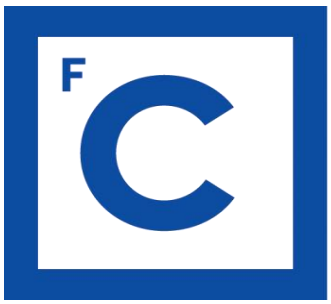


Ciências ULisboa

Faculdade de Ciências da Universidade de Lisboa

Faculdade de Ciências da Universidade de Lisboa

Departamento de Informática

Mestrado em Engenharia Informática

RELATÓRIO

Configuração e Gestão de Sistemas

Class Project: Análise de Pacotes de Rede (ARP, TCP, DNS)

Rodrigo Craveiro Rodrigues (Nº64370)

Professor: **Doutor Hugo Miranda**

2º Semestre Letivo 2024/2025

março 2025

Índice

Introdução

Este relatório apresenta uma análise detalhada de três pacotes de rede capturados utilizando a ferramenta **Wireshark**. Os pacotes analisados são um **pacote ARP** (***Address Resolution Protocol***), um **pacote TCP** com a **flag FIN ativa**, e um **pedido DNS** (***Domain Name System***). Para cada pacote, é fornecida uma **reprodução hexadecimal** completa, começando pelo nível Ethernet, seguida de uma explicação do objetivo geral do pacote e uma análise detalhada de cada campo dos cabeçalhos presentes.

A análise de pacotes é fundamental para a melhor compreensão do funcionamento das redes de computadores, pois permite observar os mecanismos de comunicação entre dispositivos, tal como ocorrem na prática.

Metodologia

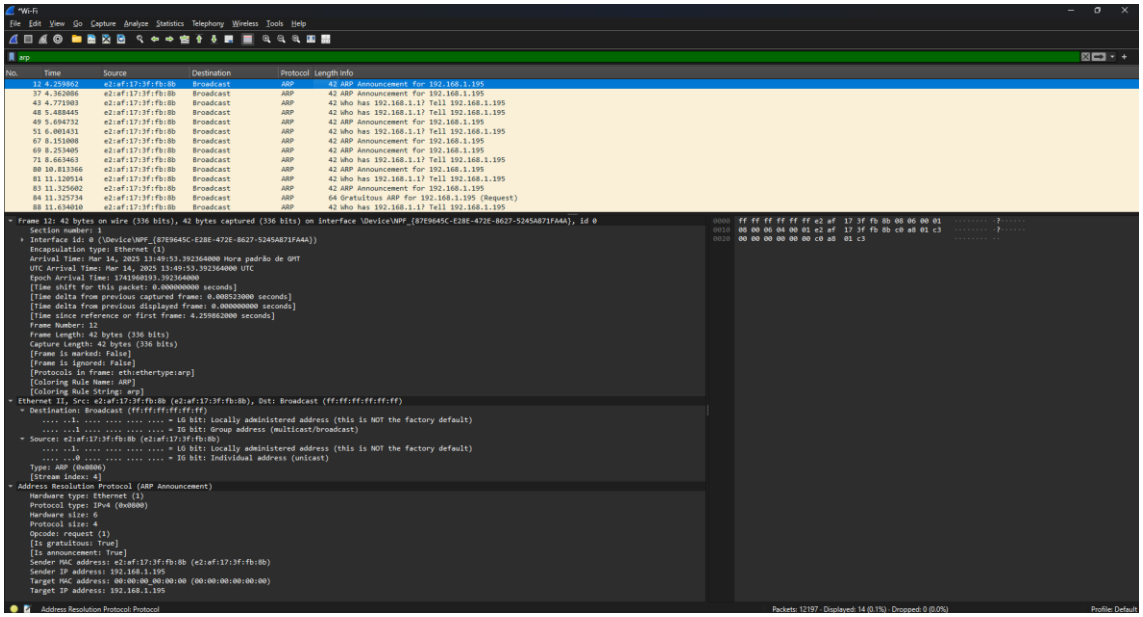
Para a realização deste trabalho, foram capturados três pacotes específicos utilizando o **Wireshark**. Os pacotes foram selecionados com base no tipo e características, de acordo com as especificações do projeto. A análise foi realizada seguindo estas etapas:

