2º TRABALHO LABORATORIAL

Computer Networks

Mestrado em Engenharia Eletrotécnica e de Computadores
Redes de Computadores 2021/2022

Gonçalo Santos up202003537 1MEEC_T02
Guilherme Moreira up201806631 1MEEC_T02

26 de janeiro de 2022

Índice

Introdução

Este trabalho foi desenvolvido no âmbito da unidade curricular de Redes de Computadores visando implementar uma aplicação que recorresse ao FTP (*File Transfer Protocol*), tal como a realização de diversas experiências em ambiente laboratorial consistindo na configuração e estudo de uma rede de computadores.

Neste relatório iremos analisar em partes distintas as funcionalidades implementadas da aplicação, apelidada *download.c*, tal como os passos e resultados das experiências de modo a explicar como as realizamos.

1. Parte 1 – Aplicação FTP

A aplicação FTP faz a transferência de um ficheiro, adota o protocolo FTP como descrito no RFC959 e recebe como argumento o endereço do ficheiro com a sintaxe *ftp://[<user>:<password>@]<host>/<url-path>* como descrito no RFC1738.

Inicialmente, é feita a análise do endereço passado na linha de comandos pelo utilizador invocando a função *url_parsing()* de modo a retirar o *host*, o diretório do ficheiro e o nome do ficheiro. Caso, sejam indicados um utilizador e uma palavra-passe na linha de comandos, estes também são retirados. Também é convertido o *host* através da função *gethostbyname()* obtendo assim uma struct do tipo *hostent*, que contêm dados do *host*, nomeadamente, o endereço IPv4 e o nome do servidor FTP.

Deste modo, é possível criar a ligação TCP *(Transmission Control Protocol)* abrindo um *socket* na porta 21. Para tal, é usada a função *open_socket()*, que abre um *socket*, na porta especificada em parâmetro e faz a conexão ao servidor. É ainda usada a função *fdopen()* para receber a *stream* de *bytes* pelo *socket*.

Na verdade, este *socket* é usado para enviar e receber comandos, para tal usamos as funções *write_to_socket() e read_from_socket()*, respetivamente. Caso não seja especificado um utilizador e uma palavra-passe é feito o *login* em modo anónimo, ou seja, enviando os comandos *user anonymous* e *pass anonymous*. Caso seja especificado, ambos os comandos anteriores são enviados, mas com o respetivo nome de utilizador e palavra-passe. Se o servidor só aceitar o *login* em modo anónimo, a ligação é cancelada se este não for o *login* utilizado.

Após o *login* ser efetuado, é iniciada a ligação em modo passivo com recurso ao comando *pasv* gerando uma resposta do servidor. Essa resposta é tratada pela função *check_port()* com fim a retirar os dois últimos valores que vão ser necessários para calcular a porta essencial na criação de um segundo *socket* que irá tratar da transferência do ficheiro. Esta nova porta é calculada na função *calc_port()* através da equação $x_5 * 256 + x_6$.

Este segundo *socket* é aberto nesta nova porta, de modo semelhante ao anterior. Paralelamente é enviado o comando *retr* com o diretório do ficheiro no primeiro *socket*, tal como, em caso de sucesso, é criado o ficheiro no diretório da aplicação. O ficheiro criado é preenchido com o que receber no segundo *socket* e à medida que é preenchido uma barra de progresso é atualizada. Esta barra de progresso compara o tamanho de *bytes* recebidos com o tamanho total do ficheiro que é previamente indicado na resposta do servidor ao comando *retr*.

Por último, são fechadas as *streams* de *bytes* dos *sockets* e consequentemente os *sockets* através da função *fclose()*.

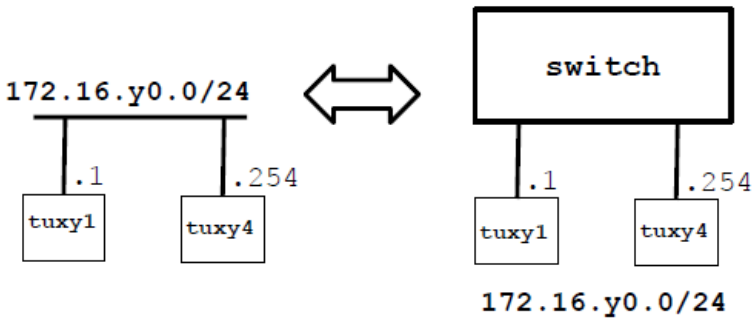
A aplicação foi testada tanto em ambiente laboratorial como em máquinas virtuais, sendo visíveis no anexo 1 o resultado no terminal de comandos de um utilizador e no anexo 2 os comandos FTP enviados ao servidor e alguns pacotes de dados da transmissão do ficheiro.

2. Parte 2 – Configuração e Estudo de uma Rede

2.1.Exp 1 – *Configure an IP Network*

2.1.1. Arquitetura da rede

Primeiramente, foi desconectado o switch da porta 5.1 (*netlab*) e conectados os cabos conforme o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*) e *tux* 53 (*tuxy1*) da bancada 5.



A seguir, foi necessário configurar o *eth0* do *tux* 53 com o IP 172.16.50.1/24 (máscara de 255.255.255.0) e o *eth0* do *tux* 54 com o IP 172.16.50.254/24, recorrendo aos comandos presentes no anexo 3.

