
# Rede de Computadores

Relatório

## RCOM - LEIC

João Gigante (up202008133@up.pt)
Ricardo Cavalheiro (up202005103@up.pt)

# Sumário

Neste relatório consta o segundo trabalho prático desenvolvido para a cadeira de redes de computadores. Este consistiu em 2 partes: uma primeira, no desenvolvimento de uma aplicação de transferência, e uma segunda na configuração e estuda de uma rede.

O projeto foi apresentado com sucesso, uma vez que a demonstração passou por todos os testes sem falhas.

# PARTE I

Esta primeira parte consistiu no desenvolvimento de uma aplicação de 'download', que permite transferir um ficheiro de qualquer tipo, de um dado servidor FTP (*File Transfer Protocol)*.
Para dar uso à aplicação é necessário como argumento um URL do tipo:
**ftp://<user>:<password>@<host>/<url-path>**. Os argumentos user e password são facultativos no caso de o objetivo for entrar como utilizador *anonymous.* Nesse caso, o URL ficava:
**ftp://<host>/<url-path>**. A implementação desta aplicação envolveu o conhecimento, principalmente, do RFC959, protocolo sobre a transferência de ficheiros, e do RFC1738, que aborda o tratamento de informação proveniente de URLs.

## Arquitetura

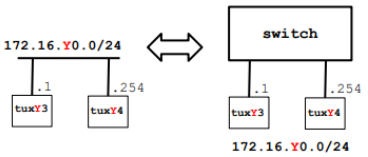
Começamos por analisar o URL, dividi-lo pelas respetivas variáveis *user*, *password*, *host*, *file_path*. Isto é feito com recurso à função **parseURL**, a função obtém também o *file_name* a partir do *file_path*. Obtemos o IP do *host* através do **getIP**, função previamente fornecida, e a porta da conexão a ser estabelecida é sempre a 21.
O programa começa por estabelecer a ligação ao servidor, com a criação de um socket que servirá de meio de comunicação entre o cliente e o servidor.
A seguir são enviadas as mensagens para o servidor de modo a dar 'login' (*user* e *password*) e ativar o modo passivo. No modo passivo, o cliente liga-se ao servidor.
Após cada mensagem enviada é lida a sua resposta e confirmada com o código de aceitação.
A seguir, é criada uma ligação a partir dos dados da resposta à mensagem anterior. Na resposta podemos obter a porta e o endereço IP onde vai ser estabelecida a conexão.
É em seguida enviado o comando *retr filepath* na primeira conexão para pedir o ficheiro e é seguidamente feito o 'download' do ficheiro com recurso à função **saveFile**. No final, são fechadas as conexões.

## Resultados

O nosso programa foi testado para diversos cenários. A utilização de 'login', assim como em modo anónimo; ficheiros grandes e pequenos; ficheiros onde a sua localização é no diretório root (*/file_name*) e ficheiros onde se encontram noutros níveis (*/dir/dir/file_name*). Durante a transferência são impressos os vários comandos enviados assim como as suas respostas para um melhor acompanhamento. Podemos ver um 'download' bem-sucedido na figura 14.

# PARTE II

## Experiência 1 - Configurar uma rede de IPs




» <u>O que são os pacotes ARP, e para o que são usados?</u>
O Address Resolution Protocol é um protocolo de rede usado para **mapear o endereço IP de um dispositivo para seu endereço MAC** correspondente numa rede local. Ele é usado para facilitar a comunicação entre dispositivos numa LAN, permitindo que os dispositivos traduzam os endereços IP de outros dispositivos nos seus endereços MAC correspondentes, necessários para enviar pacotes.