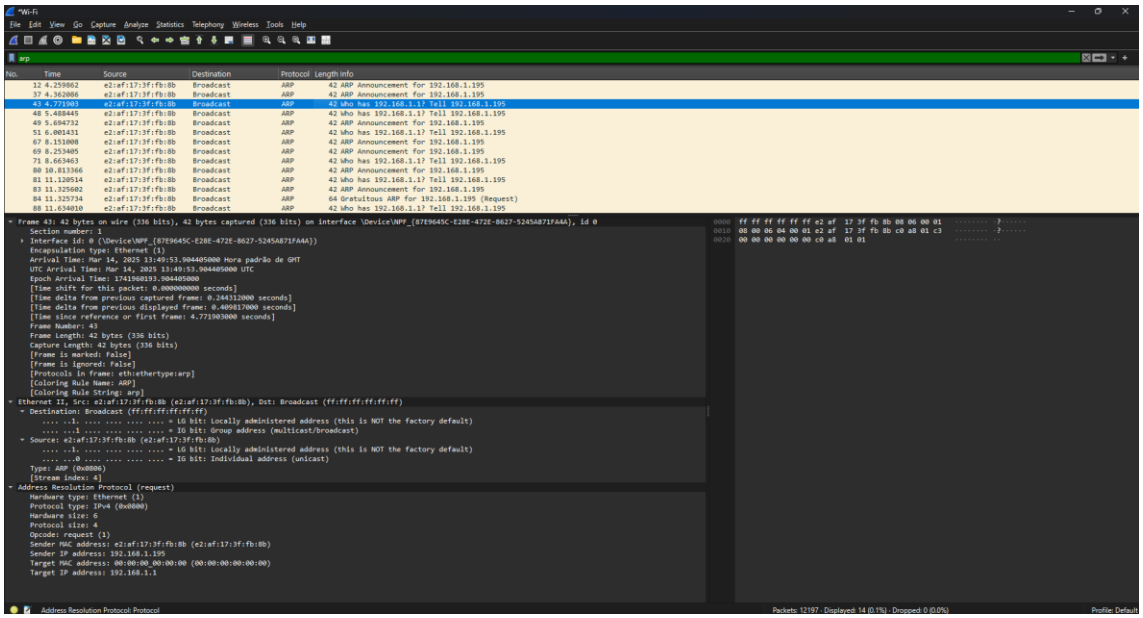
1. **Captura dos pacotes** utilizando filtros específicos para isolar os tipos de pacotes pretendidos.
2. **Exportação dos pacotes** em **formato hexadecimal** para análise detalhada.
3. **Identificação e interpretação** de cada campo presente nos cabeçalhos.
4. **Análise do objetivo geral** de cada pacote no contexto da comunicação em rede.

Análise de Pacotes

Pacote ARP

O **pacote ARP** (***Address Resolution Protocol***) tem como objetivo estabelecer uma correspondência entre **endereços IP** e **endereços MAC** (***Media Access Control***) numa rede local. No contexto específico deste pacote, trata-se de um pedido ARP onde um dispositivo está a tentar descobrir qual o endereço MAC associado a um determinado endereço IP, para que possa completar uma comunicação a nível de Ethernet.





Os pacotes capturados são principalmente anúncios ARP (também conhecidos como ARP gratuitos ou "*gratuitous ARP*"). Estes pacotes têm como objetivo:

1. Atualizar as tabelas ARP de todos os dispositivos na rede local
2. Anunciar a presença do dispositivo com MAC "*42:af:17:3f:fb:8b*" na rede
3. Verificar se há conflitos de endereços IP na rede
4. Facilitar a comunicação a nível Ethernet, associando endereços MAC a endereços IP

Na segunda imagem, também vemos um pacote ARP de solicitação (*request*), que busca descobrir o endereço MAC associado a um endereço IP específico.

Análise Detalhada dos Pacotes (Reprodução Hexadecimal)

ARP Announcement para 192.168.1.195:

ff ff ff ff ff ff e2 af 17 3f fb 8b 08 06 00 01

08 00 06 04 00 01 e2 af 17 3f fb 8b c0 a8 01 c3

00 00 00 00 00 00 c0 a8 01 c3

ARP "Who has 192.168.1.1? Tell 192.168.1.195":

ff ff ff ff ff ff e2 af 17 3f fb 8b 08 06 00 01

08 00 06 04 00 01 e2 af 17 3f fb 8b c0 a8 01 c3

00 00 00 00 00 00 c0 a8 01 01

Cabeçalho Ethernet

- **Endereço MAC de Destino (6 bytes)**: *ff:ff:ff:ff:ff:ff*
    - Valor: *Broadcast*.
    - Garante que todos os dispositivos na rede recebam e processem o pacote, aumentando a probabilidade de atualização das tabelas ARP de todos os dispositivos.
- **Endereço MAC de Origem (6 bytes)**: *42:af:17:3f:fb:8b*
    - Valor: O endereço físico do emissor.
    - Identifica inequivocamente o dispositivo que está enviando os anúncios ARP, permitindo que os dispositivos recetores saibam qual MAC associar aos endereços IP anunciados.
- **Ether Type (2 bytes)**: *08 06*
    - Valor: *0x0806* (ARP).
    - Indica ao recetor que o conteúdo deste pacote deve ser interpretado como um pacote ARP, permitindo o processamento adequado do protocolo.

Mensagem ARP

- **Tipo de Hardware (2 bytes)**: *00 01*
    - Valor: 1 (Ethernet).
    - Especifica que o protocolo de ligação física é Ethernet, definindo como os endereços MAC devem ser interpretados.
- **Tipo de Protocolo (2 bytes)**: *08 00*

- Valor: *0x0800* (endereço IPv4).
- Indica que o protocolo para o qual se está a resolver o endereço é o IPv4, estabelecendo o formato dos endereços de protocolo.

- **Comprimento do Endereço de Hardware (1 byte)**: *06*
  - Valor: 6 bytes.
  - Define o tamanho exato do campo de endereço MAC, essencial para interpretação correta dos dados.
- **Comprimento do Endereço de Protocolo (1 byte)**: *04*
  - Valor: 4 bytes.
  - Define o tamanho exato do campo de endereço IP, garantindo interpretação correta dos dados.
- **Código de Operação (2 bytes)**: *00 01* ou *00 02*
  - Valor: 1 (*Request*) ou 2 (*Reply*).
  - Determina se o pacote é uma solicitação de informação ou uma resposta/anúncio, orientando como o recetor deve processá-lo.
- **Endereço MAC do Remetente (6 bytes)**: *42:af:17:3f:fb:8b*
  - Valor: O mesmo que o endereço MAC de origem no cabeçalho Ethernet.
  - Fornece o endereço MAC que deve ser associado ao endereço IP do remetente nas tabelas ARP.
- **Endereço IP do Remetente (4 bytes)**: *c0 a8 01 xx* (*192.168.1.xx*)
  - Valor: Vários endereços da sub-rede 192.168.1.0/24, tal como observado nas imagens (*192.168.1.195*, etc.).
  - Identifica o endereço IP que está sendo associado ao endereço MAC do remetente.
- **Endereço MAC do Alvo (6 bytes)**: *00 00 00 00 00 00*
  - Valor: Zeros (endereço nulo)
  - Nos anúncios ARP, este campo é preenchido com zeros pois não se busca um endereço específico; em solicitações ARP, indica que este é o valor desconhecido sendo procurado.
- **Endereço IP do Alvo (4 bytes)**: *c0 a8 01 xx (192.168.1.xx)*
  - Valor: Em anúncios ARP, é o mesmo que o IP do remetente; em solicitações, é o IP cujo MAC está sendo procurado.
  - Para anúncios, confirma que é um ARP gratuito; para solicitações, especifica