2.1.2. Análise e discussão dos resultados

- ***What are the ARP packets and what are they used for?***

  ARP (*Address Resolution Protocol*) é um protocolo da camada 2 utilizado para converter endereços IP em endereços físicos MAC (*Media Access Control*).

  O ARP transmite um pacote para todos os *hosts* conectados à *Ethernet*. Este pacote contém o endereço IP com que o transmissor está interessado em comunicar.

- ***What are the MAC and IP addresses of ARP packets and why?***

  Na verdade, os pacotes ARP contêm os endereços MAC e IP tanto do transmissor tanto do recetor.

  Quando o transmissor quer iniciar uma transmissão e só sabe o endereço IP do recetor, o pacote ARP contem o endereço MAC e IP deste, mas só o endereço IP do recetor, pois como o seu endereço MAC é desconhecido é ignorado.

- ***What packets does the ping command generate?***

  O comando *ping* gera pacotes ICMP *(Internet Control Message Protocol)*. O pacote enviado do transmissor é chamado *ICMP_echo_request* e o pacote enviado do recetor *ICMP_echo_reply*.

  Além disso, este comando identifica o alcance de um *host*, indicando a existência de erros na rede estabelecida, perda de pacotes e ainda estatísticas dos resultados.

- ***What are the MAC and IP addresses of the ping packets?***

  Os pacotes ICMP, enviados aquando de um comando *ping*, contêm os endereços MAC e IP do transmissor e do recetor.

- ***How to determine if a receiving Ethernet frame is ARP, IP, ICMP?***

  A trama *Ethernet* contem no seu *header* um campo que identifica o tipo de protocolo dentro da trama (*EtherType*). Este campo tem 2 *bytes* (16 bits). No caso do protocolo IP o valor deste campo é 0x0800, e caso o campo de protocolo da trama IP tiver o valor 1 o protocolo internet é do tipo ICMP. No caso do protocolo ARP o valor é 0x0806.

- ***How to determine the length of a receiving frame?***

  O tamanho da trama recebida está indicado no campo *length* do header da trama *Ethernet*.

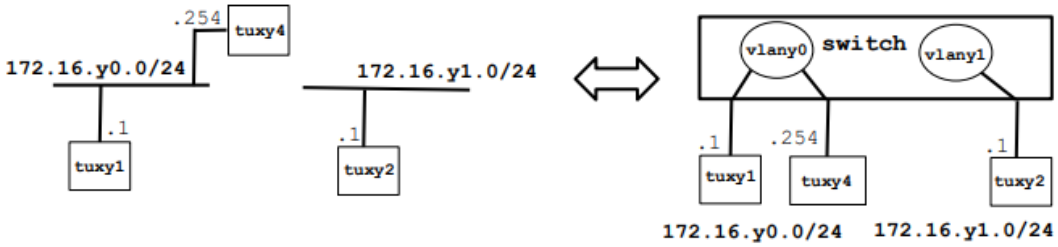
- ***What is the loopback interface and why is it important?***

  A *interface loopback* é responsável por enviar pacotes *LOOP*, de volta ao transmissor para testar a ligação. Cada pacote *LOOP* tem um espaçamento de 10s.

2.2.Exp 2 – *Implement two virtual LANs in a switch*

2.2.1. Arquitetura da rede

Primeiramente, foram conectados os cabos consoante o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 1.



Do mesmo modo que demonstrado na experiência anterior, foi configurado o *eth0* do *tux* 52 com o IP 172.16.51.1/24. Depois, foi usada a porta 2 para a conectar o *tux* 52 à *vlan* 51, recorrendo aos comandos no anexo 4 no GTKTerm.

De igual modo, foram configuradas a porta 13 para conectar o *tux* 53 à *vlan* 50 e a porta 14 para conectar o *tux* 54 à *vlan* 50.

2.2.2. Análise e discussão dos resultados

- ***How to configure vlany0?***

  A ideia por detrás da configuração das *vlan* 50 já foi referida na parte da arquitetura da rede desta experiência, estando os comandos da configuração identificados no anexo 4.

- ***How many broadcast domains are there? How can you conclude it from the logs?***

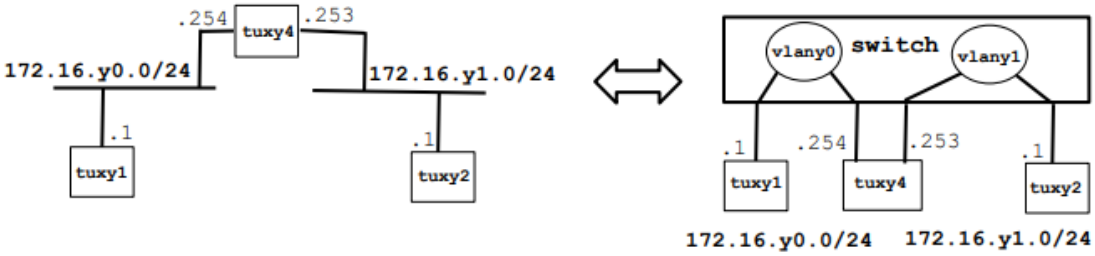
  Ao fazer um *ping* em *broadcast* do *tux* 53, recebemos resposta do *tux* 54, mas não do *tux* 52, pelo que existem dois domínios *broadcast*. Infelizmente não foi possível guardar os *logs* dos *pings* em modo *broadcast* porque a *pen* USB não era reconhecida.

