérico, como pacotes OSPF (224.0.0.5) e mensagens ICMPv6, que são destinados a grupos ou a toda a rede, mas não incluem comunicações diretas entre Aladdin e outros *hosts*.

2.3.5 Questão 5

Problema: 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?

Resposta:



```
root@Beauty:/tmp/pycore.40237/Beauty.conf# arp -a
? (10.0.2.50) at 00:00:00:aa:bb:50 [ether] on eth0
root@Beauty:/tmp/pycore.40237/Beauty.conf# netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0          10.0.2.1        0.0.0.0         UG        0 0        0 eth0
10.0.2.0         0.0.0.0         255.255.255.0   U        0 0        0 eth0
root@Beauty:/tmp/pycore.40237/Beauty.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.20 netmask 255.255.255.0  broadcast 0.0.0.0
    inet6 2001:2::20 prefixlen 64  scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:6 prefixlen 64  scopeid 0x20<link>
    ether 00:00:00:aa:00:06  txqueuelen 1000  (Ethernet)
    RX packets 158  bytes 17147 (17.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 23  bytes 3339 (3.3 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 4  bytes 328 (328.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4  bytes 328 (328.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

root@Beauty:/tmp/pycore.40237/Beauty.conf#
```

Figura 12: Comandos em Beauty.

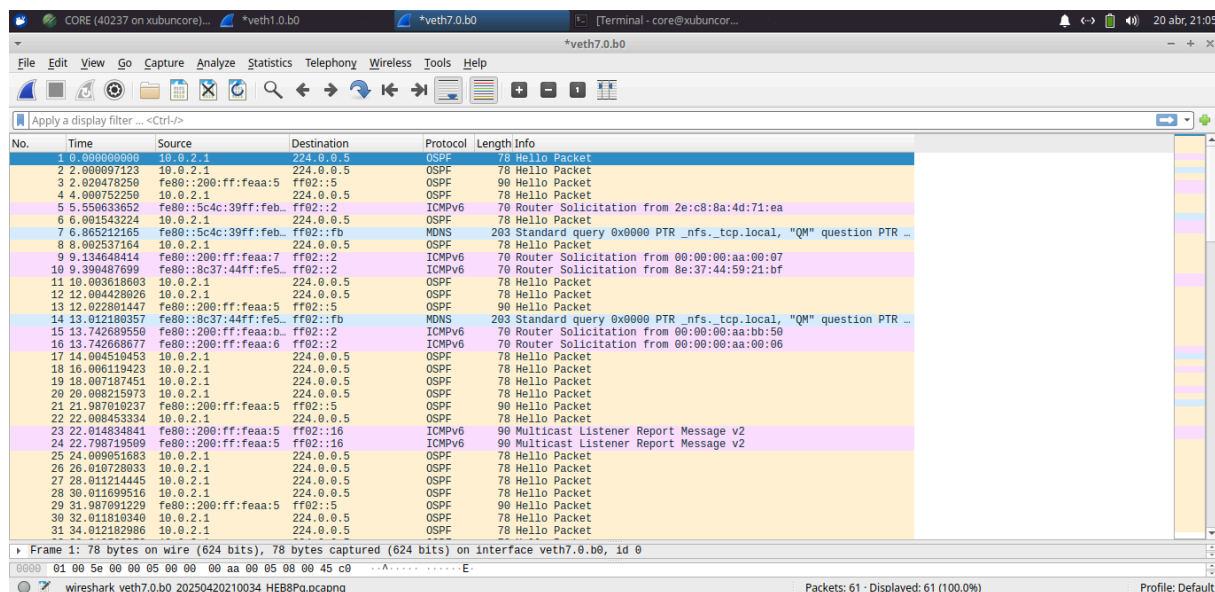


Figura 13: Wireshark em Beauty.

Da mesma forma, Beauty não teve conhecimento do tráfego secreto gerado por Beast. A análise da tabela ARP de Beauty mostra apenas uma entrada (10.0.2.50), sem evidências de comunicação direta com Beast (10.0.2.21).

As capturas de rede em Beauty exibem tráfego **broadcast**, como pacotes OSPF e mensagens ICMPv6, mas nenhuma comunicação **unicast** entre Beast e outros *hosts*. Além disso, o facto de Beauty e Beast estarem na mesma sub-rede (10.0.2.0/24) não significa que Beauty tenha acesso automático a todas as comunicações de Beast.

Portanto, tal como Jasmine, Beauty apenas detetou tráfego de controlo de rede (OSPF, ICMPv6), mas não as comunicações reservadas de Beast.

2.3.6 Questão 6

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

Resposta:

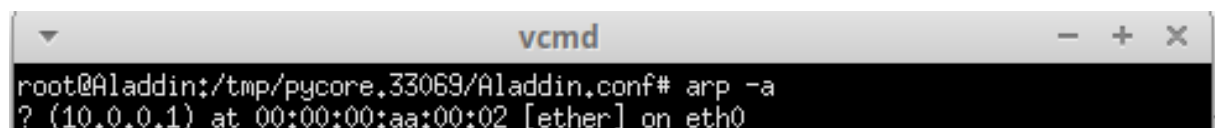


Figura 14: Tabela Arp de Aladdin.

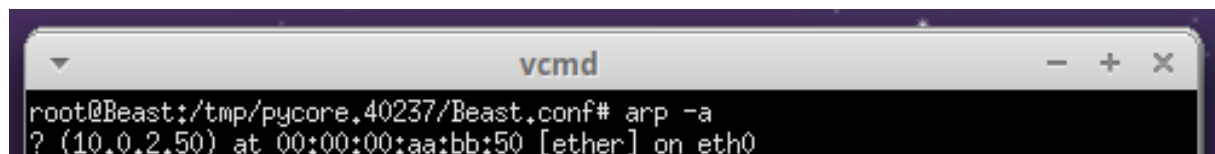


Figura 15: Tabela Arp de Beast.

Ao consultar a tabela ARP de Aladdin e Beast, é possível observar uma diferença fundamental entre as informações armazenadas em cada uma delas. Na máquina Aladdin, a tabela ARP contém a

entrada correspondente ao endereço IP 10.0.0.1 associado ao endereço MAC 00:00:00:aa:00:02. Já na máquina Beast, a tabela ARP apresenta uma entrada com o endereço IP 10.0.2.50 vinculado ao endereço MAC 00:00:00:aa:bb:50.

A principal diferença entre essas tabelas está nos endereços IP e MAC que cada máquina conhece. Isto é, as tabelas ARP possuem apenas a entrada relativa a outro dispositivo com o qual já estabeleceu comunicação direta, ou para o qual foi necessário resolver o endereço MAC correspondente ao IP. Isto reflete o funcionamento dinâmico do protocolo ARP, que apenas armazena as entradas que foram recentemente resolvidas por meio de requisições na rede local.

2.3.7 Questão 7

Problema: 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 recepção do primeiro pacote que contém dados do acesso remoto.

Resposta: A resposta a esta questão será dada por diagramas particionados, ordenados por ordem cronológica, devido à dimensão e complexidade da mesma.

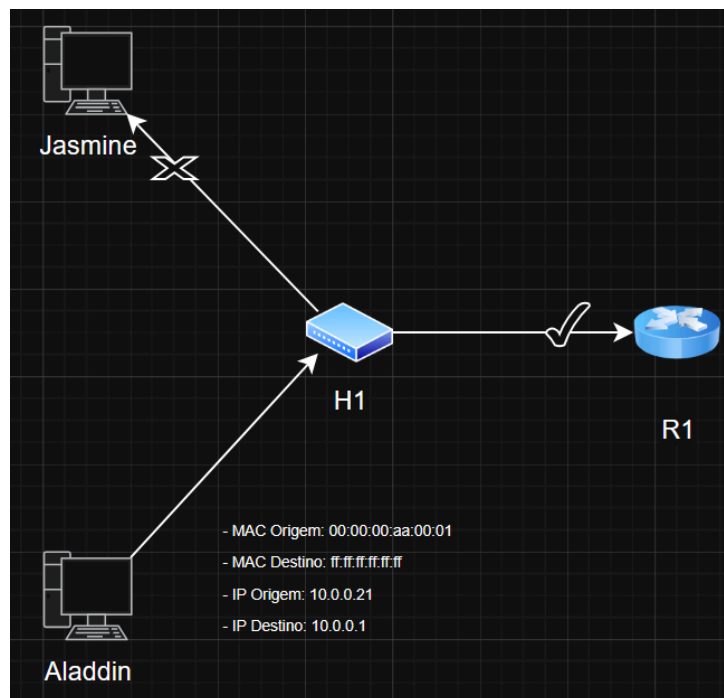


Figura 16: ARP Request de Aladdin.

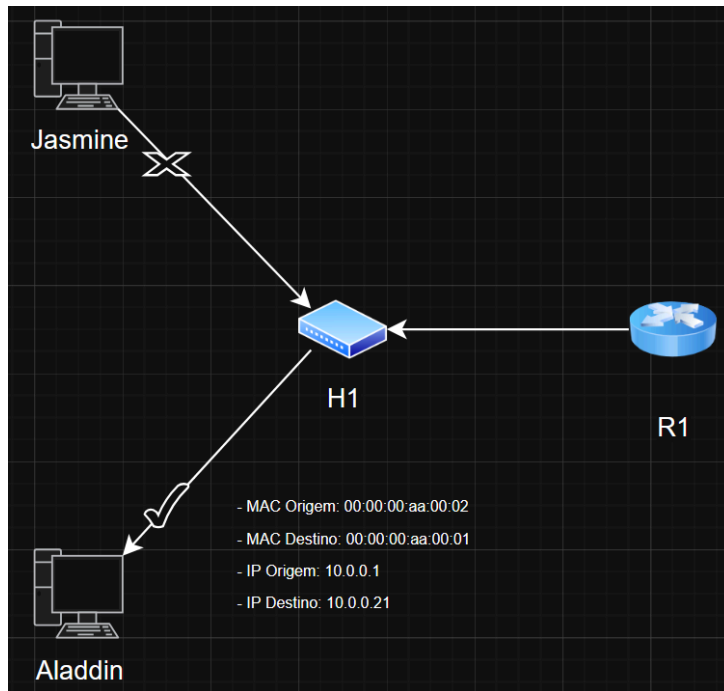


Figura 17: ARP Reply de R1.