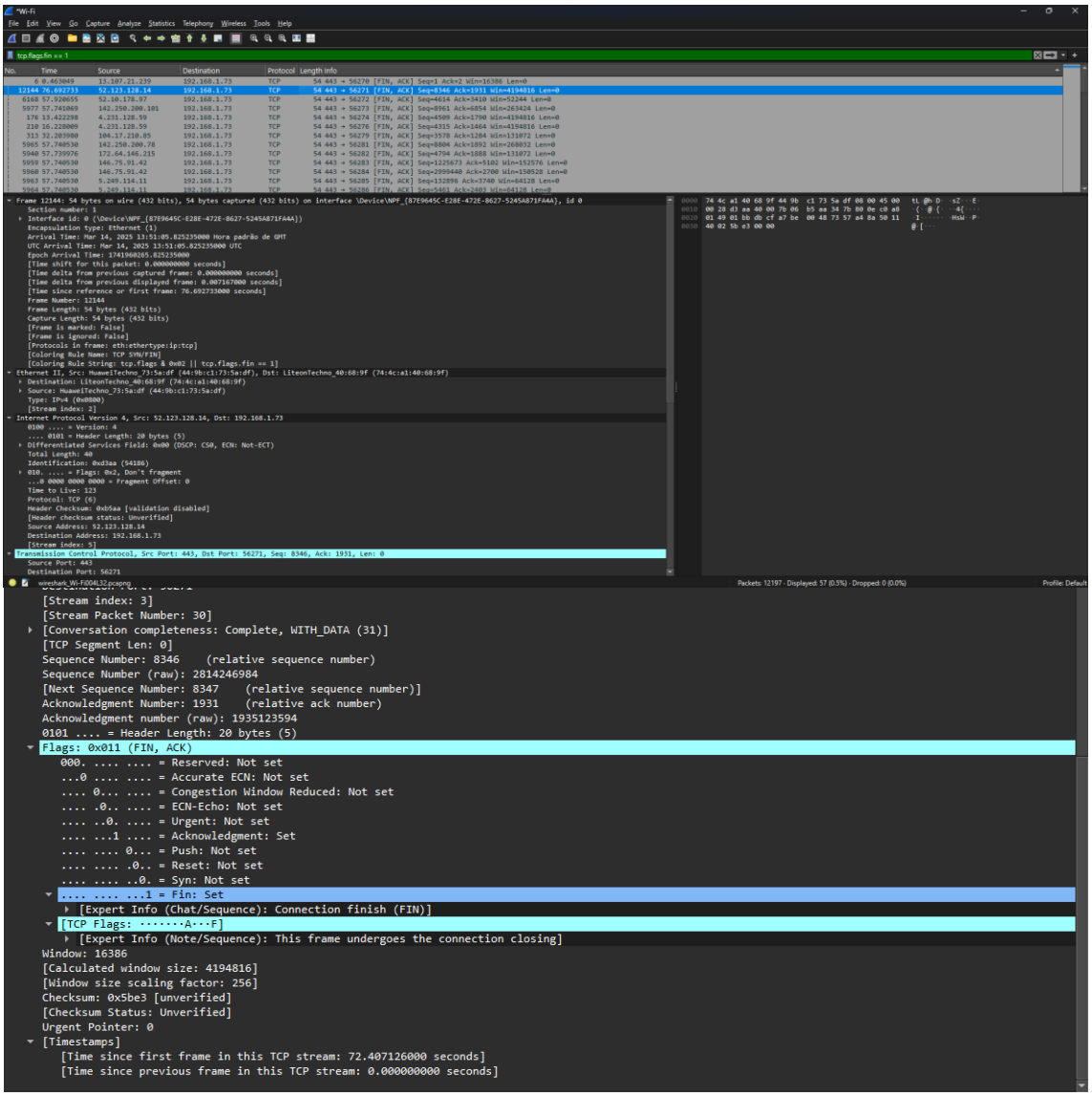
qual dispositivo deve responder com seu MAC.