2.3.Exp 3 – *Configure a Router in Linux*

Devido a um imprevisto no processo de configuração da bancada 5 foi optado por realizar esta experiência na bancada 6 pelo que nos anexos a nomenclatura considerada é a desta. No entanto, a nomenclatura usada a seguir considerará a bancada 5.

2.3.1. Arquitetura da rede

Primeiramente, foram conectados os cabos segundo o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 2.



Do mesmo modo que demonstrado na experiência anterior, foi configurado o *eth1* do *tux* 54 com o IP 172.16.51.253/24.

Tal como na experiência anterior, foi criada a *vlan* 50, tendo associada a porta 13 (*tux* 53) e a porta 14 (*eth0* do *tux* 54) e a *vlan* 51, tendo associada a porta 2 (*tux* 52) e a porta 4 (*eth1* do *tux* 54).

Além disso, foram configuradas as *gateways* do *tux* 53 e do *tux* 52 com recurso aos comandos no anexo 5.

Para terminar também foi ativado o IP *forwarding* e desativado o *echo-ignore-broadcast* do ICMP no *tux* 54 (anexo 6), tendo em atenção que estas características foram definidas na configuração inicial das experiências posterior a esta.

2.3.2. Análise e discussão dos resultados

- ***What routes are there in the tuxes? What are their meaning?***

  O *tux* 53 contem a rota para a *vlan* 51 através da *gateway* da *vlan* 50 (.254) e o *tux* 52 contem a rota para a *vlan* 50 pela *gateway* da *vlan* 51 (.253). Estas rotas permitem a comunicação entre *hosts* nas diferentes *vlans*. A tabela de roteamento do *tux* 52 encontra-se no anexo 7.

- ***What information does an entry of the forwarding table contain?***

  Uma tabela de encaminhamento contém o endereço de destino, a *gateway*, *a* rota da qual o destino pode ser alcançado e a *interface* de rede utilizada pela rede.

- ***What ARP messages, and associated MAC addresses, are observed and why?***

  Como ambas as *gateways* estão no *tux* 54 são verificadas mensagens ARP entre o *tux* 53 e o *tux* 54 e entre o *tux* 54 e o *tux* 52. Estas mensagens são do seguinte tipo: o recetor envia a mensagem ARP: *"Who has <<endereço IP>>"* e o transmissor envia a mensagem ARP: *<<endereço IP>> is at <<endereço MAC>>* e são necessárias para ser possível estabelecer a ligação entre o *tux* 53 e o *tux* 52. Uma captura das mensagens ARP entre o tux 54 e o tux 53 encontra-se no anexo 8.

- ***What ICMP packets are observed and why?***

  Os pacotes ICMP observados são os pacotes *ICMP_echo_request* e *ICMP_echo_reply* resultantes do ping entre o *tux* 53 e o *tux* 52.

- ***What are the IP and MAC addresses associated to ICMP packets and why?***

  Analisando os logs realizados no *tux* 54 (anexos 9 e 10) é possível observar que aquando de um *ping* do *tux* 53 para o *tux* 52 o endereço IP de origem é o do *tux* 53 e o de destino é o do *tux* 52 tanto no *eth0* (*vlan* 50) como no *eth1* (*vlan* 51).

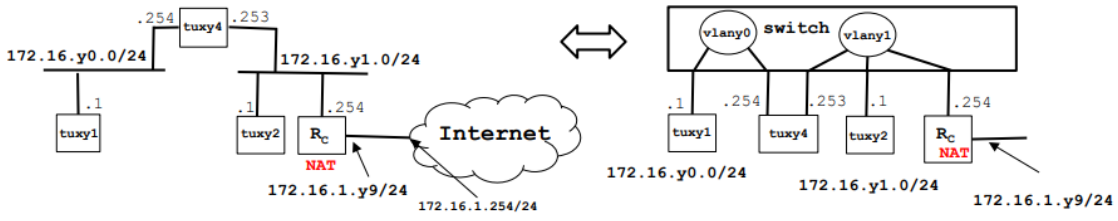
  No entanto, no *eth0* o endereço MAC de origem é o do *tux* 53 e o de destino o do *eth0* do *tux* 54 a *gateway* da vlan50, enquanto no *eth1* o endereço de origem é o do *eth1* do *tux* 54 a *gateway* da vlan51 e o de destino o do *tux* 52.

  Na verdade, como previamente é feita uma troca de mensagens ARP entre o tux 53 e o *tux* 54 e entre o *tux* 52 e o *tux* 54, os endereços MAC do *tux* 54 são usados para redirecionar os pacotes, pois nem o tux 53 nem o tux 52 sabem o endereço MAC um do outro.

2.4.Exp 4 – *Configure a Commercial Router and Implement NAT*

2.4.1. Arquitetura da rede

Primeiramente, foram conectados os cabos conforme o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 3.



A seguir, foi necessário configurar o *router* da CISCO (Rc), recorrendo aos comandos do anexo 11 no GTKTerm para definir a *interface inside* com o IP 172.16.51.254/24 e a *interface outside* com o IP 172.16.1.59/24. Também são definidas as rotas estáticas dos endereços de origem e de destino. É importante realçar que, inicialmente, não foi ativada a funcionalidade NAT (*Network Address Translation*) e os comandos relativos a esta componente foram só usados após o 6º passo do guião.