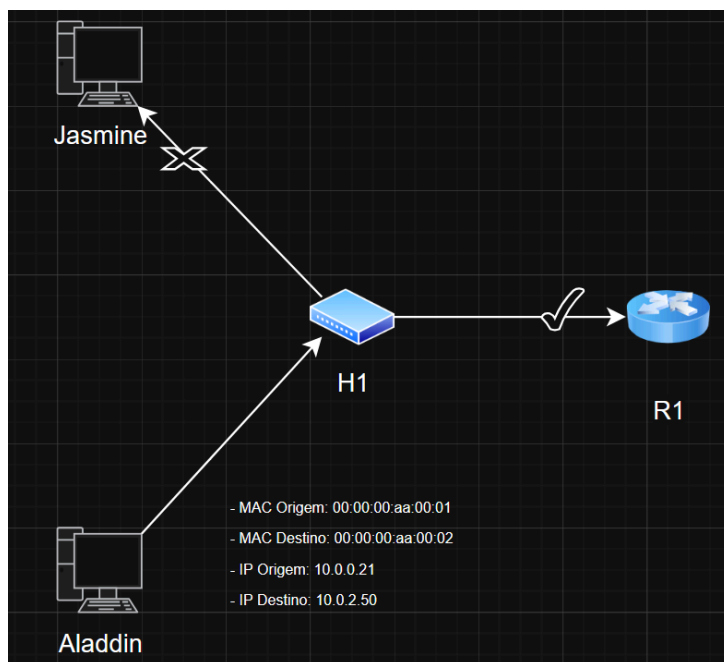


Figura 18: Tentativa de envio de frame para DServer 1.

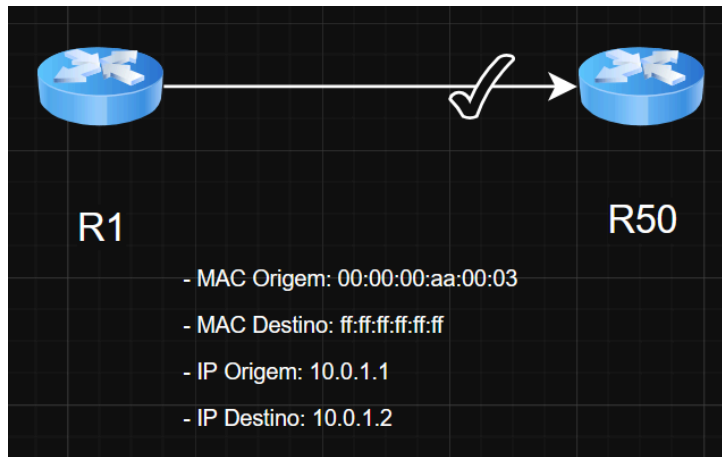


Figura 19: ARP *Request* entre R1 e R50.

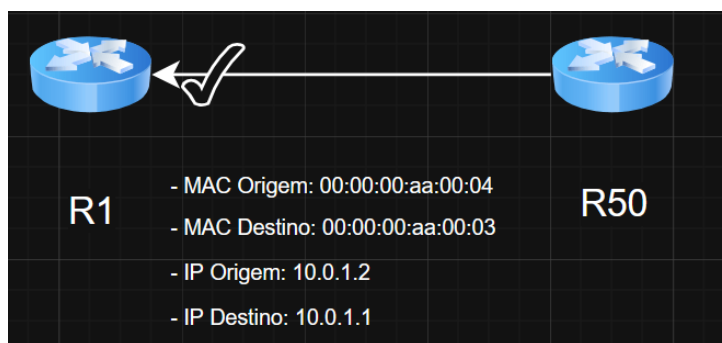


Figura 20: ARP *Reply* entre R1 e R50.

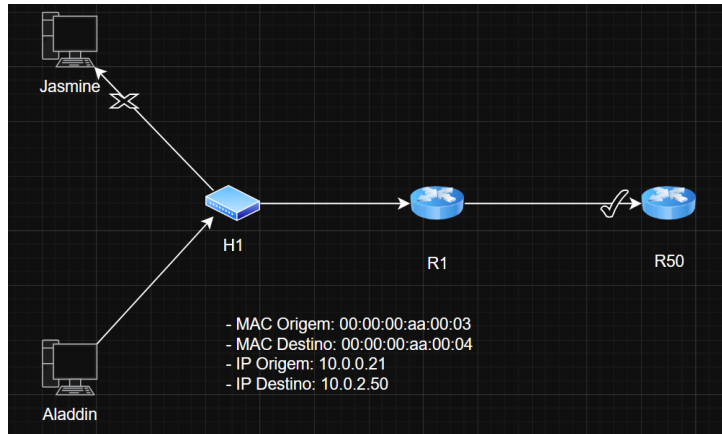


Figura 21: Tentativa de envio de frame para *DServer 2*.

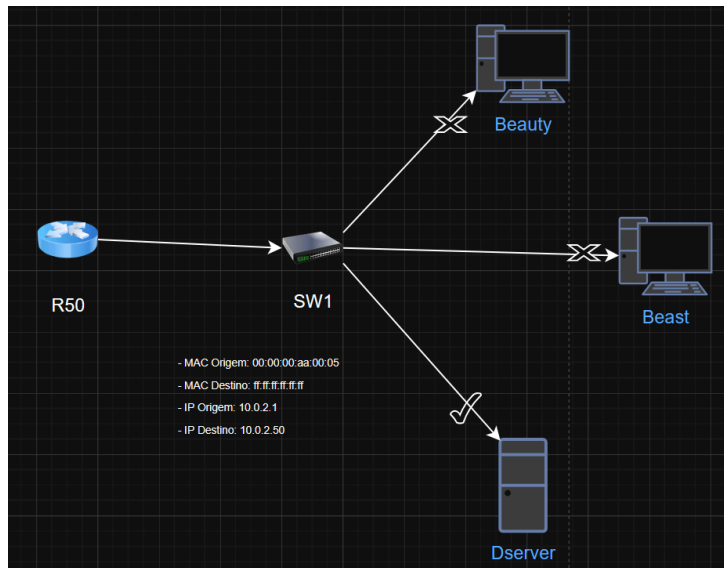


Figura 22: ARP Request entre R50 e DServer.

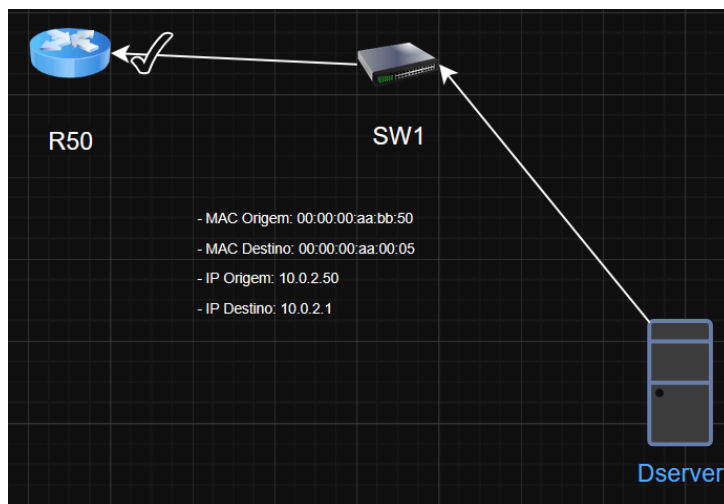


Figura 23: ARP Reply entre R50 e DServer.

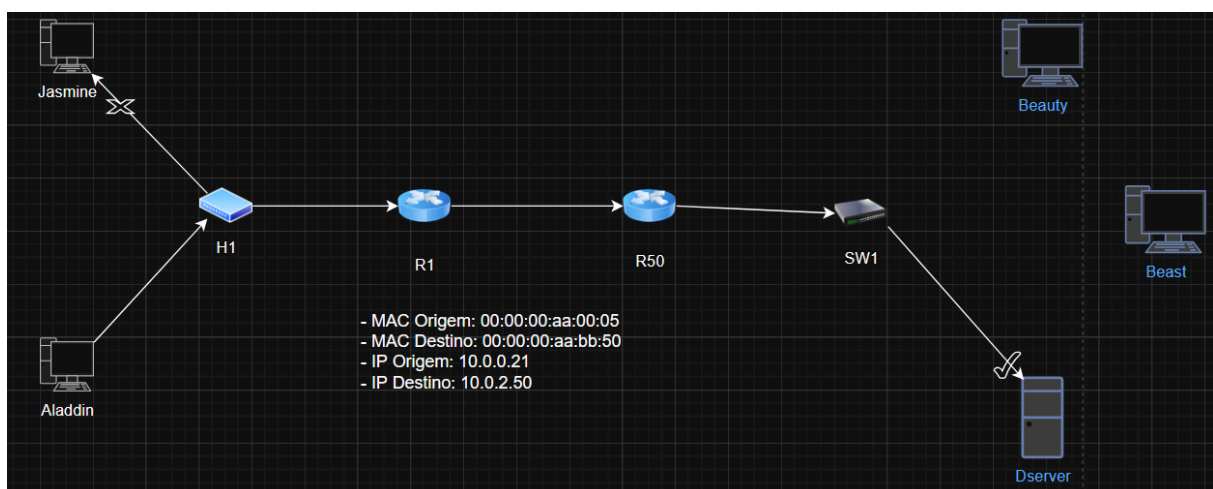


Figura 24: Envio de frame de Aladdin para DServer.

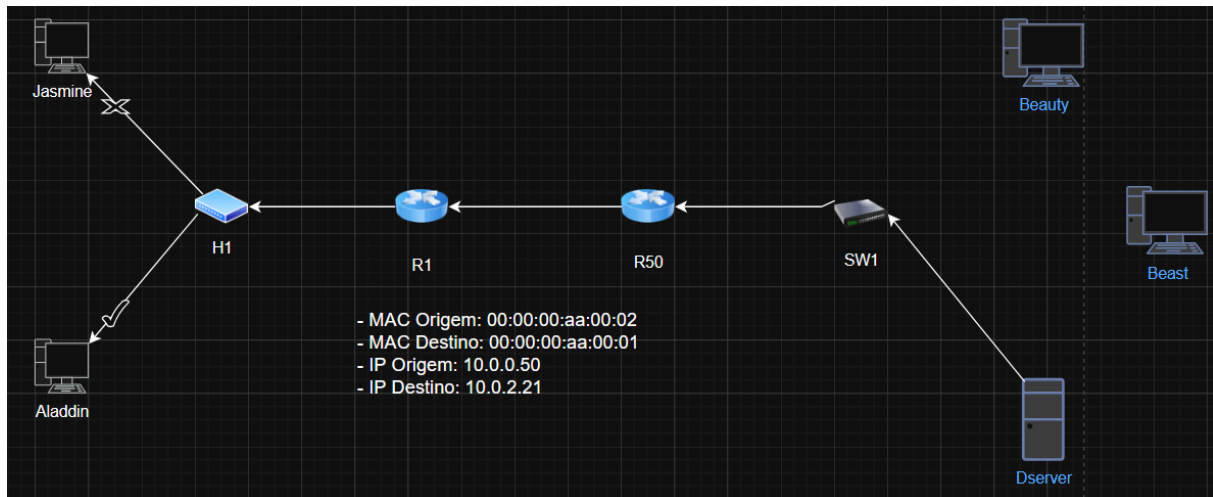


Figura 25: Resposta de DServer para Aladdin.

2.3.8 Questão 8

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

Resposta: A tabela de comutação (CAM *table*) de um *switch* associa endereços MAC de dispositivos às portas físicas do equipamento. Essa associação permite o encaminhamento eficiente de *frames* na rede, evitando *flooding* desnecessário.

Para construir manualmente a CAM *table* de SW1, foram considerados:

- Os **endereços MAC** de cada dispositivo conectado.
- A **atribuição de portas** físicas do *switch*.
- A **topologia lógica e endereços IP/sub-redes** para validação.
- **Saídas** dos Dispositivos.