Estes pacotes ARP têm duas funções distintas dentro da mesma rede local:

1. **ARP Announcement para 192.168.1.195**: O "*ARP Announcement*" é um tipo de ARP gratuito (gratuitous ARP) onde o endereço IP de origem e destino são idênticos. Isto é usado para atualizar as tabelas ARP de outros dispositivos e também pode servir para detetar conflitos de IP na rede. Este pacote tem como objetivo anunciar a presença de um dispositivo na rede. O dispositivo está a informar todos os outros dispositivos na rede local que o endereço IP "*192.168.1.195"* está associado ao endereço MAC "*E2:AF:17:3F:FB:8B*". Este tipo de anúncio é utilizado para atualizar as tabelas ARP de outros dispositivos e prevenir conflitos de endereço IP.
2. **ARP "Who has 192.168.1.1?"**: Este pacote tem como objetivo descobrir o endereço MAC associado ao endereço IP "*192.168.1.1"* (provavelmente o router/*gateway* da rede). O dispositivo com o IP "*192.168.1.195"* necessita comunicar com o *gateway*, mas precisa do seu endereço MAC para criar as tramas Ethernet. O facto de ambos os pacotes serem enviados para o endereço de *broadcast* assegura que todos os dispositivos na rede local os recebam, maximizando a probabilidade de atualização das tabelas ARP e de obtenção da resposta desejada.

Pacote TCP

O **pacote TCP com a flag FIN (Finish) ativa** é parte do processo de terminação de uma conexão TCP. A **flag FIN** indica que o emissor concluiu o envio de dados e deseja iniciar o processo de encerramento da ligação.



Análise Detalhada dos Pacotes (Reprodução Hexadecimal)

Pacote TCP:

744ca140689f449bc1735adf080045000028d3aa40007b06b5aa347b800ec0a8014901bbdbcfa7be004873 57a48a501140025be30000

Cabeçalho TCP

- **Sequence Number**: 8346 (número de sequência relativo)
    - Valor Real: 2812456964.
    - Controla a ordem dos segmentos de dados, permitindo ao recetor reorganizar pacotes que cheguem fora de ordem ou identificar pacotes perdidos.
- **Next Sequence Number**: 8347 (número relativo)

    - Valor Real: 2812456965.
    - Indica qual será o próximo número de sequência esperado, facilitando o fluxo ordenado de dados.
- **Acknowledgment Number**: 1931 (número de reconhecimento relativo)
    - Valor Real: 1535323554.
    - Confirma a receção de bytes anteriores, informando ao outro extremo que todos os dados até este número foram recebidos com sucesso.
- **Header Length**: 20 bytes (5)
    - Valor: 5 palavras de 32 bits, ou seja, 20 bytes.
    - Indica o tamanho do cabeçalho TCP, essencial para que o recetor saiba onde começam os dados.
- **Flags**: *0x11* (FIN, ACK)
    - **FIN**: Set (ativada). Indica que o emissor não tem mais dados a enviar e deseja encerrar a conexão.
    - **ACK**: Set (ativada). Confirma recebimento de dados anteriores, sendo praticamente sempre ativada em conexões estabelecidas.
    - **Reserved, Accurate ECN, Congestion Window Reduced, ECN-Echo, Urgent, Push, Reset, Syn**: Not set (Desativada). Controlam o fluxo da conexão TCP, indicando o estado atual e as ações a serem tomadas.
- **Window Size**: *16386*
    - Valor calculado real: 4194816 (considerando o fator de escala)
    - Fator de escala: 256
    - Define quantos bytes o recetor está disposto a aceitar, para controlar o fluxo de dados e prevenindo a sobrecarga.
- **Checksum Status**: *Unverified*. Verifica a integridade dos dados, embora neste caso não tenha sido verificado pelo Wireshark.
- **Urgent Pointer**: 0. Como a flag URG não está ativada, este campo não é utilizado.
- **Stream Index**: 3. Identifica esta conversa específica entre todas as conexões TCP capturadas.
- **Stream Packet Number**: 30. Indica que este é o 30º pacote desta conexão específica.
- **Conversation Completeness**: Complete, WITH_DATA (31). Confirma que a captura inclui a conexão TCP completa com dados.
- **Timestamps**:

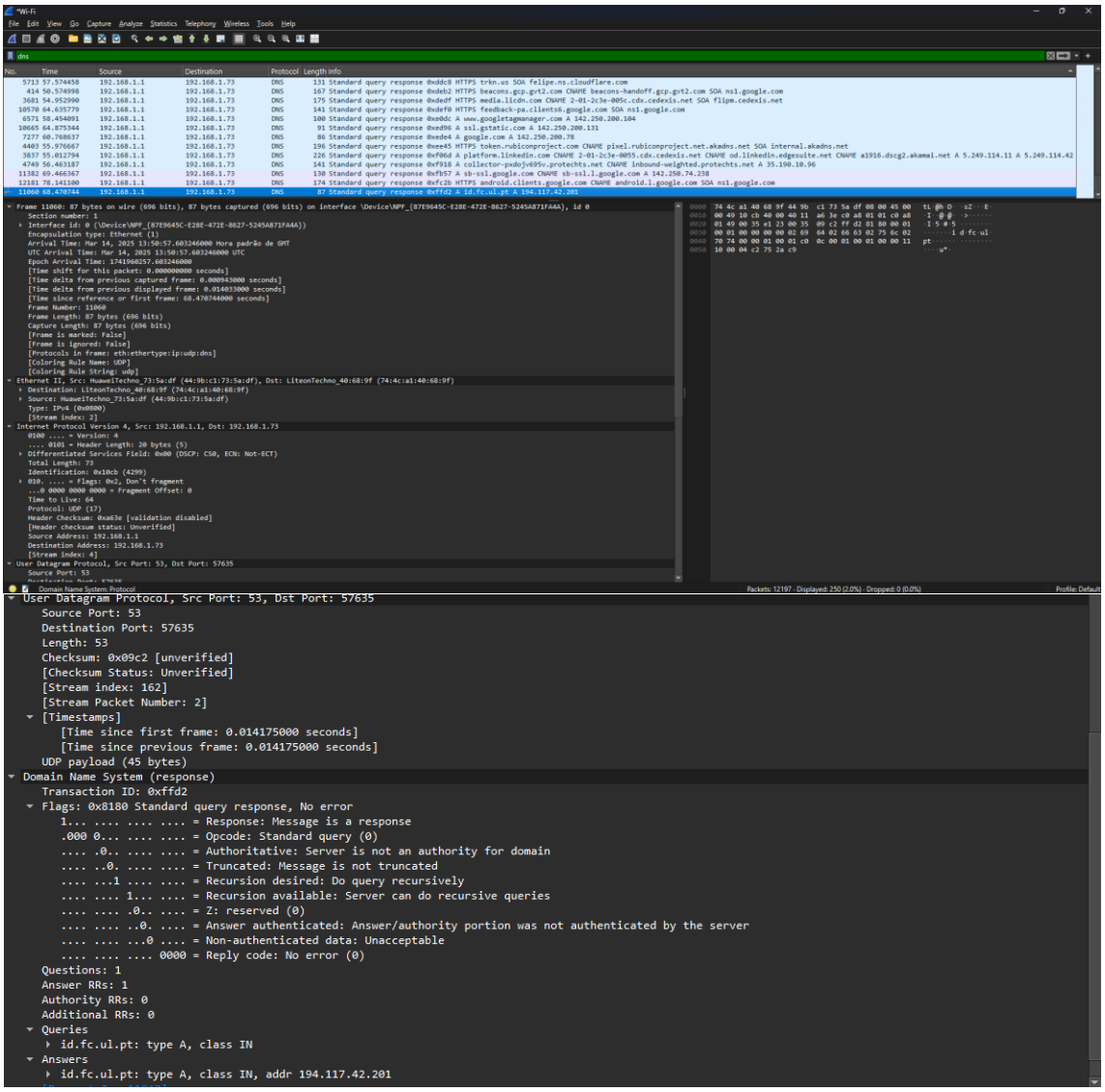
- Tempo desde o primeiro *frame*: 72.40712600 segundos
- Tempo desde o *frame* anterior: 0.00000000 segundos
- Permite análise temporal do fluxo da conexão.

Este pacote TCP (com *flag* FIN) representa um mecanismo essencial para a terminação das conexões TCP, o que garante que ambas as partes possam terminar a comunicação de forma ordenada após a transferência bem-sucedida de todos os dados.

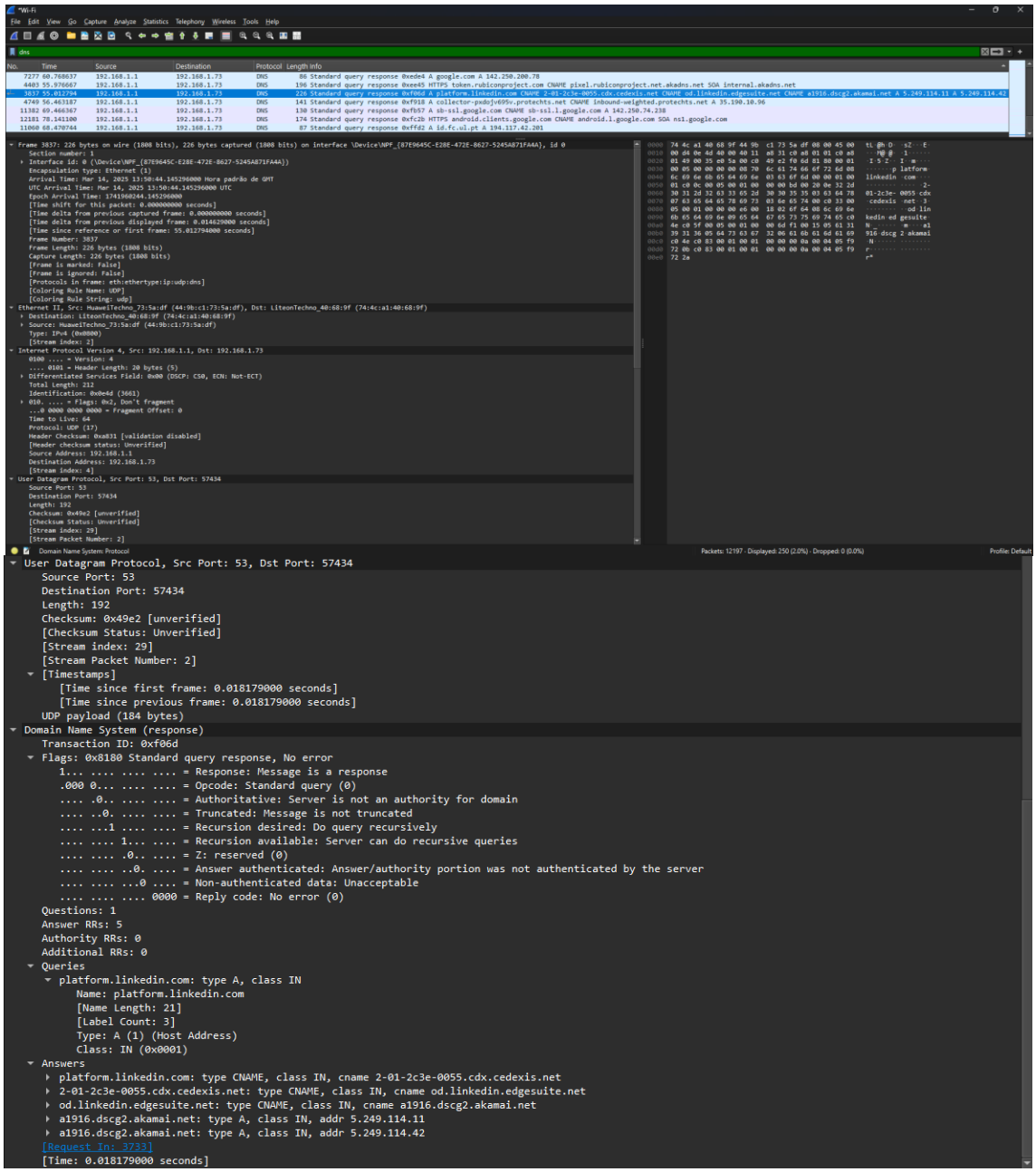
1. Neste contexto, este pacote faz parte do processo de "***handshake***" de 4 passos para encerrar uma conexão TCP de forma ordenada, garantindo que todos os dados foram corretamente entregues antes de terminar a conexão.
2. A combinação das *flags* FIN e ACK indica que o emissor está simultaneamente:
    - A solicitar o término da conexão (FIN).
    - A confirmação dos dados previamente recebidos (ACK).
3. O processo completo de terminação do TCP envolve normalmente 4 passos:
    - FIN do iniciador da terminação.
    - ACK do recetor.
    - FIN do recetor (quando estiver pronto para terminar também).
    - ACK final do iniciador.
4. A informação apresentada no pacote, representada na imagem, *"[Expert Info (Chat/Sequence): This frame undergoes the connection closing]*" confirma que este pacote está associado ao procedimento de terminação ordenado da conexão.
5. O tamanho do cabeçalho TCP de 20 bytes indica que não há opções TCP adicionais presentes neste pacote.