Tal como na experiência anterior, foi criada a *vlan* 50, tendo associada a porta 13 (tux 53) e a porta 14 (*eth0* do *tux* 54) e a *vlan* 51, tendo associada a porta 2 (*tux* 52), a porta 4 (*eth1* do *tux* 54) e a porta 7 (*eth0* do Rc).

Além disso, foram configuradas as *gateways* do *tux* 53, do *tux* 52 e do *tux* 54 com recurso aos comandos no anexo 12.

2.4.2. Análise e discussão dos resultados

- ***How to configure a static route in a commercial router?***

  A ideia por detrás da configuração de rotas estáticas no *router* já foi referida na parte da arquitetura da rede desta experiência, estando os comandos da configuração identificados no anexo 11.

- ***What are the paths followed by the packets in the experiments carried out and why?***

  Inicialmente, com a rota 172.16.50.0/24 via *tux* 54 definida e com o redirecionamento ICMP ativo é possível realizar o *ping* entre os *tux* 52 e o *tux* 53. No entanto, sem a rota definida e com o redirecionamento ICMP ativo já é possível, sendo que em caso deste não estar ativo já não é.

  Além disso, realizando o *traceroute* é possível verificar que quando a rota está definida, o pacote inicialmente vai para o *tux* 54 e, a seguir, para o *tux* 53. Enquanto, quando a rota não está definida e o redirecionamento ICMP está ativo, o pacote vai inicialmente para o *router* e só depois para o *tux* 54.

- ***How to configure NAT in a commercial router?***

  A ideia por detrás da configuração do *router* com a funcionalidade de NAT já foi referida na parte da arquitetura da rede desta experiência, estando os comandos da configuração identificados no anexo 11.

- ***What does NAT do?***

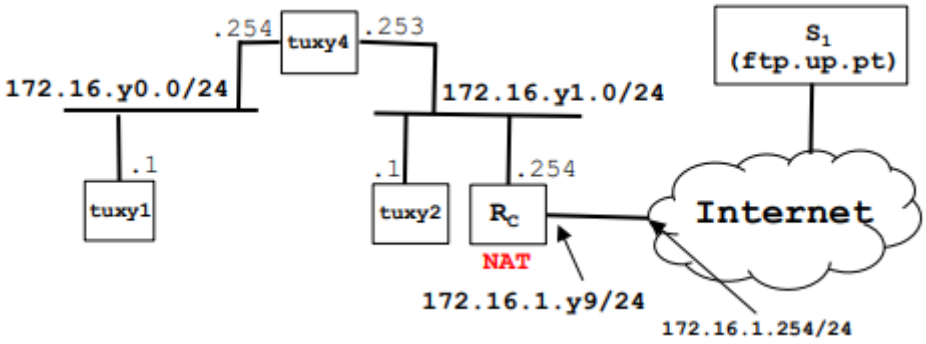
  O NAT *(Network Adress Translation)* é uma técnica utilizada para converter endereços IP privados, internos a uma rede, em endereços IP públicos, permitindo que estes consigam comunicar com outras redes.

  Este comportamento foi verificado na última fase desta experiência, visto que antes de adicionar a NAT ao *router* (Rc) não era possível comunicar com o router da sala e, após esta ser adicionada, já era possível.

2.5. Exp 5 – DNS

2.5.1. Arquitetura da rede

Primeiramente, foram conectados os cabos consoante com o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 4.



A seguir, foi verificado que o *tux* 54, o *tux* 53 e o *tux* 52 já tinham o DNS configurado, recorrendo ao comando no anexo 13.

2.5.2. Análise e discussão dos resultados

- ***How to configure the DNS service at a host?***

  Para configurar o servidor DNS no *host* é necessário aceder ao ficheiro *etc/resolv.conf* e adicionar o nome do servidor e o endereço IP deste.

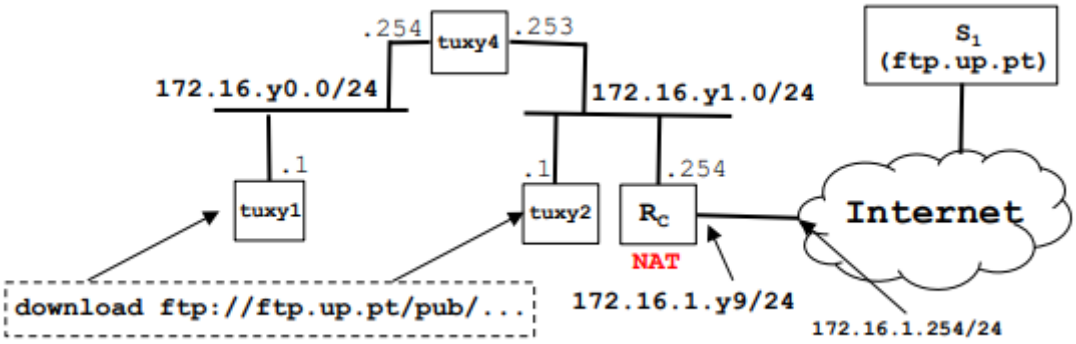
- ***What packets are exchanged by DNS and what information is transported***

  Ao realizar um *ping* do *tux* 52 para a *google.com*, o *tux* 52 envia dois pacotes para o servidor DNS da FEUP (172.16.1.1) a "perguntar" quais os endereços IP (IPv4 e IPv6) do domínio *google.com*. De seguida, recebe do servidor DNS da FEUP dois pacotes de resposta com os endereços IP. No anexo 14 conseguimos observar os pacotes DNS.