A seguir, apresentam-se os dados relevantes que resultam do comando `ifconfig` -a em cada *host* conectado ao *switch* SW1.

Beauty:

- **Interface:** eth0.
- **IP:** 10.0.2.20/24.
- **MAC:** 00:00:00:aa:00:06.

Beast:

- **Interface:** eth0.
- **IP:** 10.0.2.21/24.
- **MAC:** 00:00:00:aa:00:07.

R50:

- **Interface:** eth1.
- **IP:** 10.0.2.1/24.
- **MAC:** 00:00:00:aa:00:05.

DServer:

- **Interface:** eth0.
- **IP:** 10.0.2.50/24.
- **MAC:** 00:00:00:aa:bb:50.

Tabela de Comutação do SW1

Abaixo está a CAM *table* manual de SW1, com portas Fa0/1 a Fa0/4 atribuídas conforme a topologia.

| VLAN | MAC Address | Porta SW1 | Descrição |
|------|-------------------|-----------|-------------------------------------|
| 1 | 00:00:00:aa:00:05 | Fa0/1 | Uplink para o router R50 (10.0.2.1) |
| 1 | 00:00:00:aa:00:06 | Fa0/2 | Host Beauty (10.0.2.20/24) |
| 1 | 00:00:00:aa:00:07 | Fa0/3 | Host Beast (10.0.2.21/24) |
| 1 | 00:00:00:aa:bb:50 | Fa0/4 | Servidor DServer (10.0.2.50/24) |

Tabela 2: Tabela de Comutação do Switch SW1.

Observações:

- **VLAN 1:** Utilizada como padrão, já que a rede não possui segmentação lógica.
- **Eficiência:** A tabela elimina a necessidade de *flooding*, pois todos os dispositivos estão mapeados.
- **Campos adicionais:** Poderíamos acrescentar uma coluna com “**Tipo**” (dinâmico/estático) ou com o “*Timestamp*”, mas o que é relevante neste exercício são as 4 colunas presentes na tabela acima.

Validação da Topologia

As figuras abaixo comprovam as configurações de rede dos dispositivos, incluindo endereços MAC e IPs:

```
root@Beauty:/tmp/pycore.35919/Beauty.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.20 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 2001:2::20 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:6 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:00:06 txqueuelen 1000 (Ethernet)
    RX packets 65 bytes 7560 (7.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 752 (752.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@Beauty:/tmp/pycore.35919/Beauty.conf#
```

Figura 26: Resultado do comando `ifconfig -a` no terminal da **Beauty**.


```

root@Beast:/tmp/pycore.35919/Beast.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.21 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 2001:2::21 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:7 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:00:07 txqueuelen 1000 (Ethernet)
    RX packets 121 bytes 11898 (11.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 962 (962.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@Beast:/tmp/pycore.35919/Beast.conf# █

```

Figura 27: Resultado do comando `ifconfig -a` no terminal do **Beast**.

```

root@R50:/tmp/pycore.33149/R50.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.1.2 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 2001:1::2 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:4 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:00:04 txqueuelen 1000 (Ethernet)
    RX packets 63 bytes 7346 (7.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 25 bytes 2458 (2.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.1 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 2001:2::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:5 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:00:05 txqueuelen 1000 (Ethernet)
    RX packets 56 bytes 6679 (6.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 1888 (1.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Figura 28: Resultado do comando `ifconfig -a` no terminal do **router R50**.

```

root@DServer:/tmp/pycore.35919/DServer.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.50 netmask 255.255.255.0 broadcast 0.0.0.0
    inet6 2001:2::10 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::200:ff:feaa:bb50 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:aa:bb:50 txqueuelen 1000 (Ethernet)
    RX packets 145 bytes 13770 (13.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 962 (962.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@DServer:/tmp/pycore.35919/DServer.conf#

```

Figura 29: Resultado do comando `ifconfig -a` no terminal do **DServer**.

Conclusão

A CAM *table* acima garante que o **switch SW1** encaminha cada *frame* diretamente para a porta correta, melhorando a performance e evitando *broadcast* desnecessário na rede.

2.4 Serviço de NAT/PAT

2.4.1 Questão 1

Problema: 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 tráfego 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.).

Resposta: Tendo em conta que o nosso grupo de trabalho é o Grupo 50, Rxy será R50 a partir de agora.

Para resolver o problema de conectividade entre as redes locais (LANs) de Jasmine e Aladdin, bem como de Beauty e Beast, mantendo o acesso a serviços como SSH e outros através da *Internet*, é necessário implementar uma **solução que permita a tradução de endereços privados para públicos** de forma dinâmica. Como o ISP não encaminha tráfego para redes privadas remotas, os *routers* R1 e R50 não conseguem comunicar diretamente com esses endereços. A solução passa por utilizar o **PAT dinâmico** (*Port Address Translation*), que é uma variante do NAT (*Network Address Translation*), combinado com PAT estático no ISP para garantir acessibilidade a serviços específicos.

O PAT dinâmico permite que múltiplos dispositivos numa LAN partilhem um único endereço IP público para aceder à *Internet*. Quando um dispositivo envia um pacote, o *router* (R1 ou R50) **substitui o endereço IP privado de origem pelo endereço IP público** do *router*, atribuindo também uma porta única para distinguir as diferentes sessões. Isto resolve o problema de conectividade geral, pois

os **pacotes enviados para a *Internet* aparecem como provenientes de um endereço público**, que o ISP consegue encaminhar.

No entanto, para garantir que serviços como o SSH permaneçam acessíveis a partir da *Internet*, é necessário configurar PAT estático (também conhecido como *port forwarding*) no ISP. O **PAT estático mapeia uma porta específica do endereço IP público do router para um endereço IP privado** e porta correspondente num servidor interno.

Desta forma, o PAT dinâmico assegura a conectividade geral para a *Internet*, enquanto o PAT estático no ISP permite o acesso controlado a serviços internos. Esta abordagem mantém todas as funcionalidades anteriores **sem exigir alterações no endereçamento privado das LANs**, resolvendo o problema de encaminhamento imposto pelo ISP.

3. Parte II

3.1 Problema Geral

A princesa Jasmine, incomodada com a presença de cabos *Ethernet* visíveis no palácio, propôs a substituição da infraestrutura de rede com fios por uma solução sem fios. Aladdin, atendendo à sugestão, optou por adquirir equipamento compatível com a tecnologia *Wi-Fi*. Com o intuito de compreender melhor o funcionamento da nova rede sem fios instalada, procedeu-se à realização de uma captura de tráfego, permitindo a análise do comportamento e das características do protocolo IEEE 802.11 em ambiente real.

Para confirmar a conectividade e obter dados representativos do funcionamento da rede, foi realizada uma longa captura do tráfego *Wi-Fi* com recurso à ferramenta *Wireshark*. As próximas questões deste relatório referem-se à análise dessa captura.

3.2 Acesso Rádio

Como pode ser observado de seguida, a sequência de *bytes* capturada inclui metainformações do nível físico, nomeadamente o *radiotap header* e outras informações de rádio — fornecidas pelo *firmware* da interface *Wi-Fi*. Estas informações adicionais acompanham os *bytes* que compõem as tramas definidas pelo padrão IEEE 802.11.

No âmbito desta análise, deve ser selecionada a trama de ordem 50, correspondente ao identificador de grupo atribuído para este estudo.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|---------------------|-------------------|----------|--------|--|
| 50 | 0.971665 | AlticeLabs_fc:f0:a0 | 52:90:27:97:1c:c3 | 802.11 | 380 | Probe Response, SN=1447, FN=0, Flags=...R...C, BI=100, SSID="ME0-FCF0A0" |

Figura 30: Trama de ordem 50 capturada.

3.2.1 Questão 1

Problema: Identifique em que frequência do espectro está a operar a rede sem fios e o canal que corresponde a essa frequência.

Resposta:

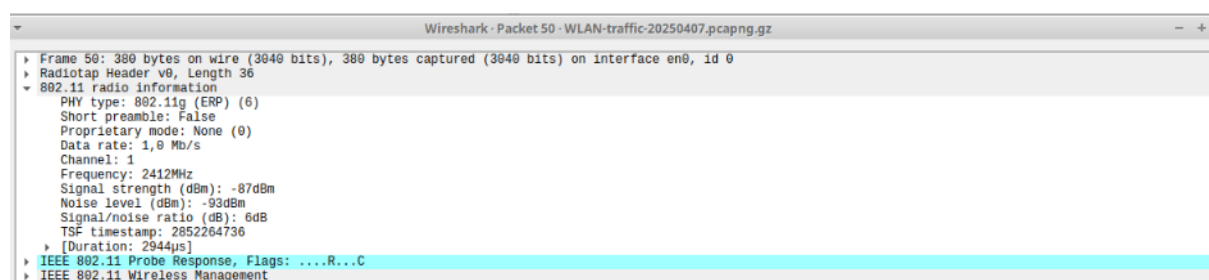


Figura 31: Trama de ordem 50 capturada, no campo 802.11 *radio information*.

Como é possível concluir pela imagem acima, a **Frequência** do espectro é de 2412MHz e o **Canal** que corresponde a essa mesma frequência é o *Channel 1*.

3.2.2 Questão 2

Problema: Identifique a versão da norma IEEE 802.11 que está a ser usada.

Resposta:

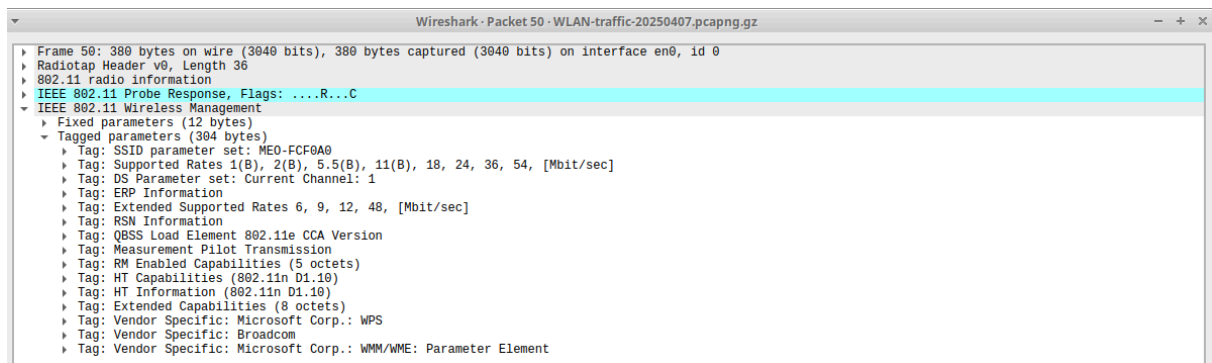


Figura 32: Trama de ordem 50 capturada, no campo 802.11 *Wireless Management*.

Ao analisar a imagem acima, deduz-se que a **versão da norma** IEEE 802.11 é a norma 802.11n, presente no campo *HT Capabilities* e *HT Information*.

3.2.3 Questão 3

Problema: 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.

Resposta:

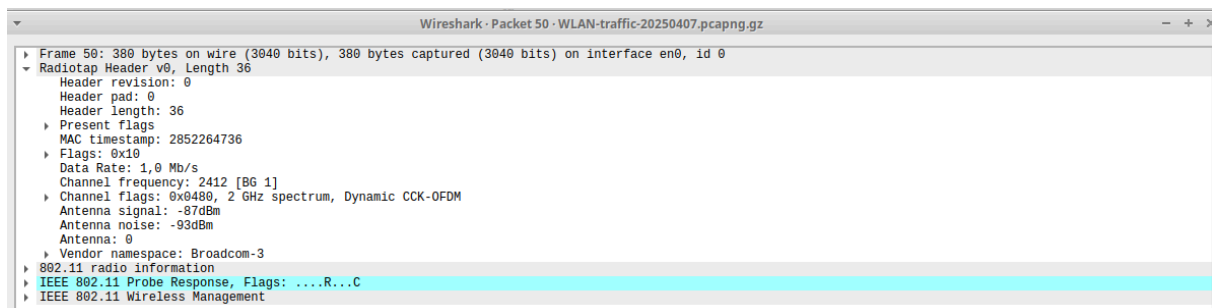


Figura 33: Trama de ordem 50 capturada, no campo *Radiotap*.

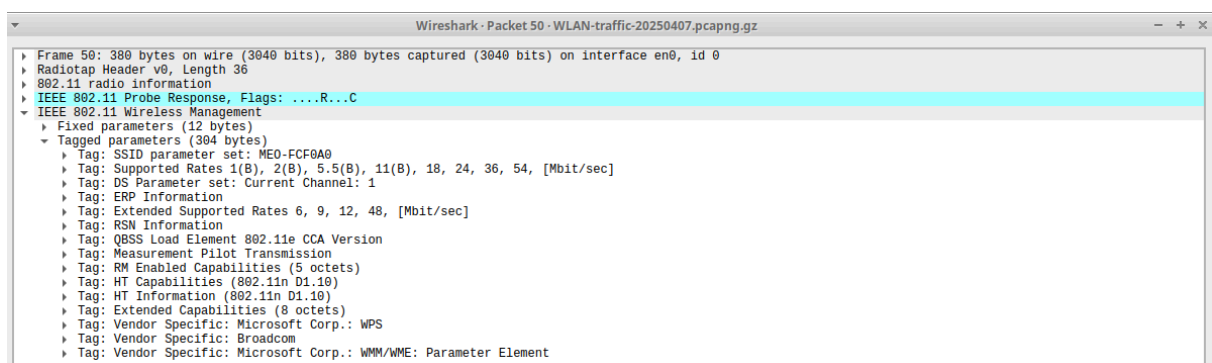


Figura 34: Trama de ordem 50 capturada, no campo 802.11 *Wireless Management*.

Analisando a Figura 33, observa-se que a **taxa de transmissão** é de 1.0 Mb/s, presente no campo *Data rate*, que corresponde à velocidade utilizada especificamente para enviar aquele quadro *Probe Response*, mas não representa a capacidade máxima da interface *Wi-Fi*. Este valor baixo é intencional e justifica-se pela **necessidade de garantir compatibilidade com dispositivos mais antigos** e pela natureza dos quadros de gestão na rede sem fios.

Os quadros de gestão, como os *Probe Response* e *Beacon frames*, são normalmente transmitidos nas **taxas mais básicas** suportadas pela rede - neste caso, 1.0 Mb/s, que é uma taxa característica do antigo padrão 802.11b. Esta abordagem **assegura que todos os dispositivos**, incluindo os que apenas suportam tecnologias *legacy* possam receber e interpretar corretamente estes quadros essenciais para o funcionamento da rede.

No entanto, quando analisamos os campos *Tagged Parameters* na Figura 34, verificamos que a rede **suporta taxas significativamente mais elevadas**. A lista de *Supported Rates* inclui valores desde 1 Mbps até 54 Mbps, abrangendo assim padrões desde o 802.11b até ao 802.11g. Mais importante ainda, a presença do campo *HT Capabilities* indica claramente que **a rede suporta o padrão 802.11n**, o que significa que a velocidade máxima teórica suportada é muito superior aos 54 Mbps do 802.11g.

Na realidade, o padrão 802.11n pode atingir velocidades teóricas de até 600 Mbps, dependendo da configuração específica (número de fluxos espaciais, largura do canal utilizado, etc.). Na prática, é comum obter **velocidades entre 150 a 300 Mbps** em condições normais de funcionamento. A razão pela qual não vemos estas velocidades mais elevadas listadas nos *Supported Rates* deve-se ao facto de estas serem negociadas dinamicamente através do MCS (*Modulation and Coding Scheme*) durante a comunicação de dados propriamente dita, e não estarem incluídas na lista estática de taxas suportadas.

Portanto, embora o quadro em análise tenha sido transmitido a apenas 1.0 Mbps, esta velocidade **não reflete de forma alguma a capacidade máxima da rede**. A presença do *HT Capabilities* confirma que a interface *Wi-Fi* está preparada para operar a velocidades muito superiores, típicas do padrão 802.11n.

3.3 Scanning Passivo e Scanning Ativo

Como referido anteriormente, as tramas *beacon* permitem realizar *scanning* passivo em redes IEEE 802.11 (*Wi-Fi*), possibilitando a deteção de redes sem a necessidade de envio ativo de pacotes por parte do cliente. Estas tramas são transmitidas periodicamente pelos pontos de acesso e contêm informações essenciais sobre a rede.

Com base na captura de tráfego disponibilizada e considerando 50 como o número do nosso Turno-Grupo (PL50), apresentam-se de seguida um conjunto de questões que visam explorar e analisar detalhadamente os dados extraídos das tramas capturadas, com especial foco nas tramas *beacon* e no seu respetivo conteúdo. Para a análise, selecionamos uma trama *beacon*, cuja ordem (ou terminação) corresponde ao ID de grupo mencionado e a mesma pode ser vista abaixo.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|--------------|--------------------|----------|--------|---|
| 89 | 1.461213 | HitronTechno | f3:9a:2: Broadcast | 802.11 | 362 | Beacon frame, SN=2542, FN=0, Flags=.....C, BI=100, SSID="FlyingNet" |

Figura 35: Trama *beacon* selecionada.

3.3.1 Questão 1

Problema: 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 1)?

Resposta: Ao aplicar o seguinte filtro no *Wireshark*:

```
1 wlan.fc.type_subtype == 0x08
```

Wireshark

Identificamos a **trama 89** como a quinquagésima (50.^o) trama *beacon* na captura de tráfego.

De seguida, para responder à segunda parte da questão colocada, foi necessário analisar a seguinte imagem:

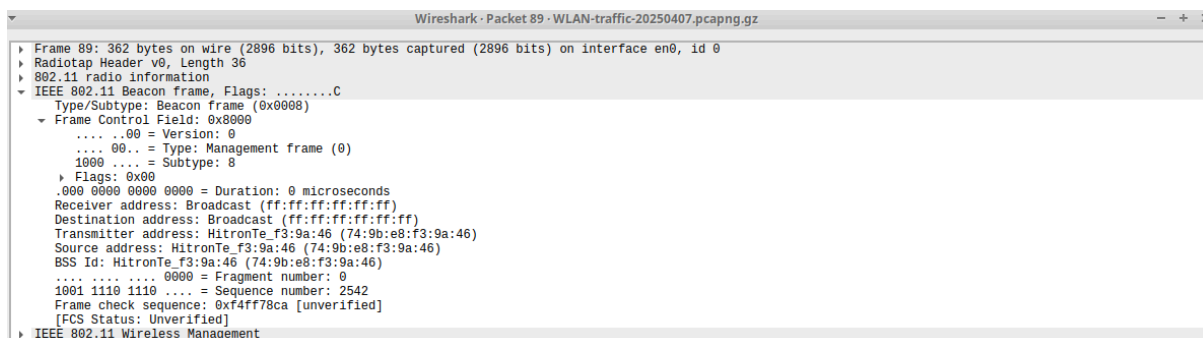


Figura 36: Trama *beacon* 89.

Como é possível constatar analisando a figura acima, vemos que o tipo desta trama é do tipo **Management frame**, com o valor de 0, presente no campo **Frame Control Field**. Nesse mesmo campo também vemos que o seu subtipo é 8.

Ao recorrer ao Anexo 2 fornecido, presente no final deste relatório, vemos que o tipo e subtipo da trama ocupam 2 *bits* e 4 *bits*, respetivamente. Isto pode ser verificado no campo **Frame Control Field**, que, na parte binária, podemos ver que o campo *Type* tem 00 e o campo *Subtype* tem 1000. Traduzindo estes valores obtemos os valores acima indicados: 0 para o tipo e 8 para o subtipo.

3.3.2 Questão 2

Problema: Verifique se está a ser usado o método de deteção de erros (CRC). Justifique.

Resposta: Antes de responder à questão, tivemos que ativar a verificação no *Wireshark*, indo a *Edit -> Preferences -> Protocols -> IPv4 -> "Validate Checksum if Possible"*.

Como é possível deduzir observando a Figura 36, presente na questão acima, o campo *Frame check sequence* tem o estado como *[unverified]*, indicando que o *Wireshark* **não verificou** se o CRC está válido. Esta conclusão ainda é reforçada pelo campo *FCS Status* que mostra o mesmo resultado. **Apesar de não ter havido verificação da validade do CRC, este foi usado.**

3.3.3 Questão 3

Problema: Justifique o porquê de ser necessário usar deteção de erros em redes sem fios.

Resposta: A deteção de erros, como o **CRC (Cyclic Redundancy Check)**, é essencial em redes sem fios devido às características intrínsecas do meio de transmissão. Num ambiente *wireless*, os dados são transmitidos através de ondas eletromagnéticas, que **estão sujeitas a uma série de interferências** e fenómenos físicos que podem corromper os pacotes durante a comunicação.

Em primeiro lugar, o meio sem fios é **altamente suscetível a interferências externas**, como sinais de outros dispositivos eletrónicos (micro-ondas, *Bluetooth*, redes *Wi-Fi* vizinhas), obstáculos físicos (paredes, móveis) ou até condições atmosféricas (em redes de longa distância). **Estas interferências**

podem causar alterações nos *bits* transmitidos, introduzindo erros nos dados recebidos. Sem um mecanismo de detecção de erros, o destinatário não teria forma de saber se a informação chegou corrompida, comprometendo a integridade da comunicação.

Além disso, as redes *Wi-Fi* **operam num meio partilhado e não confiável**, onde múltiplos dispositivos competem pelo mesmo espectro. Colisões e sobreposições de sinais são comuns, especialmente em ambientes congestionados. O CRC permite que o recetor **verifique se os dados foram recebidos exatamente como foram enviados**. Se um erro for detetado (devido a um CRC inválido), o pacote é descartado e o protocolo 802.11 pode solicitar uma retransmissão, garantindo que apenas informações válidas sejam processadas.

Outro fator crítico é a **natureza *half-duplex* das comunicações sem fios**, onde dispositivos não podem transmitir e receber simultaneamente no mesmo canal. Se um pacote corrompido fosse processado sem verificação, poderia levar a comportamentos inesperados nos protocolos de rede, como falhas no estabelecimento de ligações (ex: autenticação) ou corrupção de dados em aplicações sensíveis (ex: transferência de ficheiros, *streaming*).

Por fim, a detecção de erros é fundamental para a **eficiência da rede**. Se os erros não fossem detetados, os pacotes corrompidos seriam processados pelas camadas superiores (ex: TCP/IP), **levando a retransmissões desnecessárias a nível de protocolo**, aumentando a latência e reduzindo a largura de banda útil. O CRC, ao descartar tramas inválidas logo na camada física, otimiza o desempenho global da rede.

Em resumo, o uso de detecção de erros como o CRC em redes sem fios é indispensável para garantir **integridade dos dados, confiabilidade da comunicação e eficiência espectral**, mitigando os efeitos de um meio inerentemente ruidoso e imprevisível. Sem ele, as redes *Wi-Fi* seriam significativamente menos robustas e estariam sujeitas a falhas constantes.

3.3.4 Questão 4

Problema: 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.

Resposta: A periodicidade indica o intervalo de tempo entre o envio de tramas *beacon* pelo ponto de acesso (AP), permitindo a sincronização dos dispositivos na rede.

As taxas de transmissão suportadas mostram as velocidades com que o AP pode transmitir dados:

- **Supported Rates:** taxas básicas que todos os dispositivos devem suportar.
- **Extended Supported Rates:** taxas adicionais para melhorar o desempenho, mas não obrigatórias para todos os dispositivos.