Pacote DNS

O **pacote DNS (*Domain Name System*)** é uma consulta enviada para um servidor DNS com o objetivo de traduzir um nome de domínio de fácil compreensão num endereço IP utilizável por máquinas. Este processo é fundamental para a navegação na Internet, pois permite aos utilizadores utilizarem nomes de domínio em vez de memorizarem endereços IP.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
No. Time Source Destination Protocol Length Info
5713 57.574458 192.168.1.1 192.168.1.73 DNS 131 Standard query response 0xddc8 HTTPS trkn.us SOA felipe.ns.cloudflare.com
414 50.574998 192.168.1.1 192.168.1.73 DNS 167 Standard query response 0xdeb2 HTTPS beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com SOA ns1.google.com
3681 54.952990 192.168.1.1 192.168.1.73 DNS 175 Standard query response 0xdedf HTTPS media.licdn.com CNAME 2-01-2c3e-005c.cdx.cedexis.net SOA flipm.cedexis.net
10570 64.635779 192.168.1.1 192.168.1.73 DNS 141 Standard query response 0xdef0 HTTPS feedback-pa.clients6.google.com SOA ns1.google.com
6571 58.454091 192.168.1.1 192.168.1.73 DNS 100 Standard query response 0xeb0c A www.googletagmanager.com A 142.250.200.104
10665 64.875344 192.168.1.1 192.168.1.73 DNS 91 Standard query response 0xed96 A ssl.gstatic.com A 142.250.200.131
7277 60.768637 192.168.1.1 192.168.1.73 DNS 86 Standard query response 0xede6 A google.com A 142.250.200.78
4403 55.576667 192.168.1.1 192.168.1.73 DNS 196 Standard query response 0xee45 HTTPS token.rubiconproject.com CNAME pixel.rubiconproject.net.akadns.net SOA internal.akadns.net
3837 55.012794 192.168.1.1 192.168.1.73 DNS 226 Standard query response 0xf06d A platform.linkedin.com CNAME 2-01-2c3e-0055.cdx.cedexis.net CNAME od.linkedin.edgesuite.net CNAME a1916.dscg2.akamai.net A 5.249.114.11 A 5.249.114.42
4749 56.463187 192.168.1.1 192.168.1.73 DNS 141 Standard query response 0xf918 A collector-pxdojy695v.protechts.net CNAME inbound-weighted.protechts.net A 35.190.10.96
11382 69.466367 192.168.1.1 192.168.1.73 DNS 130 Standard query response 0xfb57 A sb-ssl.google.com CNAME sb-ssl.l.google.com A 142.250.74.238
12181 78.141100 192.168.1.1 192.168.1.73 DNS 174 Standard query response 0xfc2b HTTPS android.clients.google.com CNAME android.l.google.com SOA ns1.google.com
11060 68.470744 192.168.1.1 192.168.1.73 DNS 87 Standard query response 0xffd2 A id.fc.ul.pt A 194.117.42.201
Frame 11060: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface \Device\NPF_{87E9645C-E28E-472E-8627-5245A871FA4A}, id 0
Section number: 1
Interface id: 0 (\Device\NPF_{87E9645C-E28E-472E-8627-5245A871FA4A})
Encapsulation type: Ethernet (1)
Arrival Time: Mar 14, 2025 13:50:57.603246000 Hora padrão de GMT
UTC Arrival Time: Mar 14, 2025 13:50:57.603246000 UTC
Epoch Arrival Time: 1741960257.603246000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000943000 seconds]
[Time delta from previous displayed frame: 0.014033000 seconds]
[Time since reference or first frame: 68.470744000 seconds]
Frame Number: 11060
Frame Length: 87 bytes (696 bits)
Capture Length: 87 bytes (696 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:udp:dns]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]
Ethernet II, Src: HuaweiTechno_73:5a:df (44:9b:c1:73:5a:df), Dst: LiteonTechno_40:68:9f (74:4c:a1:40:68:9f)
Destination: LiteonTechno_40:68:9f (74:4c:a1:40:68:9f)
Source: HuaweiTechno_73:5a:df (44:9b:c1:73:5a:df)
Type: IPv4 (0x0800)
[Stream index: 2]
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.73
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 73
Identification: 0x10cb (4299)
010. .... = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: UDP (17)
Header Checksum: 0xa63e [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.1
Destination Address: 192.168.1.73
[Stream index: 4]
User Datagram Protocol, Src Port: 53, Dst Port: 57635
Source Port: 53
Domain Name System: Protocol
Packets: 12197 · Displayed: 250 (2.0%) · Dropped: 0 (0.0%)
Profile: Default