» <u>Quais são os endereços MAC e IP dos pacotes ARP e porquê?</u>
Os endereços MAC e IP são usados em rede para **identificar dispositivos numa rede**. O endereço MAC é um **identificador exclusivo** para cada interface de rede num dispositivo usado para identificar o dispositivo numa rede local.
O endereço IP é um rótulo numérico atribuído a cada dispositivo conectado a uma rede de computadores que usa o Protocolo de Internet para comunicação.
Os pacotes ARP são usados para **mapear o endereço MAC de um dispositivo para o endereço IP** respetivo ou para mapear o endereço IP de um dispositivo para seu endereço MAC. É necessário porque os dispositivos numa rede comunicam-se usando endereços MAC, mas os endereços IP são usados para identificar os dispositivos e rotear o tráfego entre eles.
É possível observá-los na figura 1 que correspondem aos logs do Wireshark dos pacotes ARP.
Os respetivos MACs das máquinas são:

- tux3 - **00:22:64:a7:32:ab**;
- tux4 - **00:21:5a:5a:75:bb**;

» <u>Que pacotes gera o comando ping?</u>
O comando ping, caso a tabela ARP esteja vazia, **gera primeiro pacotes ARP** para descobrir o MAC do IP correspondente e, a seguir, **envia pacotes ICMP** (Internet Control Message Protocol). São utilizados para testar a conectividade das máquinas na rede.

» <u>Quais são os endereços MAC e IP dos pacotes ping?</u>
Os pacotes ping ficam com o endereço IP e MAC da máquina que envia o ping e com o endereço IP e MAC do destinatário, como é possível observar nas figuras 2 e 3 que são os logs do Wireshark do **ping request** e **ping reply** respetivamente.

» <u>Como determinar se a trama Ethernet recetora é ARP, IP, ICMP</u>?
É possível determinar observando o header da trama:

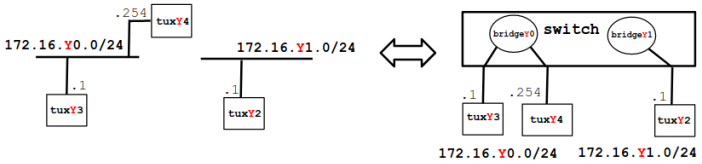
- Em relação às tramas do tipo IP (**0x0800**) (Figura 6) podemos distingui-las analisando o campo Protocol no header: no caso de ser 1 trata-se do tipo ICMP, caso contrário é uma trama IP;

- Caso o Type seja **0x0806** (Figura 7) trata-se de uma trama do tipo ARP;

» Como determinar o comprimento da trama recebida?
É possível determinar o comprimento da trama observando a **Frame Length** no Wireshark (Figura 7).

» O que é a interface loopback e qual é a sua importância?
A interface loopback é uma interface virtual que permite à máquina **enviar respostas para si próprio**, para verificar se a rede está corretamente configurada.

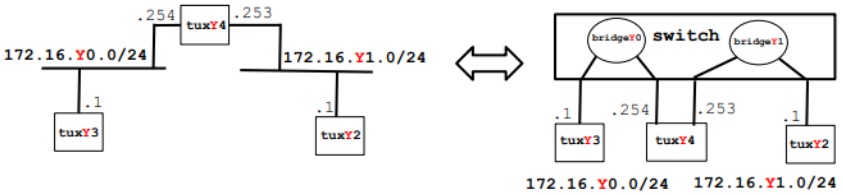
## Experiência 2 - Implementar duas bridges num switch




» Como se configura a bridgeY0?
Para configurar a bridgeY0, utilizando o GTKTERM configurado, **criamos a bridge** (*/interface bridge add name=bridgeY0*), a seguir **removemos as máquinas da bridge default** (*/interface bridge port remove [find interface=etherXX]*) **e conectamos na nossa bridge** (*/interface bridge port add interface=etherXX bridge=bridgeY0*). Cada máquina tem um cabo de rede que conecta ao switch na porta XX.

» Quantos domínios de broadcast existem? Como podes concluir isso a partir dos logs?
Existem **dois domínios de broadcast**. Um por cada sub-rede, analisamos isto através de pings broadcasts. No tux3, um ping broadcast, apenas obtém pacotes ICMP do tux4, e no tux2, um ping broadcast, não recebe estes pacotes de nenhuma outra máquina (figuras 4 e 5).