```
IEEE 802.11 Wireless Management
  Fixed parameters (12 bytes)
    Timestamp: 56021709187
    Beacon Interval: 0,102400 [Seconds]
    Capabilities Information: 0x0431
  Tagged parameters (286 bytes)
    Tag: SSID parameter set: "FlyingNet"
    Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6(B), 9, 12(B), 18, [Mbit/sec]
      Tag Number: Supported Rates (1)
      Tag length: 8
      Supported Rates: 1(B) (0x82)
      Supported Rates: 2(B) (0x84)
      Supported Rates: 5.5(B) (0x8b)
      Supported Rates: 11(B) (0x96)
      Supported Rates: 6(B) (0x8c)
      Supported Rates: 9 (0x12)
      Supported Rates: 12(B) (0x98)
      Supported Rates: 18 (0x24)
    Tag: DS Parameter set: Current Channel: 1
    Tag: Traffic Indication Map (TIM): DTIM 0 of 1 bitmap
    Tag: Country Information: Country Code PT, Environment All
    Tag: ERP Information
    Tag: Extended Supported Rates 24(B), 36, 48, 54, [Mbit/sec]
      Tag Number: Extended Supported Rates (50)
      Tag length: 4
      Extended Supported Rates: 24(B) (0xb0)
      Extended Supported Rates: 36 (0x48)
      Extended Supported Rates: 48 (0x60)
      Extended Supported Rates: 54 (0x6c)
```

Figura 37: Periodicidade e taxas de transmissão da trama *beacon* selecionada.

Na trama *beacon* anterior:

- **Periodicidade (*Beacon Interval*):** 0.102400 segundos \approx 102.4 ms.
- **Taxas básicas (*Supported Rates*):** 1, 2, 5.5, 11, 6, 9, 12, 18 Mbit/s.
- **Taxas adicionais (*Extended Supported Rates*):** 24, 36, 48, 54 Mbit/s.

3.3.5 Questão 5

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

Resposta: Para identificar os SSIDs (identificadores de rede) dos pontos de acesso (APs) que estão a operar na vizinhança de uma estação (STA), seguimos os seguintes passos durante a análise da captura de tráfego no *Wireshark*:

1. **Aplicação do Filtro:** As tramas *beacon* são enviadas periodicamente pelos pontos de acesso (APs) para anunciar a sua presença e a disponibilidade da rede. Para isolar estas tramas, usámos o seguinte filtro abaixo no *Wireshark*. Este seleciona exclusivamente as tramas *beacon*, que contêm informações sobre os SSIDs e as capacidades do AP.

```
1 wlan.fc.type_subtype == 0x08
```

Wireshark

2. **Exportação para .txt:** A captura filtrada foi exportada para um arquivo .txt nomeado *wireshark_export.txt* usando a opção *File > Export Packet Dissections > As txt*. Este arquivo contém as informações dos SSIDs capturados.

3. **Processamento:** O arquivo de texto foi processado com recurso a um *script* Python para agrupar os SSIDs e contar as ocorrências:

```
1 def find_ssids(txt_file):  
2     ssids_count = {}  
3  
4     with open(txt_file, 'r') as file:  
5         for line in file:  
6             if 'SSID="' in line:  
7                 start = line.find('SSID="') + len('SSID="')  
8                 end = line.find('"', start)  
9                 if start != -1 and end != -1:  
10                     ssid = line[start:end]  
11                     ssids_count[ssid] = ssids_count.get(ssid, 0) + 1  
12  
13     return ssids_count  
14  
15 txt_file = 'wireshark_export.txt'  
16 found_ssids = find_ssids(txt_file)  
17  
18 for ssid, count in found_ssids.items():  
19     print(f'{ssid}: {count} occurrence(s)')
```

Este processo resultou na lista dos SSIDs únicos e sua respectiva contagem, permitindo identificar os APs ativos na vizinhança da STA. Sendo o *output*:

```
1 MEO-FCF0A0: 1934 occurrence(s)  
2 MEO-WiFi: 6637 occurrence(s)  
3 FlyingNet: 2930 occurrence(s) (Pode-se ignorar é o AP da rede local)  
4 MEO-9BF2A0: 1698 occurrence(s)  
5 phi_F41927C3C600: 232 occurrence(s)  
6 Masmorra do Sexo: 1483 occurrence(s)  
7 GVBRAGA_EXT: 748 occurrence(s)  
8 MEO-66DB70: 148 occurrence(s)  
9 NOS-26F6: 1155 occurrence(s)  
10 NOS-9946_EXT: 670 occurrence(s)  
11 MEO-828830: 1239 occurrence(s)  
12 MEO-854C80: 56 occurrence(s)  
13 NOS-52C6: 45 occurrence(s)  
14 NOS-C8B6: 73 occurrence(s)  
15 GVBRAGA_quarto: 293 occurrence(s)  
16 Vodafone-D0ED8A: 248 occurrence(s)  
17 GVBRAGA: 43 occurrence(s)  
18 MEO-F17570: 6 occurrence(s)  
19 NOS-FD24: 2 occurrence(s)
```

Todos estes valores são os SSIDs dos APs que operam na vizinhança da STA de captura. Dispensa-se mencionar que alguns tenham nomes minimamente questionáveis.

3.3.6 Questão 6

Problema: Estabeleça um filtro *Wireshark* apropriado que lhe permita visualizar todas as tramas *probing request* e *probing response*, simultaneamente.

Resposta: Para visualizar todas as tramas *Probe Request* e *Probe Response* simultaneamente no *Wireshark*, pode utilizar o seguinte filtro de exibição:

```
1 wlan.fc.type_subtype == 0x04 || wlan.fc.type_subtype == 0x05
```

Wireshark

- `wlan.fc.type_subtype == 0x04`: Este filtro exibe as tramas *Probe Request*, que são enviadas por estações (STAs) para descobrir redes disponíveis.
- `wlan.fc.type_subtype == 0x05`: Este filtro exibe as tramas *Probe Response*, que são enviadas pelos pontos de acesso (APs) em resposta a uma *Probe Request*.
- O operador `||` é utilizado para combinar ambos os filtros, permitindo que as tramas de *Probe Request* e *Probe Response* sejam exibidas simultaneamente.

O filtro composto pode ser aplicado na barra de filtros do *Wireshark* para capturar e visualizar as tramas de sondagem no processo de descoberta de redes sem fio.

3.3.7 Questão 7

Problema: 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.

Resposta: Para determinar a melhor qualidade de ligação possível para a STA, procedeu-se à análise das tramas do tipo *Beacon* (provenientes de *scanning* passivo) e *Probe Response* (resultantes de *scanning* ativo), capturadas com o seguinte filtro aplicado no *Wireshark*:

```
1 (wlan.fc.type_subtype == 0x05 || wlan.fc.type_subtype == 0x08) &&
   wlan.ssid == "ssid"
```

Wireshark

Este filtro permitiu isolar as tramas mais relevantes para o processo de descoberta de pontos de acesso (APs), mais especificamente, *Beacon Frames* (subtipo 0x08) e *Probe Responses* (subtipo 0x05), **permitindo a análise de cada AP**. O `wlan.ssid` restringe as tramas para aquelas que contêm o SSID especificado, garantindo que apenas as tramas correspondentes ao SSID desejado sejam exibidas.

A força do sinal foi determinada a partir do campo *Signal Strength*, incluído na camada 802.11 *radio information* das tramas capturadas. Estes valores são apresentados em dBm, sendo que **valores menos negativos correspondem a sinais mais fortes** e, portanto, melhor qualidade de ligação.

Abaixo segue uma imagem de exemplo do *Radiotap Header* de uma trama capturada, com destaque para o campo *Antenna Signal*:

```

▼ 802.11 radio information
  PHY type: 802.11b (HR/DSSS) (4)
  Short preamble: False
  Data rate: 1,0 Mb/s
  Channel: 1
  Frequency: 2412MHz
  Signal strength (dBm): -87 dBm
  Noise level (dBm): -93 dBm
  Signal/noise ratio (dB): 6 dB
  TSF timestamp: 2851296031
  ▶ [Duration: 2344µs]

```

Figura 38: Cabeçalho 802.11 *radio information* de *frame* capturado - Questão 7.

Para cada SSID identificado anteriormente, por questões de simplicidade, **foi considerada apenas a primeira ocorrência visível nas tramas filtradas**, a fim de evitar variações temporárias que pudessem dificultar a análise. A força de sinal desta primeira ocorrência foi tomada como referência.

A tabela seguinte apresenta os SSIDs encontrados e os respectivos valores de força de sinal:

| SSID | Força do sinal (dBm) |
|------------------|----------------------|
| MEO-FCF0A0 | -87 |
| MEO-WiFi | -87 |
| MEO-9BF2A0 | -93 |
| phi_F41927C3C600 | -95 |
| Masmorra do Sexo | -91 |
| GVBRAGA_EXT | -87 |
| MEO-66DB70 | -92 |
| NOS-26F6 | -93 |
| NOS-9946_EXT | -95 |
| MEO-828830 | -94 |
| MEO-854C80 | -93 |
| NOS-52C6 | -94 |
| NOS-C8B6 | -92 |
| GVBRAGA_quarto | -94 |
| Vodafone-D0ED8A | -92 |
| GVBRAGA | -94 |
| MEO-F17570 | -94 |
| NOS-FD24 | -96 |

Tabela 3: Força de sinal dos APs detectados

Concluiu-se que os APs das redes com os SSIDs “MEO-FCF0A0” e “MEO-WiFi” apresentaram a melhor intensidade de sinal, com -87 dBm, sendo, portanto, ambos adequados para associação por parte da STA, garantindo melhor qualidade de ligação.

3.3.8 Questão 8

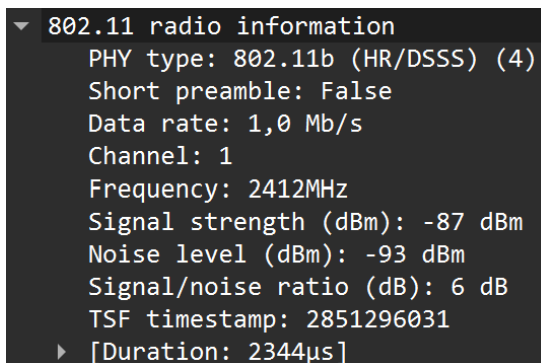
Problema: Os valores de taxa de transmissão do *Wi-Fi* estão diretamente associados à qualidade da recepção do sinal. Considerando os valores de sensibilidade mínima (*Minimum Sensivity*) e taxa de transmissão (*Data Rate*), que constam nas tabelas de referência (ver Anexo 5 e Anexo 7), e a força do sinal recebido nas tramas do AP identificado na resposta anterior, estime o débito que a STA obterá nessa ligação.

Resposta: Das duas opções de pontos de acesso que apresentaram a melhor intensidade de sinal (-87 dBm), *MEO-FCF0A0* e *MEO-WiFi*, optou-se por analisar o AP da rede *MEO-FCF0A0* para efeitos de estimativa de débito. Esta escolha foi arbitrária, uma vez que ambos apresentaram exatamente o mesmo nível de sinal e características semelhantes.

Com base na análise do *Beacon Frame* correspondente ao AP *MEO-FCF0A0*, capturado no *Wireshark*, foram identificadas as seguintes informações relevantes:

- **Força do sinal recebido:** -87 dBm (campo: *Signal strength* (dBm): -87 dBm).
- **Taxas de transmissão suportadas (*Tags Supported Rates e Extended Supported Rates*):** 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54 [Mbit/s].
- **Largura de canal:** 20 MHz. Apesar de o AP anunciar suporte a HT (802.11n) e largura de canal de 40 MHz via a tag *HT Capabilities* (45), a tag *HT Information* (61) não indica presença de canal secundário, confirmando a operação em 20 MHz.

Todos estes dados podem ser observados nas seguintes imagens:



```
▼ 802.11 radio information
  PHY type: 802.11b (HR/DSSS) (4)
  Short preamble: False
  Data rate: 1,0 Mb/s
  Channel: 1
  Frequency: 2412MHz
  Signal strength (dBm): -87 dBm
  Noise level (dBm): -93 dBm
  Signal/noise ratio (dB): 6 dB
  TSF timestamp: 2851296031
  ▶ [Duration: 2344µs]
```

Figura 39: Cabeçalho 802.11 *radio information* de *frame* capturado - Questão 8.

```

▼ Tagged parameters (229 bytes)
  ▶ Tag: SSID parameter set: "ME0-FCF0A0"
  ▼ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54, [Mbit/sec]
    Tag Number: Supported Rates (1)
    Tag length: 8
    Supported Rates: 1(B) (0x82)
    Supported Rates: 2(B) (0x84)
    Supported Rates: 5.5(B) (0x8b)
    Supported Rates: 11(B) (0x96)
    Supported Rates: 18 (0x24)
    Supported Rates: 24 (0x30)
    Supported Rates: 36 (0x48)
    Supported Rates: 54 (0x6c)
  ▶ Tag: DS Parameter set: Current Channel: 1
  ▶ Tag: Traffic Indication Map (TIM): DTIM 0 of 1 bitmap
  ▶ Tag: ERP Information
  ▶ Tag: Extended Supported Rates 6, 9, 12, 48, [Mbit/sec]
  ▶ Tag: RSN Information
  ▶ Tag: QBSS Load Element 802.11e CCA Version
  ▶ Tag: Measurement Pilot Transmission
  ▶ Tag: RM Enabled Capabilities (5 octets)
  ▼ Tag: HT Capabilities (802.11n D1.10)
    Tag Number: HT Capabilities (802.11n D1.10) (45)
    Tag length: 26
    ▶ HT Capabilities Info: 0x08ad
    ▶ A-MPDU Parameters: 0x17
    ▶ Rx Supported Modulation and Coding Scheme Set: MCS Set
    ▶ HT Extended Capabilities: 0x0000
    ▶ Transmit Beam Forming (TxBF) Capabilities: 0x00000000
    ▶ Antenna Selection (ASEL) Capabilities: 0x00
  ▼ Tag: HT Information (802.11n D1.10)
    Tag Number: HT Information (802.11n D1.10) (61)
    Tag length: 22
    Primary Channel: 1
    ▼ HT Information Subset (1 of 3): 0x08
      .... 00 = Secondary channel offset: No secondary channel (0x0)
      .... 0.. = Supported channel width: 20 MHz channel width only
      .... 1... = Reduced Interframe Spacing (RIFS): Permitted
      0000 .... = Reserved: 0x0
    ▶ HT Information Subset (2 of 3): 0x0000
    ▶ HT Information Subset (3 of 3): 0x0000

```

Figura 40: *Supported Rates* e *HT Information* de frame capturado.

Com auxílio das tabelas presentes no Anexo 5 e no Anexo 7, fornecidas pela equipa docente, podemos cruzar os dados para obter a seguinte tabela:

| Modulação | Taxa (Mbps) | Sensibilidade mínima (dBm) |
|-----------|-------------|----------------------------|
| BPSK | 6.5 | -82 |
| QPSK | 13.0 | -79 |
| QPSK | 19.5 | -77 |
| 16-QAM | 26.0 | -74 |
| 16-QAM | 39.0 | -70 |
| 64-QAM | 52.0 | -66 |
| 64-QAM | 58.5 | -65 |
| 64-QAM | 65.0 | -64 |

Tabela 4: Relação modulação - débito - sensibilidade mínima

Com base na tabela anterior, podemos determinar qual é a modulação que pode ser utilizada à medida que a força de sinal diminui, e assim fazer uma estimativa aproximada da taxa de transmissão:

A força de sinal recebida de -87 dBm está abaixo de todos os limiares de sensibilidade das modulações listadas. Isto significa que a **conexão Wi-Fi não conseguirá atingir taxas de transmissão altas**, e a modulação mais baixa disponível (BPSK com taxa de 6.5 Mbps) será utilizada.

Portanto, o débito estimado para a ligação da STA ao AP da rede *MEO-FCF0A0* será por volta de 6.5 Mbps, o que é suportado pelas taxas de transmissão apresentadas.

3.4 Processo de Associação

Numa rede *Wi-Fi* estruturada, um nó (STA – *Station*) deve associar-se a um ponto de acesso (AP – *Access Point*) antes de poder enviar dados. O processo de associação em redes IEEE 802.11 inicia-se com a transmissão de uma trama *Association Request* por parte da STA para o AP, seguida da resposta correspondente (*Association Response*) enviada pelo AP para a STA, confirmando ou recusando a associação. Este procedimento é precedido por uma fase de autenticação, que valida a identidade dos dispositivos envolvidos.

Com base na sequência de tramas capturada, procede-se à análise das tramas relativas ao processo de autenticação e associação, de forma a compreender a troca de mensagens entre a STA e o AP, bem como os parâmetros envolvidos neste processo.

3.4.1 Questão 1

Problema: 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.

Resposta: O objetivo desta questão é identificar na captura de tráfego fornecida uma sequência completa de tramas IEEE 802.11 que correspondem a um processo de **autenticação** seguido de **associação** com sucesso entre uma estação *wireless* (STA) e um ponto de acesso (AP).

Procedimento de Análise

A análise foi realizada através da ferramenta *Wireshark*, aplicando os seguintes filtros de visualização para isolar as tramas relevantes do processo de autenticação e associação:

```
1 wlan.fc.type_subtype == 0x0b || wlan.fc.type_subtype == 0x00 ||  
wlan.fc.type_subtype == 0x01
```

Wireshark

E, desta forma, foram identificadas, cronologicamente, as seguintes tramas relevantes:

| Nº da Trama | Tipo/Subtipo IEEE 802.11 | Direção | MAC de Origem | MAC de Destino | Observações Principais |
|-------------|--------------------------|----------|-------------------|-------------------|---|
| 2042 | Authentication Request | STA → AP | fe:bd:a5:05:6c:84 | 74:9b:e8:f3:9a:46 | Algoritmo: Open System (0); Sequence Nº: 0x0001 |
| 2044 | Authentication Response | AP → STA | 74:9b:e8:f3:9a:46 | fe:bd:a5:05:6c:84 | Status code: (0) Success; Sequence Nº: 0x0002 |
| 2046 | Association Request | STA → AP | fe:bd:a5:05:6c:84 | 74:9b:e8:f3:9a:46 | SSID: "Flying-Net" |
| 2048 | Association Response | AP → STA | 74:9b:e8:f3:9a:46 | fe:bd:a5:05:6c:84 | Status code: (0) Success; AID: 0x0001 |

Tabela 5: Tabela Resumo das Tramas.

Interpretação dos Resultados

A sequência de tramas confirma que:

- A estação (STA) **inicia o processo de autenticação** com o AP, utilizando o algoritmo especificado.
- O AP **responde positivamente**, permitindo o avanço para a fase seguinte.
- A STA **envia então um *Association Request***, contendo o SSID da rede pretendida e as taxas de transmissão suportadas, omitidas por serem bastantes (até 54 Mbps).
- O AP **finaliza o processo com um *Association Response***, atribuindo um AID (*Association ID*) e confirmando o sucesso da associação com *Status Code* 0.

Assim, através da análise da captura de tráfego, foi possível identificar com clareza uma sequência de tramas que representa uma associação bem-sucedida entre uma STA e um AP, incluindo a fase de autenticação. Esta sequência valida o funcionamento esperado do protocolo IEEE 802.11.

3.4.2 Questão 2

Problema: Efetue um diagrama que ilustre a sequência de todas as tramas trocadas no processo.

Resposta: A figura seguinte ilustra a sequência cronológica de trocas entre a STA e o AP durante o processo de autenticação e associação bem-sucedido.

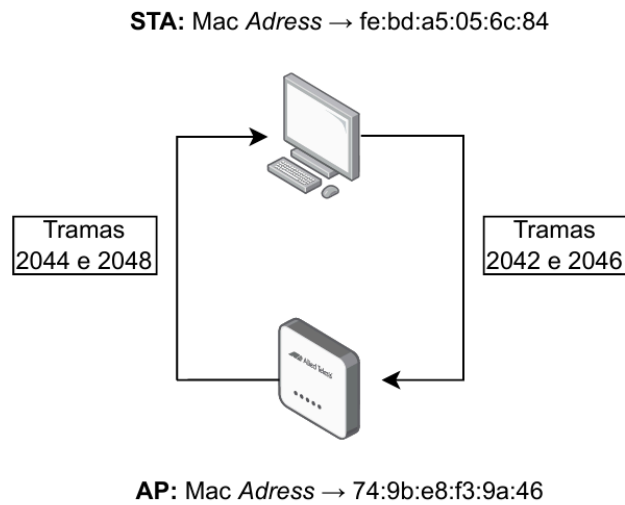


Figura 41: Diagrama de Sequência do Processo de Associação.

Descrição Textual dos Eventos

t₁ – Authentication Request:

- **Timestamp:** Apr 7, 2025 13:22:28.007861000 Hora de Verão de GMT.
- **Frame nº:** 2042.
- **Origem:** STA (MAC fe:bd:a5:05:6c:84) → **Destino:** AP (MAC 74:9b:e8:f3:9a:46).
- **Authentication Algorithm:** Open System (0).
- **Sequence Number:** 0x0001.

t₂ – Authentication Response:

- **Timestamp:** Apr 7, 2025 13:22:28.007886000 Hora de Verão de GMT.
- **Frame nº:** 2044.
- **Origem:** AP (MAC 74:9b:e8:f3:9a:46) → **Destino:** STA (MAC fe:bd:a5:05:6c:84).
- **Status Code:** 0 (Success).
- **Sequence Number:** 0x0002.

t₃ – Association Request:

- **Timestamp:** Apr 7, 2025 13:22:28.010893000 Hora de Verão de GMT.
- **Frame nº:** 2046.
- **Origem:** STA (MAC fe:bd:a5:05:6c:84) → **Destino:** AP (MAC 74:9b:e8:f3:9a:46).
- **SSID:** “FlyingNet”.
- **Supported Rates:** 1(B), 2(B), 5.5(B), 11(B), 18, 24(B), 36, 54, [Mbit/sec].

t_4 – Association Response:

- **Timestamp:** Apr 7, 2025 13:22:28.017260000 Hora de Verão de GMT.
- **Frame nº:** 2048.
- **Origem:** AP (MAC 74:9b:e8:f3:9a:46) → **Destino:** STA (MAC fe:bd:a5:05:6c:84).
- **Status Code:** 0 (*Success*).
- **Association ID (AID):** 0x0001.

O diagrama e a sua descrição textual demonstram de forma clara e sequencial as quatro tramas essenciais do processo de associação IEEE 802.11 — desde o **pedido de autenticação** até à **resposta de associação bem-sucedida**. Através da combinação do fluxo gráfico com os *timestamps* e campos-chave registados, verifica-se que a STA e o AP completam corretamente o *hand-shake*, confirmando o algoritmo de autenticação *Open System*, o código de estado *Success* e a atribuição do AID.

3.5 Transferência de Dados

O *trace* de tráfego disponibilizado, para além de incluir tramas de gestão relacionadas com o estabelecimento e manutenção da ligação de dados, contém também tramas de dados propriamente ditas, bem como tramas de controlo responsáveis pela coordenação e fiabilidade da transferência desses dados na rede IEEE 802.11.

3.5.1 Questão 1

Problema: 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 50, ou 0 caso não exista 50). Sabendo que o campo *Frame Control* contido no cabeçalho das tramas 802.11 permite especificar a direcionalidade das tramas, o que pode concluir face à direcionalidade dessa trama, será local à WLAN?

Resposta: O objetivo desta questão é determinar, através dos *bits To DS* e *From DS* no campo *Frame Control* do cabeçalho 802.11, se uma trama de dados selecionada (*Data* ou *QoS Data*) com número de *frame* a terminar em 50 se mantém apenas no domínio da WLAN ou se é encaminhada através do *Distribution System* (DS).

Para alcançarmos o objetivo desta questão, realizamos uma análise à captura de tráfego no *Wireshark*, usando o seguinte filtro:

```
1 (wlan.fc.type == 2) && (wlan.fc.subtype == 0x08  
  || wlan.fc.subtype == 0x28)
```

Wireshark

Este filtro permite isolar todas as tramas **Data** e **QoS Data**. Analisando a captura filtrada selecionamos a *frame* 1350, uma vez que é a primeira que aparece com o número do grupo no fim (PL50) e obtivemos os seguintes dados:

| Parâmetro | Valor |
|--------------------------|---|
| Frame nº | 1350 |
| Timestamp (Arrival time) | Apr 7, 2025 13:22:20.974368000 Hora de Verão de GMT |
| Tipo/Subtipo | QoS Data |
| MAC Origem | de:62:79:01:e2:39 |
| MAC Destino | ff:ff:ff:ff:ff:ff (Broadcast) |
| To DS | 1 |
| From DS | 0 |

Tabela 6: Resumo da frame 1350 para análise.

Interpretação dos *Bits* de Direcionalidade

- *To DS* = 1, *From DS* = 0:
 - **STA** → **AP**: a estação sem-fios envia a *frame* ao ponto de acesso para que ele o encaminhe no DS.

Observação Adicional (*Broadcast*)

Como o MAC de destino é ff:ff:ff:ff:ff:ff (*broadcast*), trata-se de uma *frame* de *broadcast* de nível 2. Neste caso, ao receber este *frame* com **To DS** = 1, o AP reencaminhará a trama para:

- Todos os STAs associados na mesma rede sem-fios (garantindo o *broadcast* dentro do *Basic Service Set* - BSS).
- O *Distribution System* (rede cabeada), de modo que todos os nós do domínio de *broadcast* na LAN física também o recebam (por exemplo, um pedido ARP).

Conclusão

A análise dos *bits To DS* e *From DS*, juntamente com o endereço de destino *broadcast*, confirma que o *frame* nº 1350 é enviado pela STA ao AP para distribuição ampla:

- **Não permanece apenas na WLAN.**
- É um *broadcast* de camada 2 que o AP propagará a todos os nós no domínio de *broadcast* (tanto sem-fios quanto cabeados), tipicamente utilizado por protocolos como ARP.

3.5.2 Questão 2

Problema: 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)?

Resposta: No contexto do processo de transferência de dados numa rede IEEE 802.11, cada *frame* transporta três endereços MAC distintos no seu cabeçalho, que refletem a topologia lógica entre a **estação sem fios (STA)**, o **ponto de acesso (AP)** e o **Distribution System (DS)**. A correta identificação desses endereços permite validar o percurso da *frame* desde a origem até ao destino final no sistema de distribuição.

O objetivo desta questão é transcrever os valores dos campos *Address 1*, *Address 2* e *Address 3* do cabeçalho 802.11 para o *frame* selecionado, e identificar qual dispositivo corresponde a cada endereço.

| Dispositivo | Campo no Cabeçalho IEEE 802.11 | Valor (MAC) |
|------------------------|--------------------------------|-------------------|
| Estação sem fios (STA) | Source Address | de:62:79:01:e2:39 |
| Ponto de Acesso (AP) | Receiver Address | d0:cf:0e:7f:87:74 |
| Distribution System | Destination Address | ff:ff:ff:ff:ff:ff |

Tabela 7: Tabela Resumo das Tramas.

Com base na transcrição dos três endereços do cabeçalho 802.11, verificamos que:

- A **Estação sem Fios (STA)**, com endereço MAC de:62:79:01:e2:39, é a origem da trama (indicado pelo *Source Address*).
- O **Ponto de Acesso (AP)**, com endereço MAC d0:cf:0e:7f:87:74, é o recetor imediato da trama (indicado pelo *Receiver Address*).
- O **Destination Address** da trama é ff:ff:ff:ff:ff:ff, que é o endereço de *broadcast*. Isto significa que a trama se destina a todos os dispositivos na rede local (tanto na **WLAN** quanto, potencialmente, na **LAN cabeada** através do AP).

É importante notar que o endereço de *broadcast* como *Destination Address* não significa que este seja o endereço MAC específico do *router* de acesso ao sistema de distribuição (DS). Em vez disso, indica que a informação precisa ser entregue a todos os dispositivos dentro do domínio de *broadcast*. O AP, ao receber um *broadcast* com “**To DS = 1**” (implícito pelo fluxo STA -> AP -> DS), é responsável por reencaminhá-lo tanto para as outras STAs associadas quanto para a rede cabeada, onde o *router* reside e outros dispositivos podem estar à escuta desse *broadcast*.

3.5.3 Questão 3

Problema: 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 direcionalidade das tramas e os sistemas envolvidos.

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

Resposta: Para responder à pergunta, foi necessário aplicar um novo filtro à captura de tráfego *Wireshark*, para verificar a existência de *frames* RTS/CTS próximas da *frame* escolhida (1350). O filtro em questão é o seguinte:

```
1 (wlan.fc.type_subtype == 0x1b) || (wlan.fc.type_subtype == 0x1c) || (frame.number == 1350)
```

Wireshark

A aplicação deste produziu o seguinte *output*:

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------|------------------------|------------------------|----------|--------|--|
| 1181 | 14.783459 | HitronTechno_4d:52:... | 86:94:a6:57:f2:54 | 802.11 | 76 | Request-to-send, Flags=.....C |
| 1182 | 14.786355 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1184 | 14.794368 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1185 | 14.798362 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1188 | 14.809087 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1189 | 14.815306 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1191 | 14.821503 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1192 | 14.821515 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1193 | 14.821519 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1195 | 14.829752 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1196 | 14.829866 | HitronTechno_4d:52:... | HitronTechno_4d:52:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1350 | 16.673880 | de:62:79:01:e2:39 | Broadcast | 802.11 | 116 | QoS Data, SN=487, FN=0, Flags=.p....TC |
| 1425 | 17.640774 | AMPAKTechno1_7a:9b:... | HitronTechno_f3:9a:... | 802.11 | 76 | Request-to-send, Flags=.....C |
| 1426 | 17.640778 | AMPAKTechno1_7a:9b:... | AMPAKTechno1_7a:9b:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 1584 | 19.563038 | HitronTechno_4d:52:... | 86:94:a6:57:f2:54 | 802.11 | 76 | Request-to-send, Flags=.....C |
| 1618 | 19.828204 | HitronTechno_4d:52:... | 86:94:a6:57:f2:54 | 802.11 | 76 | Request-to-send, Flags=.....C |
| 1619 | 19.828309 | HitronTechno_4d:52:... | 86:94:a6:57:f2:54 | 802.11 | 76 | Request-to-send, Flags=.....C |
| 2038 | 23.674629 | | Broadcom_04:6c:84 | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 2060 | 23.778871 | | SagemcomBroa_7f:87:... | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 2087 | 23.885320 | fe:bd:a5:05:6c:84 | HitronTechno_f3:9a:... | 802.11 | 76 | Request-to-send, Flags=.....C |
| 2088 | 23.885323 | | fe:bd:a5:05:6c:84 | 802.11 | 68 | Clear-to-send, Flags=.....C |
| 2109 | 23.885380 | fe:bd:a5:05:6c:84 | HitronTechno_f3:9a:... | 802.11 | 76 | Request-to-send, Flags=.....C |

Figura 42: Aplicação de um filtro ao *Wireshark* para visualizar *frames* RTS/CTS.

Análise do Trecho da Captura

Na imagem vê-se, imediatamente antes do *frame* 1350 (QoS Data, SN=487, destino *broadcast*), várias tramas RTS e CTS entre HitronTechno-4d:52... e 86:94:a6:57:f2:54 (*frames* 1181 – 1196). Porém, estas ocorrem com mais de 1s de antecedência ao *frame* 1350. Entre as tramas de RTS/CTS e o QoS Data n° 1350 não há *handshake* RTS/CTS imediato.

Conclusão sobre o Frame 1350

- To DS = 1, From DS = 0 → STA → AP.
- MAC de destino = ff:ff:ff:ff:ff:ff → *broadcast* de camada 2.
- Não existe um par RTS/CTS imediatamente anterior ao *frame* 1350.

Portanto, não foi usado o mecanismo RTS/CTS para o QoS Data *frame* n° 1350: a STA transmitiu diretamente em *broadcast*, confiando apenas nas regras de acesso aleatório do 802.11.

Exemplos Contrastantes

Ao analisarmos a imagem disponível acima podemos verificar duas situações contrastantes:

- **Com RTS/CTS:**
 - RTS (*frame* 1181) STA → AP.
 - CTS (*frame* 1182) AP → STA.
 - Data (*frame* 1183) STA → AP.

- **Sem RTS/CTS:**

- **Data** (*frame* 1350) STA → AP (*broadcast*).

Esse comportamento é típico: o RTS/CTS é geralmente usado para *unicast* em ambientes com risco de colisão (STAs escondidas), mas não para tramas *broadcast*.

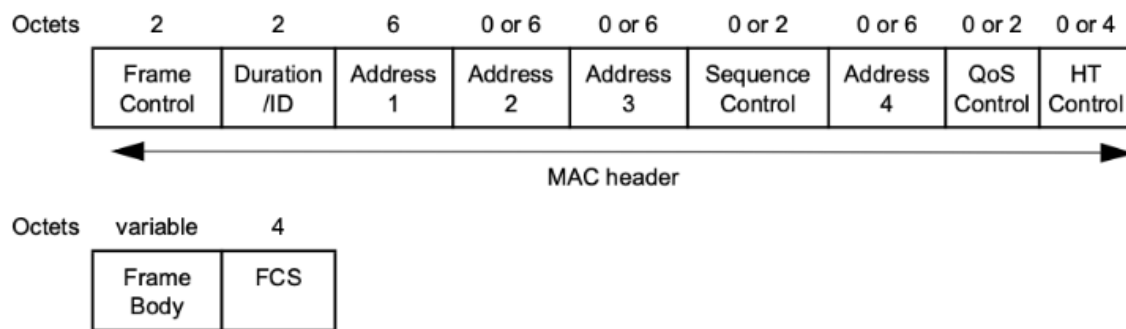
4. Conclusão

A realização deste trabalho permitiu uma aprendizagem sólida e aplicada dos princípios fundamentais do **nível de ligação lógica**, evidenciando a importância dos protocolos **ARP**, **IEEE 802.11** e **IEEE 802.3** na comunicação entre dispositivos numa **rede local**. As tarefas desenvolvidas reforçaram a capacidade de análise e interpretação de **tramas de rede**, demonstraram o papel dos **switches**, **hubs** e **routers** na interligação de dispositivos e destacaram as particularidades das **redes com e sem fios**. Esta abordagem prática contribuiu significativamente para uma visão mais clara e integrada das **camadas inferiores da pilha protocolar**, essenciais para o funcionamento da *Internet* moderna.

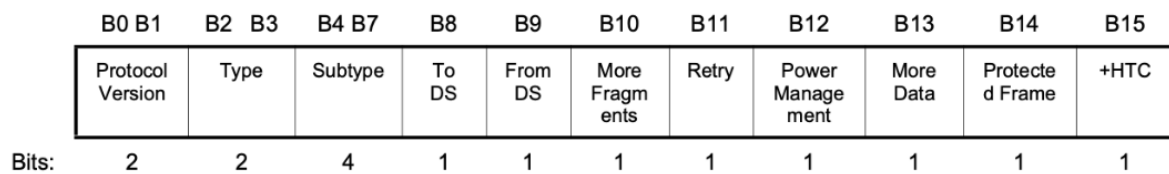
5. Bibliografia

- [1] «EtherType». [Online]. Disponível em: <https://en.wikipedia.org/wiki/EtherType>
- [2] K. W. Kurose J. F. & Ross, *Computer Networking: A Top-Down Approach*, 8.º ed. Pearson, 2020.
- [3] M. Gast, *802.11 Wireless Networks: The Definitive Guide*, 2.º ed. O'Reilly Media, 2005.
- [4] IEEE Computer Society, *IEEE Std 802.11™-2020: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2020.º ed. IEEE, 2020.

6. Anexos



Anexo 1: Formato trama MAC.



Anexo 2: Formato do campo de controlo de quadro nos PPDU S1G quando os subcampos de Tipo são iguais a 0 ou 2.

| | | | |
|----|------------|------|------------------------|
| 00 | Management | 0000 | Association Request |
| 00 | Management | 0001 | Association Response |
| 00 | Management | 0010 | Reassociation Request |
| 00 | Management | 0011 | Reassociation Response |
| 00 | Management | 0100 | Probe Request |
| 00 | Management | 0101 | Probe Response |
| 00 | Management | 0110 | Timing Advertisement |

| Type value B3 B2 | Type description | Subtype value B7 B6 B5 B4 | Subtype description |
|---------------------|---------------------|------------------------------|---------------------------------|
| 00 | Management | 0111 | Reserved |
| 00 | Management | 1000 | Beacon |
| 00 | Management | 1001 | ATIM |
| 00 | Management | 1010 | Disassociation |
| 00 | Management | 1011 | Authentication |
| 00 | Management | 1100 | Deauthentication |
| 00 | Management | 1101 | Action |
| 00 | Management | 1110 | Action No Ack |
| 00 | Management | 1111 | Reserved |
| 01 | Control | 0000–0010 | Reserved |
| 01 | Control | 0011 | TACK |
| 01 | Control | 0100 | Beamforming Report Poll |
| 01 | Control | 0101 | VHT NDP Announcement |
| 01 | Control | 0110 | Control Frame Extension |
| 01 | Control | 0111 | Control Wrapper |
| 01 | Control | 1000 | Block Ack Request (BlockAckReq) |
| 01 | Control | 1001 | Block Ack (BlockAck) |
| 01 | Control | 1010 | PS-Poll |
| 01 | Control | 1011 | RTS |
| 01 | Control | 1100 | CTS |
| 01 | Control | 1101 | Ack |
| 01 | Control | 1110 | CF-End |
| 01 | Control | 1111 | Reserved |
| 10 | Data | 0000 | Data |
| 10 | Data | 0001 | Reserved |
| 10 | Data | 0010 | Reserved |
| 10 | Data | 0011 | Reserved |
| 10 | Data | 0100 | Null |
| 10 | Data | 0101 | Reserved |
| 10 | Data | 0110 | Reserved |
| 10 | Data | 0111 | Reserved |
| 10 | Data | 1000 | QoS Data |
| 10 | Data | 1001 | QoS Data +CF-Ack |
| 10 | Data | 1010 | QoS Data +CF-Poll |

Anexo 3: Combinações válidas de tipo e subtipo.

| Type value B3 B2 | Type description | Subtype value B7 B6 B5 B4 | Subtype description |
|---------------------|---------------------|------------------------------|---------------------|
| 10 | Data | 1101 | Reserved |
| 10 | Data | 1110 | QoS CF-Poll |
| 10 | Data | 1111 | QoS CF-Ack +CF-Poll |
| 11 | Extension | 0000 | DMG Beacon |
| 11 | Extension | 0001 | S1G Beacon |
| 11 | Extension | 0010–1111 | Reserved |

Anexo 4: Combinações válidas de tipo e subtipo - Continuação.

| MCS Index | Modulation | R | $N_{BPSCS(i_{SS})}$ | N_{SD} | N_{SP} | N_{CBPS} | N_{DBPS} | Data rate (Mb/s) | |
|--|------------|-----|---------------------|----------|----------|------------|------------|------------------|-------------------------|
| | | | | | | | | 800 ns GI | 400 ns GI (see NOTE) |
| 0 | BPSK | 1/2 | 1 | 52 | 4 | 52 | 26 | 6.5 | 7.2 |
| 1 | QPSK | 1/2 | 2 | 52 | 4 | 104 | 52 | 13.0 | 14.4 |
| 2 | QPSK | 3/4 | 2 | 52 | 4 | 104 | 78 | 19.5 | 21.7 |
| 3 | 16-QAM | 1/2 | 4 | 52 | 4 | 208 | 104 | 26.0 | 28.9 |
| 4 | 16-QAM | 3/4 | 4 | 52 | 4 | 208 | 156 | 39.0 | 43.3 |
| 5 | 64-QAM | 2/3 | 6 | 52 | 4 | 312 | 208 | 52.0 | 57.8 |
| 6 | 64-QAM | 3/4 | 6 | 52 | 4 | 312 | 234 | 58.5 | 65.0 |
| 7 | 64-QAM | 5/6 | 6 | 52 | 4 | 312 | 260 | 65.0 | 72.2 |
| NOTE—Support of 400 ns GI is optional on transmit and receive. | | | | | | | | | |

Anexo 5: Parâmetros MCS para 20MHz obrigatórios, $N_{SS} = 1$, $N_{ES} = 1$.

| Symbol | Explanation |
|---------------------|--|
| N_{SS} | Number of spatial streams |
| R | Coding rate |
| N_{BPSC} | Number of coded bits per single carrier (total across spatial streams) |
| $N_{BPSCS}(i_{SS})$ | Number of coded bits per single carrier for each spatial stream, $i_{SS} = 1, \dots, N_{SS}$ |
| N_{SD} | Number of complex data numbers per spatial stream per OFDM symbol |
| N_{SP} | Number of pilot values per OFDM symbol |
| N_{CBPS} | Number of coded bits per OFDM symbol |
| N_{DBPS} | Number of data bits per OFDM symbol |
| N_{ES} | Number of BCC encoders for the DATA field |
| N_{TBPS} | Total bits per subcarrier |

Anexo 6: Símbolos usados nas tabelas de parâmetros MCS.

| Modulation | Rate (R) | Adjacent channel rejection (dB) | Nonadjacent channel rejection (dB) | Minimum sensitivity (20 MHz channel spacing) (dBm) | Minimum sensitivity (40 MHz channel spacing) (dBm) |
|------------|----------|---------------------------------|------------------------------------|--|--|
| BPSK | 1/2 | 16 | 32 | -82 | -79 |
| QPSK | 1/2 | 13 | 29 | -79 | -76 |
| QPSK | 3/4 | 11 | 27 | -77 | -74 |
| 16-QAM | 1/2 | 8 | 24 | -74 | -71 |
| 16-QAM | 3/4 | 4 | 20 | -70 | -67 |
| 64-QAM | 2/3 | 0 | 16 | -66 | -63 |
| 64-QAM | 3/4 | -1 | 15 | -65 | -62 |
| 64-QAM | 5/6 | -2 | 14 | -64 | -61 |

Anexo 7: Sensibilidade mínima do nível de entrada do recetor.