User Datagram Protocol, Src Port: 53, Dst Port: 57635
Source Port: 53
Destination Port: 57635
Length: 53
Checksum: 0x09c2 [unverified]
[Checksum Status: Unverified]
[Stream index: 162]
[Stream Packet Number: 2]
[Timestamps]
[Time since first frame: 0.014175000 seconds]
[Time since previous frame: 0.014175000 seconds]
UDP payload (45 bytes)
Domain Name System (response)
Transaction ID: 0xffd2
Flags: 0x8180 Standard query response, No error
1... .... = Response: Message is a response
.000 0... .... = Opcode: Standard query (0)
.... .0.. .... = Authoritative: Server is not an authority for domain
.... ..0. .... = Truncated: Message is not truncated
....1 .... = Recursion desired: Do query recursively
.... 1... .... = Recursion available: Server can do recursive queries
.... .0.. .... = Z: reserved (0)
....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
....0 .... = Non-authenticated data: Unacceptable
.... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
id.fc.ul.pt: type A, class IN
Answers
id.fc.ul.pt: type A, class IN, addr 194.117.42.201



Consulta DNS para “*id.fc.ul.pt*”

O propósito deste primeiro pacote DNS é resolver o nome de domínio **"id.fc.ul.pt"** para o seu endereço IPv4 correspondente. Isto permite comunicações de rede com o servidor ao fornecer o endereço IP numérico necessário para encaminhamento. O servidor DNS responde diretamente com um único registo A, fornecendo o endereço IPv4 “*194.117.42.201*”. Este é um caso direto de resolução de nome para IP.

Consulta DNS para “*platform.linkedin.com*”

A segunda consulta, para o **domínio "platform.linkedin.com"**, é mais complexa e mostra como muitos websites utilizam várias camadas de redireccionamento. O domínio utiliza uma série de registos CNAME (aliases) que redirecionam através de múltiplos sistemas antes de resolver para endereços IP. Esta abordagem é comum para redes de distribuição de conteúdo (CDN) como a Akamai, que o LinkedIn utiliza para distribuir conteúdo. O processo completo inclui:

1. "*platform.linkedin.com*" é um CNAME que aponta para "*2-01-2c3e-0055.cdx.cedexis.net*".
2. Este, por sua vez, é um CNAME para "od.linkedin.edgesuite.net".
3. Que é outro CNAME apontando para "a1916.dscg2.akamai.net".
4. Finalmente, este último domínio resolve para dois endereços IP diferentes (*95.249.114.11* e *95.249.114.42*).

Esta resolução em múltiplas etapas demonstra como serviços grandes como o LinkedIn utilizam o DNS para implementar distribuição global de conteúdo, balanceamento de carga geográfico e tolerância a falhas através dos seus parceiros de **CDN**. Os **TTL** curtos nos registos A finais (apenas 10 segundos) permitem que a infraestrutura mude rapidamente os endereços IP para manutenção ou balanceamento de carga.

O pacote para *"platform.linkedin.com"* é consideravelmente maior (192 bytes vs. 53 bytes) em comparação com o *"id.fc.ul.pt",* devido à quantidade de informação necessária para transmitir todos os **CNAME** e endereços IP na cadeia de resolução. Esta cadeia de CNAME é uma prática comum em websites de grande escala, permitindo flexibilidade na infraestrutura sem alterar o URL que os utilizadores conhecem.

Análise Detalhada dos Pacotes (Reprodução Hexadecimal)

Consulta DNS para “id.fc.ul.pt”:

744ca140689f449bc1735adf08004500004910cb40004011a63ec0a80101c0a801490035e123003509c2ffd28180000100010000000002696402666302756c0270740000010001c00c000100010000011100004c2752ac9

Cabeçalho DNS

- **ID da Transação**: 0xffd2 - Um identificador único para associar respostas às consultas.
- ***Flags***: *0x8180*. Resposta padrão de consulta sem erro.
    - Bit de resposta está ativo (1).
    - *Opcode*: Consulta padrão (0).

    - Autoritário: Não (O servidor não é uma autoridade para o domínio).
    - Truncado: Não (A mensagem não está truncada).
    - Recursão Desejada: Sim (O cliente solicitou resolução recursiva).
    - Recursão Disponível: Sim (O servidor pode fazer consultas recursivas).
    - Z: Reservado (0).
    - Resposta Autenticada: Não (A porção de resposta/autoridade não foi autenticada pelo servidor).
    - Dados Não Autenticados: Inaceitável.
- **Questões**: 1.
- **RRs de Resposta**: 1 (Registos de Recursos na secção de resposta).
- **RRs de Autoridade e Adicionais**: 0 (Sem registos de autoridade e adicionais).

Secção de Consulta

- **Nome**: *id.fc.ul.pt.*
- **Tipo**: A (registo de endereço IPv4).
- **Classe**: IN (Internet).