## Experiência 3 - Configurar um router em Linux




» Que rotas existem nos tuxs? Qual significado têm?
Nesta experiência tínhamos que **transformar o tux4 num router**. Para tal, precisamos de configurar as rotas necessários. Primeiro configuramos os ips das máquinas:
Máquina 2 → ifconfig eth0 172.16.Y1.1/24
Máquina 3 → ifconfig eth0 172.16.Y0.1/24
Máquina 4 → ifconfig eth0 172.16.Y0.254/24
Máquina 4 → ifconfig eth1 172.16.Y1.253/24
Seguidamente, adicionamos uma rota a cada um das seguintes máquinas, máquina 3 e máquina 2:
Máquina 3 → route add default gw 172.16.Y0.254

Máquina 2 -> route add default gw 172.16.Y1.253
Estas rotas permitem que quando uma das máquinas quer enviar um pacote para outra máquina fora da sua sub-rede, **o pacote é enviado para a route default** que adicionamos. A route default encaminha o pacote para a outra máquina.

» Que informação contém uma entrada da forwarding table?
Cada entrada na tabela possui a seguinte informação (Figura 9):

- **Destination** - endereço da rede de destino para o qual os pacotes devem ser enviados;
- **Gateway** - o endereço do próximo salto pelo qual os pacotes devem ser encaminhados;
- **Genmask** - máscara da rede respetiva;
- **Flags** - flags que indicam o estado da rota;
- **Metric** - um valor que representa o custo ou a distância até o destino;
- **Ref** - o número de vezes que a entrada foi usada ou referenciada;
- **Use** - contador de pesquisas da rota;
- **Interface** - a interface pelo qual os pacotes devem ser enviados;

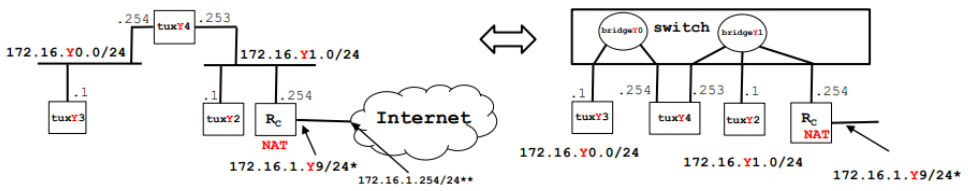
» Que mensagens ARP e endereços MAC associados são observados e porquê?
Com as rotas definidas, a tux3 passa a conseguir comunicar com as restantes (**172.16.Y0.254** ,**172.16.Y0.254, 172.16.Y1.1**).
Quando a máquina faz o ping, envia primeiro um pacote ARP com o seu MAC em broadcast para a rede, pedindo o MAC da máquina associada àquele IP. A máquina com o IP respetivo é a única que responde enviando um pacote ARP com o seu MAC que foi solicitado. Os respectivos MAC's das máquinas são:

- tux2 - **00:22:64:a3:43:bc;**
- tux3 - **00:22:64:a7:32:ab**;
- tux4 - **00:21:5a:5a:75:bb**;

» Que pacotes ICMP são observados e porquê?
Após a máquina descobrir os MAC's associados aos IP's, é possível observar **os pacotes ICMP de request e reply** que dão a entender que as **máquinas se conseguem comunicar entre si**. No caso de não se reconhecerem o pacote ICMP seria do tipo **Host Unreachable**.

» Quais são os endereços IP e MAC associados aos pacotes ICMP e porquê?
Os endereços IP e MAC associados aos pacotes ICMP são relativos ao tux de origem, ou seja, o que enviou o pacote e ao tux de destino.