2.6.Exp 6 – TCP *connections*

2.6.1. Arquitetura da rede

Primeiramente, foram conectados os cabos segundo o seguinte esquema, tendo sido usados o *tux* 54 (*tuxy4*), *tux* 53 (*tuxy1*) e *tux* 52 (*tuxy2*) da bancada 5 com todas as configurações efetuadas na experiência 5.



2.6.2. Análise e discussão dos resultados

- ***How many TCP connections are opened by your ftp application?***

  São abertas *2* conexões, uma na porta 21 para a transferência de comandos e respostas FTP e uma na porta 59220, definida pela aplicação após entrar em modo passivo, para a transferências de dados. No anexo 15 conseguimos observar essas conexões e as portas associadas.

- ***In what connection is transported the FTP control information?***

  A informação de controlo é transportada na conexão de transferência de comandos e respostas FTP na porta 21.

- ***What are the phases of a TCP connection?***

  As fases de uma conexão TCP são 3. A fase de estabelecimento da conexão, a fase de troca de dados e a fase de encerramento da conexão. A fase de estabelecimento e de encerramento encontram-se nos anexos 15 e 16.

- ***How does the ARQ TCP mechanism work? What are the relevant TCP fields? What relevant information can be observed in the logs?***

  O mecanismo ARQ *(Automatic Repeat Request),* nomeadamente, o método *Go Back N,* é utilizado pelo protocolo TCP para controlo de erros na transmissão de dados. Este controlo de erros é efetuado por pacotes ACK *(acknowledgment),* que, através do número de sequência, que indica o número do pacote, o número ACK, que indica se o pacote foi corretamente recebido e o *window size*, que indica a quantidade de informação recebida que pode ser armazenada, controla a transferência de dados. Estes campos encontram-se demonstrados no anexo 17.

- ***How does the TCP congestion control mechanism work? What are the relevant fields? How did the throughput of the data connection evolve along the time? Is it according to the TCP congestion control mechanism?***

  O TCP usa uma janela de congestionamento que controla a quantidade de bytes enviados na rede para que esta não fique sobrecarregada e neste caso diminui a velocidade de transmissão.

- ***Is the throughput of a TCP data connections disturbed by the appearance of a second TCP connection? How?***

  O *throughput* de uma conexão TCP irá sofrer uma redução caso surja uma segunda conexão TCP, pois a largura de banda disponível também é diminuída.

Conclusão

Assim sendo, a aplicação FTP implementada cumpriu os objetivos principais deste trabalho, tendo passado com sucesso os testes efetuados na experiência 6, tal como, explorados conhecimentos de configuração e estudo de uma rede de computadores em ambiente laboratorial.

Anexos

Anexo 1 – Aplicação FTP

```
┌──(kali㉿kali)-[~/Desktop/RCOM/Lab 2/Final]
└─$ ./download ftp://ftp.up.pt/pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt
Host: ftp.up.pt
Path: pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt
File: changelog.txt
Host name  : ftp.up.pt
IP Address : 193.137.29.15

220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
220-                              
220-
220-All connections and transfers are logged. The max number of connections is 200.
220-
220-For more information please visit our website: http://mirrors.up.pt/
220-Questions and comments can be sent to mirrors@uporto.pt
220-
220-
220
Starting Default Logging Procedure
230 Login successful.
227 Entering Passive Mode (193,137,29,15,196,17)
150 Opening BINARY mode data connection for ./pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt (3175 bytes).
100% [OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO]
226 Transfer complete.
```

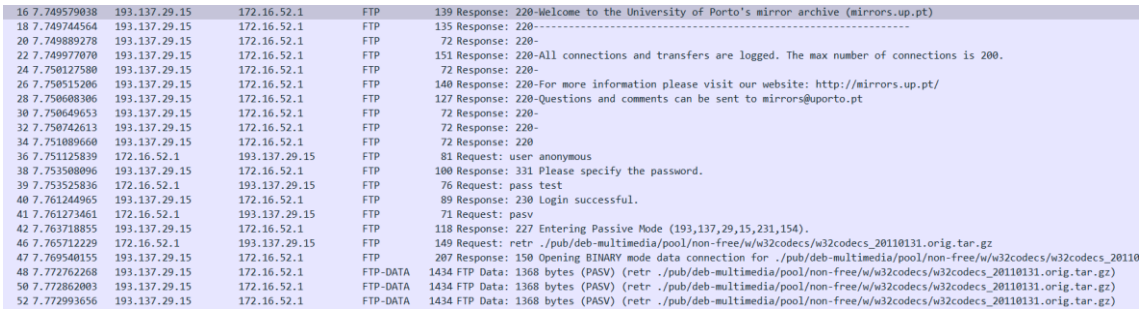
```
┌──(kali㉿kali)-[~/Desktop/RCOM/Lab 2/Final]
└─$ ./download ftp://user:pass@ftp.up.pt/pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt
User: user
Pass: pass
Host: ftp.up.pt
Path: pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt
File: changelog.txt
Host name  : ftp.up.pt
IP Address : 193.137.29.15

220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
220-                              
220-
220-All connections and transfers are logged. The max number of connections is 200.
220-
220-For more information please visit our website: http://mirrors.up.pt/
220-Questions and comments can be sent to mirrors@uporto.pt
220-
220-
220
530 This FTP server is anonymous only.
```

```
┌──(kali㉿kali)-[~/Desktop/RCOM/Lab 2/Final]
└─$ ./download ftp://anonymous:anonymous@ftp.up.pt/pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt
User: anonymous
Pass: anonymous
Host: ftp.up.pt
Path: pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt
File: changelog.txt
Host name  : ftp.up.pt
IP Address : 193.137.29.15

220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
220-                              
220-
220-All connections and transfers are logged. The max number of connections is 200.
220-
220-For more information please visit our website: http://mirrors.up.pt/
220-Questions and comments can be sent to mirrors@uporto.pt
220-
220-
220
230 Login successful.
227 Entering Passive Mode (193,137,29,15,200,250)
150 Opening BINARY mode data connection for ./pub/ubuntu-feup-legacy/2010/ubuntu-feup/changelog.txt (3175 bytes).
100% [OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO]
226 Transfer complete.
```