Secção de Resposta

- **Nome**: *id.fc.ul.pt.*
- **Tipo**: A (registo de endereço IPv4).
- **Classe**: IN (Internet).
- **Endereço**: *194.117.42.201.*

Consulta DNS para "platform.linkedin.com":

744ca140689f449bc1735adf0800450000d40e4d40004011a831c0a80101c0a801490035e05a00c049e2f06d8180000100050000000008706c6174666f726d086c696e6b6564696e03636f6d0000010001c00c00050001000000bd00200e322d30312d326333652d3030353503636478076365646578697303 6e657400c03300050001000000e60018026f64086c696e6b6564696e096564676573756974 65c04ec05f0005000100006df10015056131393136056473636732 06616b616d6169c04ec0830001000100000000a000405f9720bc083000100010000000a000405f9722a

Cabeçalho DNS

- **ID da Transação**: *0xf06d*. Identificador único para esta consulta.

- ***Flags***: *0x8180*. Resposta padrão de consulta sem erro. Mesmos detalhes de *flags* que o pacote anterior.
- **Questões**: 1.
- **RRs de Resposta**: 5 (Múltiplas respostas devolvidas).
- **RRs de Autoridade e Adicionais**: 0.

Secção de Consulta

- **Nome**: *platform.linkedin.com.*
- **Tipo**: A (registo de endereço IPv4).
- **Classe**: IN (Internet).

Secção de Resposta

- **Primeira Resposta (registo CNAME)**:
  - **Nome**: *platform.linkedin.com.*
  - **Tipo**: CNAME (Nome Canónico).
  - **Classe**: IN.
  - **CNAME**: *2-01-2c3e-0055.cdx.cedexis.net.*
- **Segunda Resposta (registo CNAME)**:
  - **Nome**: *2-01-2c3e-0055.cdx.cedexis.net*
  - **Tipo**: CNAME.
  - **Classe**: IN.
  - **CNAME**: *od.linkedin.edgesuite.net.*
- **Terceira Resposta (registo CNAME)**:
  - **Nome**: *od.linkedin.edgesuite.net.*
  - **Tipo**: CNAME.
  - **Classe**: IN.
  - **CNAME**: *a1916.dscg2.akamai.net.*
- **Quarta Resposta (registo A)**:
  - **Nome**: *a1916.dscg2.akamai.net.*
  - **Tipo**: A.
  - **Classe**: IN.
  - **Endereço**: *5.249.114.11.*
- **Quinta Resposta (registo A)**:
  - **Nome**: *a1916.dscg2.akamai.net.*
  - **Tipo**: A.
  - **Classe**: IN.

- **Endereço**: *5.249.114.42.*

Conclusão

Ao concluir com sucesso a análise dos três tipos de pacotes de rede sugeridos no enunciado (ARP, TCP e DNS) permitiu compreender os mecanismos fundamentais de comunicação utilizados na Internet.

O pacote **ARP** demonstra como os dispositivos numa rede local descobrem os endereços físicos uns dos outros, o que permite a comunicação ao nível da camada de ligação de dados. O pacote **TCP** com a ***flag* FIN** ilustra o processo de terminação ordenada de ligações, componente crucial do protocolo TCP que garante a fiabilidade das comunicações. Por fim, o pacote **DNS** exemplifica como os nomes de domínio são convertidos em endereços IP, tornando a Internet mais acessível para os utilizadores.

Esta análise revela a complexidade dos diferentes tipos protocolos de rede, onde cada campo dos cabeçalhos tem uma função específica e contribui para o objetivo geral do pacote. Compreender estes detalhes é fundamental para profissionais de redes, segurança informática e no desenvolvimento de software, pois permite diagnosticar com mais precisão problemas, otimizar desempenho e implementar novas funcionalidades em sistemas de redes.

Referências

1. Moodle de Configuração e Gestão de Sistemas (Ano Letivo 2024/2025): https://moodle.ciencias.ulisboa.pt/course/view.php?id=5723
2. Documentação oficial do Wireshark: https://www.wireshark.org/docs/

Anexos

Anexo A: Configuração do Ambiente do Wireshark

- **Versão do Wireshark utilizada:** 4.4.5 (v4.4.5-0-g47253bcf3773).
- **Interface de rede:** WI-FI.
- **Filtros utilizados:**
    - Para ARP: *arp*
    - Para TCP com FIN: *tcp.flags.fin == 1*
    - Para DNS: *dns*

Anexo B: Glossário

- ***ARP (Address Resolution Protocol)***: Protocolo utilizado para mapear um endereço IP para um endereço MAC numa rede local.
- ***TCP (Transmission Control Protocol)****:* Protocolo de transporte orientado à conexão que garante a entrega fiável dos dados.
- ***DNS (Domain Name System)****:* Sistema de tradução de nomes de domínio em endereços IP.
- ***Flag FIN****:* Sinalização utilizada no protocolo TCP para indicar que o emissor terminou o envio de dados.
- ***MAC (Media Access Control):*** Endereço físico único atribuído a cada interface de rede.
- ***TTL (Time To Live):*** Campo no cabeçalho IP que limita o tempo de vida de um pacote na rede.
- ***Checksum****:* Valor calculado para verificar a integridade dos dados transmitidos.
- ***Port****:* Número que identifica uma aplicação ou serviço específico num dispositivo.
- ***Fragmentation***: Processo de dividir um pacote IP em partes menores para atravessar redes com diferentes MTUs *(Maximum Transmission Units).*
- ***Broadcast****:* Transmissão de dados para todos os dispositivos numa rede.