## Experiência 4 - Configurar um router comercial e implementar NAT



» Como se configura uma rota estática num router comercial?
De forma a configurar o router foi necessário **ligar a interface eth1 do routerboard à rede** do netlab (PY.1). Introduz-se então o seguintes comando na interface do route:
/ip address add address=172.16.2.X9/24 interface=ether1
Liga-se então a interface eth2 do routerboard a uma porta do switch e **adicionamos a porta à bridgeY1**. Depois configuramos os ips no route, através de:
/ip address add address=172.16.X1.254/24 interface=ether2
Finalmente configuramos as routes:

- Na máquina 3 (adicionar máquina 4 como default router):
  route add default gw 172.16.X0.254
- Na máquina 2 e 4 (adicionar RC as default router):
  route add default gw 172.16.X1.254
- Criamos a rota estáticas no router para pacotes enviados na rede serem enviado para o tux4:
  /ip route add dst-address=0.0.0.0/0 gateway=172.16.2.254 (ONLY FOR LAB I320)
  /ip route add dst-address=172.16.X0.0/24 gateway=172.16.X1.253

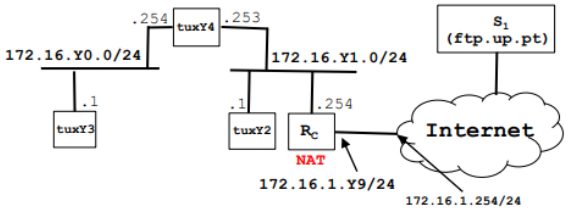
» Quais são os caminhos seguidos pelos pacotes nas experiências efetuadas e porquê?
Com as rotas configuradas, **podemos observar que os pacotes têm como destino uma delas**, eles seguem as rotas, caso contrário **são enviadas para o router para ser redirecionado** posteriormente. Para experimentarmos, ativamos os redirects na máquina 2 e retiramos a route para 172.16.Y0.0/24 via tuxY4. Para isso tivemos que ativar os redirects com:

echo 0 > /proc/sys/net/ipv4/conf/eth0/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects

» Como configurar o NAT num router comercial?
O **NAT vem ativado por default**, contudo se não viesse era necessário correr o seguinte comando: **/ip firewall nat enable 0**

» Qual é a função do NAT?
NAT (Network Address Translation) é um método usado para **permitir que dispositivos em uma rede privada acessem a Internet** usando um único endereço IP público. O NAT funciona **traduzindo os endereços IP privados dos dispositivos na rede local para um endereço IP público** quando eles enviam ou recebem dados pela Internet e convertendo o endereço IP de destino nos pacotes de resposta de volta para o endereço IP privado do dispositivo. A NAT opera num router estabelecendo a ligação entre a rede local e a Internet.

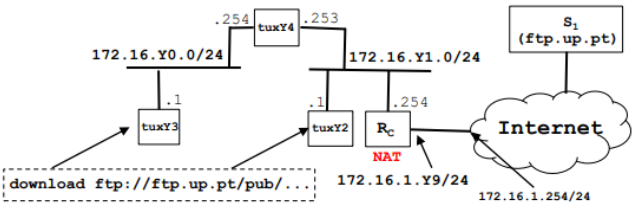
## Experiência 5



» Como configurar o serviço DNS num host?
Para configurar o serviço DNS é necessário configurar adicionar o servidor **nameserver 172.16.2.1** no ficheiro ***resolv.conf*** localizado na pasta /etc.
Após isso já é possível pingar hostnames como www.google.com.

» Que pacotes são trocados pelo DNS e que informação é transportada?
É enviado para o servidor um **pacote com o hostname desejado**. O servidor responde com um pacote contendo o **IP associado àquele hostname**.

## Experiência 6




» Quantas conexões TCP foram abertas pela aplicação FTP?
Durante o decorrer da nossa aplicação são abertas **duas conexões TCP**. Uma primeira para **entrar em contacto com o servidor e enviar comandos FTP** pelo cliente para o servidor, obtendo as respostas. A segunda tem como objetivo fazer a transferência do ficheiro em si, ou seja, receber os dados provenientes do servidor.