Anexo 2 – Captura Wireshark da aplicação FTP



16	7.749579038	193.137.29.15	172.16.52.1	FTP	139 Response: 220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
18	7.749744564	193.137.29.15	172.16.52.1	FTP	135 Response: 220----------
20	7.749889278	193.137.29.15	172.16.52.1	FTP	72 Response: 220-
22	7.749977070	193.137.29.15	172.16.52.1	FTP	151 Response: 220-All connections and transfers are logged. The max number of connections is 200.
24	7.750127580	193.137.29.15	172.16.52.1	FTP	72 Response: 220-
26	7.750515206	193.137.29.15	172.16.52.1	FTP	140 Response: 220-For more information please visit our website: http://mirrors.up.pt/
28	7.750608306	193.137.29.15	172.16.52.1	FTP	127 Response: 220-Questions and comments can be sent to mirrors@uporto.pt
30	7.750649653	193.137.29.15	172.16.52.1	FTP	72 Response: 220-
32	7.750742613	193.137.29.15	172.16.52.1	FTP	72 Response: 220-
34	7.751089660	193.137.29.15	172.16.52.1	FTP	72 Response: 220
36	7.751125839	172.16.52.1	193.137.29.15	FTP	81 Request: user anonymous
38	7.753508096	193.137.29.15	172.16.52.1	FTP	100 Response: 331 Please specify the password.
39	7.753525836	172.16.52.1	193.137.29.15	FTP	76 Request: pass test
40	7.761244965	193.137.29.15	172.16.52.1	FTP	89 Response: 230 Login successful.
41	7.761273461	172.16.52.1	193.137.29.15	FTP	71 Request: pasv
42	7.763718855	193.137.29.15	172.16.52.1	FTP	118 Response: 227 Entering Passive Mode (193,137,29,15,231,154).
46	7.765712229	172.16.52.1	193.137.29.15	FTP	149 Request: retr ./pub/deb-multimedia/pool/non-free/w/w32codecs/w32codecs_20110131.orig.tar.gz
47	7.769540155	193.137.29.15	172.16.52.1	FTP	207 Response: 150 Opening BINARY mode data connection for ./pub/deb-multimedia/pool/non-free/w/w32codecs/w32codecs_20110
48	7.772762268	193.137.29.15	172.16.52.1	FTP-DATA	1434 FTP Data: 1368 bytes (PASV) (retr ./pub/deb-multimedia/pool/non-free/w/w32codecs/w32codecs_20110131.orig.tar.gz)
50	7.772862003	193.137.29.15	172.16.52.1	FTP-DATA	1434 FTP Data: 1368 bytes (PASV) (retr ./pub/deb-multimedia/pool/non-free/w/w32codecs/w32codecs_20110131.orig.tar.gz)
52	7.772993656	193.137.29.15	172.16.52.1	FTP-DATA	1434 FTP Data: 1368 bytes (PASV) (retr ./pub/deb-multimedia/pool/non-free/w/w32codecs/w32codecs_20110131.orig.tar.gz)

Anexo 3 – Configuração IPs

```
ifconfig eth0 up
```
(tux 53)

```
ifconfig eth0 172.16.50.1/24
```
(tux 53)

```
ifconfig eth0
```
(tux 53)

```
ifconfig eth0 up
```
(tux 54)

```
ifconfig eth0 172.16.50.254/24
```
(tux 54)

```
ifconfig eth0
```
(tux 54)

Anexo 4 – Configuração *VLANs*

```
configure terminal

vlan 51

end

show vlan id 51

configure terminal

interface fastethernet 0/2

switchport mode access

switchport access vlan 51

end

show running-config interface fastethernet 0/2

show interfaces fastethernet 0/2 switchport
```

Anexo 5 – Configuração rotas Exp 3

```
route add -net 172.16.51.0/24 gw 172.16.50.254
```
(tux 53)

```
route add -net 172.16.50.0/24 gw 172.16.51.253
```
(tux 52)