» Em qual conexão é transportado o controlo de informação FTP?
O controlo da informação é transportado na **primeira conexão FTP**, que é a responsável pelo **envio de pedidos ao servidor**, sendo por isso a que trata de enviar toda a informação necessária para que a transferência ocorra sem problemas.

» Quais são as fases de uma conexão TCP?
Uma conexão TCP está divida em 3 fases:

- Estabelecimento da conexão: é o processo que se estabelece a conexão entre dois dispositivos a partir de uma série de comandos;
- Transferência de dados: fase em que os dados são transferidos do servidor para o dispositivo;
- Encerramento da conexão: é o processo em que se termina a conexão a partir novamente de uma série de comandos;

» Como funciona o mecanismo ARQ TCP? Quais os campos TCP relevantes? Qual a informação relevante observada nos logs?

ARQ é um mecanismo usado pelo TCP para garantir a entrega fiável de dados numa rede. Ele funciona da seguinte forma: o remetente envia uma mensagem com um número de sequência e o destinatário confirma se a recebeu com uma mensagem de confirmação (ACK) que inclui o próximo número de sequência esperado. Se o remetente não receber a mensagem de confirmação do destinatário dentro do tempo esperado, ele irá retransmitir a mensagem.

Os campos TCP relevantes para o mecanismo ARQ são:

- **Sequence number**: é um número atribuído a cada byte de dados transmitidos pela conexão. O *sequence number* é usado pelo recetor para determinar a ordem em que os bytes foram transmitidos e é incluído na mensagem ACK para indicar ao remetente quais bytes foram recebidos com sucesso.

- **Acknowledgement number**: Este é um número que está incluído na mensagem ACK e é usado para indicar ao remetente quais bytes foram recebidos com sucesso. É igual ao número de sequência do próximo byte de dados esperado.

- **Window size**: Este é um valor que está incluído nas mensagens SYN e ACK e indica a quantidade máxima de dados que o remetente pode transmitir antes de receber um ACK. Ele é usado para controlar o fluxo de dados e evitar que o remetente sobrecarregue o destinatário com muitos dados de uma só vez.

- **Checksum**: É um valor calculado que é para cada segmento TCP para garantir a integridade dos dados. É calculado com base nos valores dos campos do cabeçalho TCP e na carga útil de dados do segmento.

Ao usar esses campos, o TCP pode implementar o mecanismo ARQ e garantir a entrega confiável de dados em uma rede.

» Como é que o mecanismo de controlo de congestão TCP funciona? Como é que o fluxo de dados da conexão evoluiu ao longo do tempo? Está de acordo com o mecanismo de controlo de congestão TCP?

Para controlar o congestionamento, o TCP usa uma **congestion window** no lado do remetente, que indica a **quantidade máxima de dados que podem ser enviados** antes de receber um ACK. Quando o **congestionamento na rede aumenta, a congestion window diminui**.
Verificamos que no início da primeira transferência a taxa de transmissão aumentou rapidamente, tendo estabilizado num valor elevado. Por outro lado, após a segunda transferência ter sido iniciada, o congestionamento na rede aumentou e por consequência a taxa de transferência diminuiu.
No início houve também uma fase de **slow start**, em que é inicialmente determinada a transmissão máxima do remetente e por isso a taxa de transmissão vai aumentando até atingir esse *threshold*.
Estas alterações vão de encontro ao mecanismo de controlo do congestionamento, uma vez que quando havia apenas um download a ser feito o **bitrate era superior** do que quando os dois downloads estavam a ser feitos ao mesmo tempo.

» De que forma é afetada a conexão de dados TCP pelo aparecimento de uma segunda conexão TCP? Como?

Ao surgir uma segunda conexão TCP, o que acontece é que, como a taxa de transferência é dividida igualmente por cada ligação, existe uma queda na taxa de transmissão dos pacotes.

## Conclusões

A realização deste trabalho permitiu-nos aprender sobre o conceito de comunicação cliente-servidor usando o protocolo TCP/IP, incluindo o conhecimento de protocolos de aplicação e o comportamento do FTP. Também nos permitiu adquirir habilidades práticas, como ler RFCs, implementar um cliente FTP em C, usar sockets e TCP na programação C e compreender e usar o DNS dentro de um programa cliente.

A segunda parte do trabalho consistiu em construir uma rede, o que permitiu o conhecimento sobre o funcionamento de bridges, um router board e um switch. Trabalhamos também com a ligação de conexões, assim como técnicas como o NAT e o DNS.

De forma sucinta, os objetivos do trabalho foram concluídos com sucesso, o que permitiu aos elementos do grupo um aprofundamento, tanto teórico como prático, e um melhor entendimento sobre a estruturação e funcionamento de um protocolo de uma rede.

# Anexo I

Todo o código e protocolo no setup está presente em: [Github Repositório]

# Anexo II

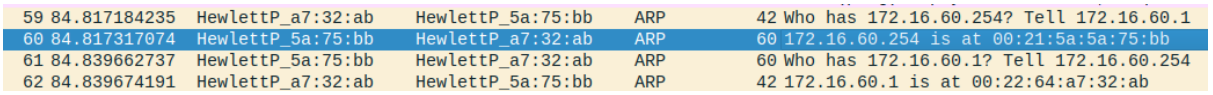



Figura 1

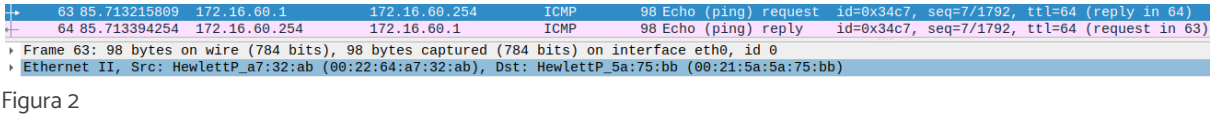



Figura 2

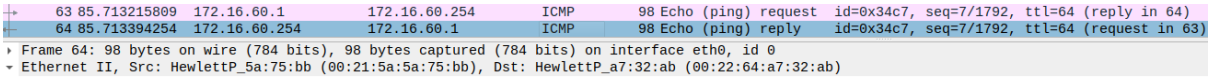



Figura 3

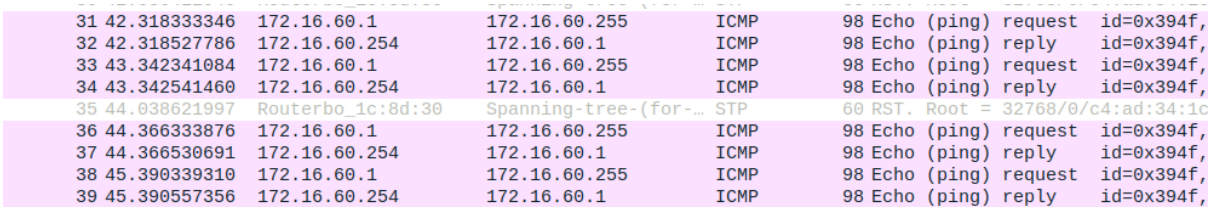



Figura 4

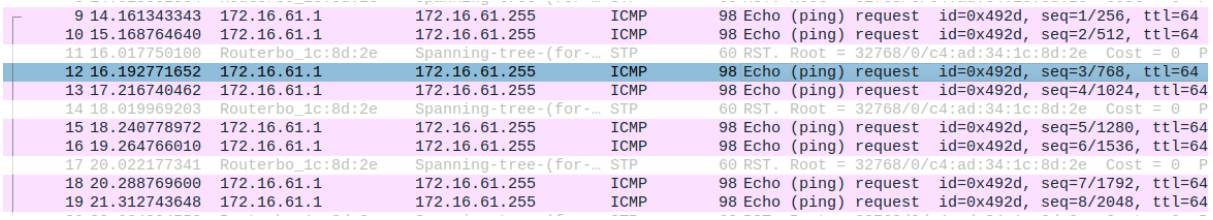



Figura 5

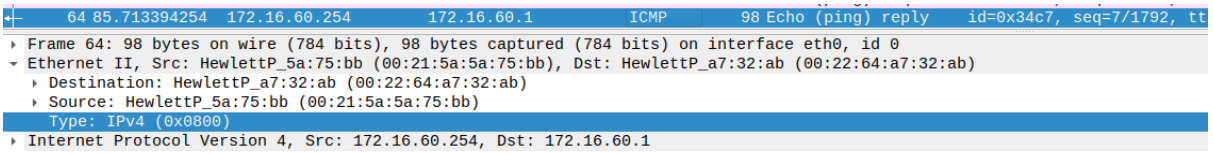



Figura 6

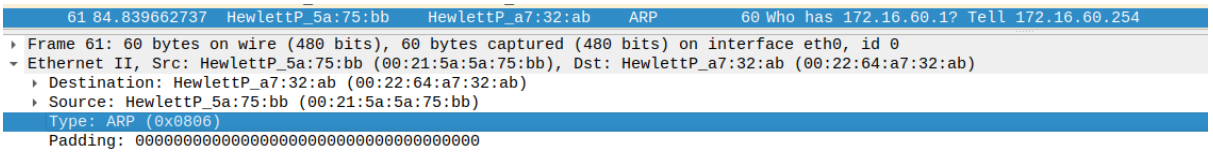



Figura 7

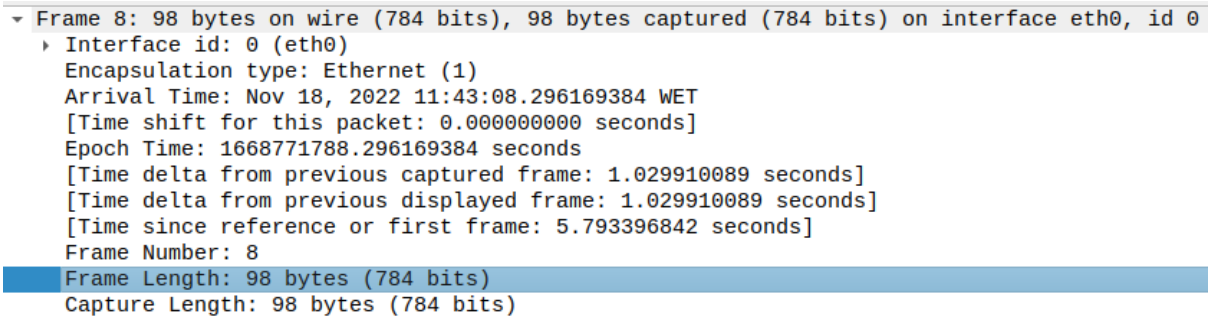



Figura 8


Figura 9

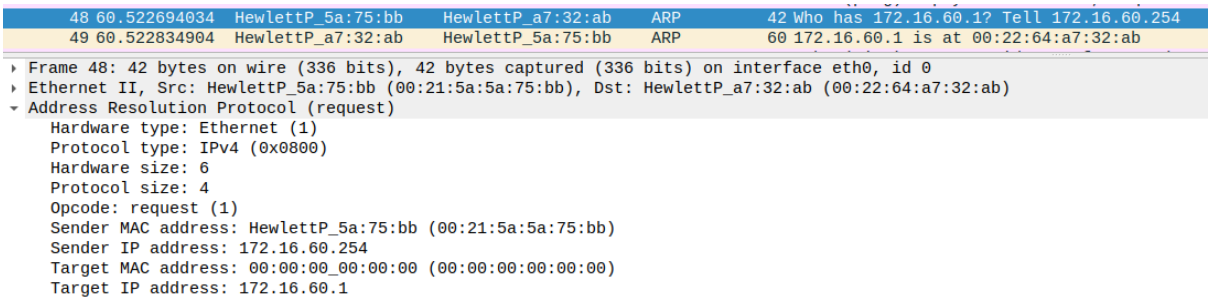



Figura 10

```
    48 60.522694034  HewlettP_5a:75:bb    HewlettP_a7:32:ab    ARP          42 Who has 172.16.60.1? Tell 172.16.60.254