Anexo 6 – Ativação IP *forwarding* e desativação do *echo-ignore-broadcast* do ICMP

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Anexo 7 – Exp 3 - captura no *tux* 52 - *routing table*

```
root@tux62:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.16.60.0     172.16.61.253   255.255.255.0   UG    0      0        0 eth0
172.16.61.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

Anexo 8 – Exp 3 - captura no *tux* 54 - *mensagens* ARP

```
5 0.815367024   HewlettP_c5:61:bb   HewlettP_61:2f:4e   ARP   42 Who has 172.16.60.1? Tell 172.16.60.254
6 0.815483450   HewlettP_61:2f:4e   HewlettP_c5:61:bb   ARP   60 172.16.60.1 is at 00:21:5a:61:2f:4e
```

Anexo 9 – Exp 3 - captura no *tux* 54 no *eth0* - *ping* do *tux* 53 para o *tux* 54

```
> Frame 38: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
v Ethernet II, Src: HewlettP_61:2f:4e (00:21:5a:61:2f:4e), Dst: HewlettP_c5:61:bb (00:21:5a:c5:61:bb)
    > Destination: HewlettP_c5:61:bb (00:21:5a:c5:61:bb)
    > Source: HewlettP_61:2f:4e (00:21:5a:61:2f:4e)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 172.16.60.1, Dst: 172.16.61.1
> Internet Control Message Protocol
```

Anexo 10 – Exp 3 - captura no *tux* 54 *no* eth1 - *ping* do *tux* 53 para o *tux* 54

```
> Frame 38: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth1, id 0
v Ethernet II, Src: 3Com_a1:35:69 (00:01:02:a1:35:69), Dst: HewlettP_5a:7d:9c (00:21:5a:5a:7d:9c)
    > Destination: HewlettP_5a:7d:9c (00:21:5a:5a:7d:9c)
    > Source: 3Com_a1:35:69 (00:01:02:a1:35:69)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 172.16.60.1, Dst: 172.16.61.1
> Internet Control Message Protocol
```

Anexo 11 – Configuração *router* com NAT

```
configure terminal
interface gigabitethernet 0/0
ip address 172.16.51.254 255.255.255.0
no shutdown
ip nat inside    (NAT)
exit
configure terminal
interface gigabitethernet 0/1
ip address 172.16.1.59 255.255.255.0
no shutdown
ip nat outside  (NAT)
exit
ip nat pool ovrld 172.16.1.59 172.16.1.59 prefix 24 (NAT)
ip nat inside source list 1 pool ovrld overload (NAT)
access-list 1 permit 172.16.50.0 0.0.0.7 (NAT)
access-list 1 permit 172.16.51.0 0.0.0.7 (NAT)
ip route 0.0.0.0 0.0.0.0 172.16.1.254
ip route 172.16.50.0 255.255.255.0 172.16.51.253
end
```

Anexo 12 – Configuração rotas Exp 4

```
route add default gw 172.16.50.254 (tux 53)
route add default gw 172.16.51.254 (tux 52)
route add default gw 172.16.51.254 (tux 54)
```

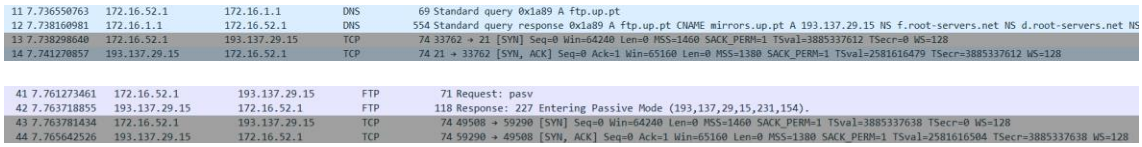
Anexo 13 – Verificação DNS