    49 60.522834904  HewlettP_a7:32:ab    HewlettP_5a:75:bb    ARP          60 172.16.60.1 is at 00:22:64:a7:32:ab

▸ Frame 49: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
▸ Ethernet II, Src: HewlettP_a7:32:ab (00:22:64:a7:32:ab), Dst: HewlettP_5a:75:bb (00:21:5a:5a:75:bb)
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: HewlettP_a7:32:ab (00:22:64:a7:32:ab)
    Sender IP address: 172.16.60.1
    Target MAC address: HewlettP_5a:75:bb (00:21:5a:5a:75:bb)
    Target IP address: 172.16.60.254
```

Figura 11

```
    78 21.825471095  172.16.40.1      172.16.2.1       DNS         74 Standard query 0x9eb9 A www.github.com
    79 21.825483178  172.16.40.1      172.16.2.1       DNS         74 Standard query 0xd5c6 AAAA www.github.com
    80 21.827564239  172.16.2.1       172.16.40.1      DNS        175 Standard query response 0xd5c6 AAAA www.github.com CNAME github.com SOA ns-1707.awsdns-21.co.uk
    81 21.827798076  172.16.2.1       172.16.40.1      DNS        104 Standard query response 0x9eb9 A www.github.com CNAME github.com A 140.82.121.4
    82 21.827996711  172.16.40.1      140.82.121.4     ICMP        98 Echo (ping) request  id=0x78db, seq=1/256, ttl=64 (reply in 83)
    83 21.872494588  140.82.121.4     172.16.40.1      ICMP        98 Echo (ping) reply    id=0x78db, seq=1/256, ttl=46 (request in 82)
    84 21.872721370  172.16.40.1      172.16.2.1       DNS         85 Standard query 0x82da PTR 4.121.82.140.in-addr.arpa
    85 21.873493352  172.16.2.1       172.16.40.1      DNS        129 Standard query response 0x82da PTR 4.121.82.140.in-addr.arpa PTR lb-140-82-121-4-fra.github.com
```

Figura 13

```
john@ubuntu-machine:~/Documents/RCOMsecond/code$ ./download ftp://anonymous:1@ftp.up.pt/pub/scientific/documents/graphics/latest/LICENSE.TXT
Parsing URL...

Printing Commands...
user_command: user anonymous

pass_command: pass 1

file_command: retr pub/scientific/documents/graphics/latest/LICENSE.TXT

passive_command: pasv

Host name  : mirrors.up.pt
IP Address : 193.137.29.15

Connecting to server...
Response: 220

Sending user command
Response: 331 Please specify the password.

Sending pass command
Response: 230 Login successful.

Sending passive command
Response: 227 Entering Passive Mode (193,137,29,15,197,243).

Sending retr command
Response: 150 Opening BINARY mode data connection for pub/scientific/documents/graphics/latest/LICENSE.TXT (20133 bytes).

john@ubuntu-machine:~/Documents/RCOMsecond/code$ cat LICENSE.TXT
Attribution-ShareAlike 4.0 International

=======================================================================

Creative Commons Corporation ("Creative Commons") is not a law firm and
does not provide legal services or legal advice. Distribution of
Creative Commons public licenses does not create a lawyer-client or
other relationship. Creative Commons makes its licenses and related
information available on an "as-is" basis. Creative Commons gives no
warranties regarding its licenses, any material licensed under their
terms and conditions, or any related information. Creative Commons
disclaims all liability for damages resulting from their use to the
```

Figura 14