```
vi /etc/resolv.conf
```

Anexo 14 – Exp 5 - pacotes DNS

No.	Time	Source	Destination	Protocol	Length	Info
2	1.442442705	172.16.52.1	172.16.1.1	DNS	70	Standard query 0x8d08 A google.com
3	1.442453252	172.16.52.1	172.16.1.1	DNS	70	Standard query 0x9b10 AAAA google.com
4	1.445400183	172.16.1.1	172.16.52.1	DNS	334	Standard query response 0x8d08 A google.com A 216.58.209.78 NS ns3.google.com NS ns4.google.com NS ns2.google.com NS ns1.google.com A
5	1.445451797	172.16.1.1	172.16.52.1	DNS	346	Standard query response 0x9b10 AAAA google.com AAAA 2a00:1450:4003:801::200e NS ns4.google.com NS ns1.google.com NS ns3.google.com NS

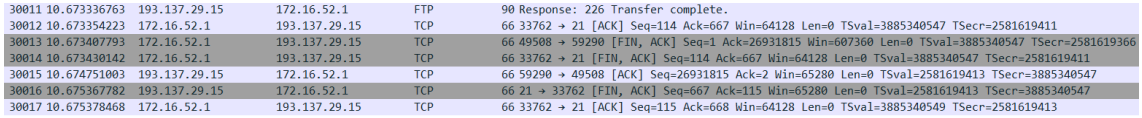
Anexo 15 – Exp 6 - Conexões TCP



11	7.736550763	172.16.52.1	172.16.1.1	DNS	69 Standard query 0x1a89 A ftp.up.pt
12	7.738160981	172.16.1.1	172.16.52.1	DNS	554 Standard query response 0x1a89 A ftp.up.pt CNAME mirrors.up.pt A 193.137.29.15 NS f.root-servers.net NS d.root-servers.net NS
13	7.738298640	172.16.52.1	193.137.29.15	TCP	74 33762 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3885337612 TSecr=0 WS=128
14	7.741270857	193.137.29.15	172.16.52.1	TCP	74 21 → 33762 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1380 SACK_PERM=1 TSval=2581616479 TSecr=3885337612 WS=128

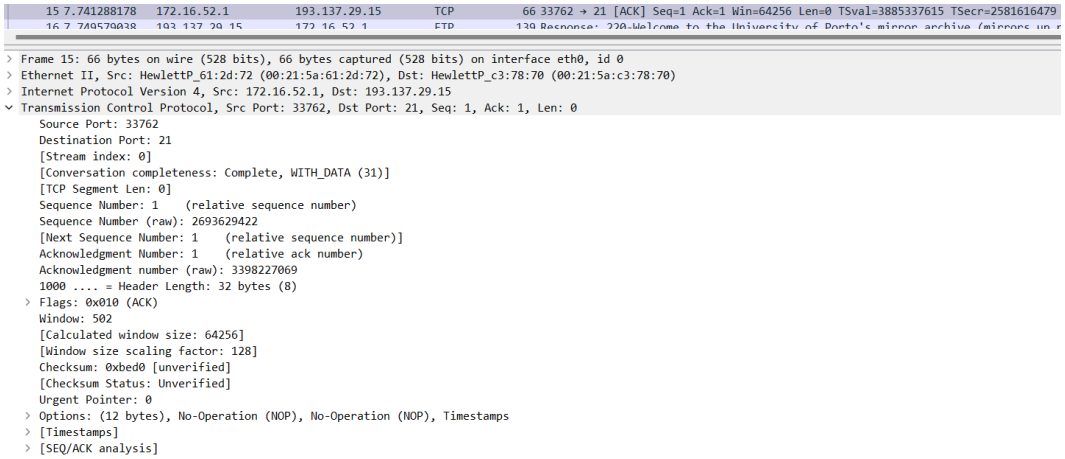
41	7.761273461	172.16.52.1	193.137.29.15	FTP	71 Request: pasv
42	7.763718855	193.137.29.15	172.16.52.1	FTP	118 Response: 227 Entering Passive Mode (193,137,29,15,231,154).
43	7.763781434	172.16.52.1	193.137.29.15	TCP	74 49508 → 59290 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3885337638 TSecr=0 WS=128
44	7.765642526	193.137.29.15	172.16.52.1	TCP	74 59290 → 49508 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1380 SACK_PERM=1 TSval=2581616504 TSecr=3885337638 WS=128

Anexo 16 – Exp 6 – Encerramento Conexão TCP



30011	10.673336763	193.137.29.15	172.16.52.1	FTP	90 Response: 226 Transfer complete.
30012	10.673354223	172.16.52.1	193.137.29.15	TCP	66 33762 → 21 [ACK] Seq=114 Ack=667 Win=64128 Len=0 TSval=3885340547 TSecr=2581619411
30013	10.673407793	172.16.52.1	193.137.29.15	TCP	66 49508 → 59290 [FIN, ACK] Seq=1 Ack=26931815 Win=607360 Len=0 TSval=3885340547 TSecr=2581619366
30014	10.673430142	172.16.52.1	193.137.29.15	TCP	66 33762 → 21 [FIN, ACK] Seq=114 Ack=667 Win=64128 Len=0 TSval=3885340547 TSecr=2581619411
30015	10.674751003	193.137.29.15	172.16.52.1	TCP	66 59290 → 49508 [ACK] Seq=26931815 Ack=2 Win=65280 Len=0 TSval=2581619413 TSecr=3885340547
30016	10.675367782	193.137.29.15	172.16.52.1	TCP	66 21 → 33762 [FIN, ACK] Seq=667 Ack=115 Win=65280 Len=0 TSval=2581619413 TSecr=3885340547
30017	10.675378468	172.16.52.1	193.137.29.15	TCP	66 33762 → 21 [ACK] Seq=115 Ack=668 Win=64128 Len=0 TSval=3885340549 TSecr=2581619413

Anexo 17 – Exp 6 – Campos TCP



15	7.741288178	172.16.52.1	193.137.29.15	TCP	66 33762 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3885337615 TSecr=2581616479
16	7.749579038	193.137.29.15	172.16.52.1	FTP	139 Response: 220-Welcome to the University of Porto's mirror archive (mirrors.up.p

```
> Frame 15: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
> Ethernet II, Src: HewlettP_61:2d:72 (00:21:5a:61:2d:72), Dst: HewlettP_c3:78:70 (00:21:5a:c3:78:70)
> Internet Protocol Version 4, Src: 172.16.52.1, Dst: 193.137.29.15
v Transmission Control Protocol, Src Port: 33762, Dst Port: 21, Seq: 1, Ack: 1, Len: 0
    Source Port: 33762
    Destination Port: 21
    [Stream index: 0]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2693629422
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3398227069
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x010 (ACK)
    Window: 502
    [Calculated window size: 64256]
    [Window size scaling factor: 128]
    Checksum: 0xbed0 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
```