Aplicação de download e configuração de rede



Relatório

Redes de Computadores

Francisco Teixeira Lopes

ei11056

Nelson André Garrido da Costa

up201403128

**Sumário**

Este relatório incide sobre a criação de uma aplicação de *download* e configuração de uma rede.

A aplicação utiliza o protocolo FTP com ligações TCP para efetuar a transferência, implementando uma versão leve do RFC959.

A rede consiste numa série de experiências que culminaram em configurar um *router* comercial com NAT, um *switch* com duas LAN virtuais, e cada computador da rede com respetivo endereço IP e DNS.

O objetivo final, foi efetuar uma transferência usando a aplicação na rede configurada.

Índice

[Introdução 1](#_Toc501707013)

[Parte 1 – Aplicação de *download* 2](#_Toc501707014)

[Arquitetura da aplicação 2](#_Toc501707015)

[Exemplo de *download* 3](#_Toc501707016)

[Parte 2 – Configuração de rede 4](#_Toc501707017)

[Experiência 1 – Configurar uma rede IP 4](#_Toc501707018)

[Experiência 2 – Implementar duas LAN virtuais num *switch* 5](#_Toc501707019)

[Experiência 3 – Configurar um *router* em Linux 5](#_Toc501707020)

[Experiência 4 – Configurar um *router* comercial e implementar NAT 6](#_Toc501707021)

[Experiência 5 – DNS 6](#_Toc501707022)

[Experiência 6 – Ligações TCP 7](#_Toc501707023)

[Conclusão 9](#_Toc501707024)

[Anexos 10](#_Toc501707025)

[Anexo 1 – Comandos de configuração 10](#_Toc501707026)

[Experiência 1 10](#_Toc501707027)

[Experiência 2 10](#_Toc501707028)

[Experiência 3 11](#_Toc501707029)

[Experiência 4 (sem NAT) 11](#_Toc501707030)

[Experiência 4 (NAT) 12](#_Toc501707031)

[Experiência 5 12](#_Toc501707032)

[Anexo 2 – Capturas 13](#_Toc501707033)

[Anexo 3 – Código da aplicação de download 14](#_Toc501707034)

# Introdução

O segundo trabalho laboratorial de Redes de Computadores divide-se em duas partes: criação de uma aplicação de *download* e configuração de uma rede.

A primeira parte, tem por objetivo efetuar a transferência de um ficheiro a partir de um servidor, para isto, é utilizado o protocolo FTP com implementação do RFC959 e conexões TCP ao servidor.

A segunda parte, tem por objetivo configurar uma rede de forma a poder efetuar a transferência de um ficheiro. Foram efetuadas várias experiências incrementais, que eventualmente, permitiram acesso à Internet a partir da rede configurada.

Este relatório encontra-se dividido em duas seções principais:

* Aplicação de *download*: nesta seção descreve-se a arquitetura da aplicação e apresenta-se um relato de uma transferência feita com sucesso.
* Configuração de rede: nesta seção apresentam-se os passos para configuração da rede e reflete-se sobre os objetivos de aprendizagem de cada experiência.

# Parte 1 – Aplicação de *download*

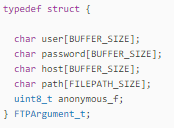
## Arquitetura da aplicação

O processamento da aplicação acontece na função main do programa, esta chama, sequencialmente, todas as funções auxiliares necessárias para descodificar o argumento passado pelo utilizador, a obtenção dos dados necessários para poder fazer a ligação TCP ao servidor FTP, e ainda, as operações necessárias para conseguir fazer o *download* do ficheiro pedido.

Para este efeito são usadas duas *structs* com dados utéis ao programa, a primeira *struct*, FTPFile\_t, contém o caminho para onde guardar o ficheiro localmente, e também, o descritor de ficheiro do *socket* de dados.

A segunda *struct*, FTPArgument\_t, contém os comandos FTP necessários para obter o ficheiro, bem como, um valor booleano que determina se o modo é *anonymous*.

Figura 1 - estruturas de dados da aplicação



A aplicação é executada recorrendo a um argumento, que indica o nome de utilizador e a palavra-passe, juntamente com, o URL do servidor e ficheiro a transferir. Pode-se também omitir o utilizador e palavra-passe, sendo que, é assumido o modo *anonymous*.

Figura 2 - utilização da aplicação

Após executar a aplicação, é chamada uma sequência de funções com a finalidade de transferir o ficheiro pedido. Primeiramente, é validado o argumento fornecido, para esse efeito, a função “*parseArgument*” verifica que todos os dados necessários estão presentes. A *struct* FTPArgument\_t é preenchida por esta função, no caso de argumento válido. De seguida, é obtido o endereço IP do servidor através da função “*getAddress*”, o qual é fornecido à função “*getTCPSocket*”. Esta, cria uma ligação TCP à porta 21 do servidor, isto é, a porta de controlo de FTP. Após estabelecer ligação ao servidor, são enviados os comandos necessários para transferir o ficheiro, a função “FTPLogin” envia os comandos “USER” e “PASS”, de seguida “FTPCommand” envia “TYPE I” para usar modo binário e finalmente “FTPPassive” envia o comando “PASV” e retorna a porta de dados. Sabendo a porta de dados, “getTCPSocket” cria outra ligação TCP com o mesmo endereço IP mas porta distinta. Criada esta ligação, “FTPCommand” envia “RETR” e inicia a transferência do ficheiro, isto é feito através de um valor booleano na função, que indica a possibilidade do comando ativar uma transferência. Termina, assim, a aplicação com o ficheiro transferido na pasta onde foi executada, qualquer erro é devidamente mostrado ao utilizador e, tratando-se da seção FTP, é tratado de acordo com o RFC959.

## Exemplo de *download*

Para fazer transferência de um ficheiro, basta fornecer à aplicação um URL correto e, opcionalmente, o utilizador e palavra-passe. Seguem-se imagens de testes bem sucedidos, um em modo anónimo, e o outro com dados de utilizador.

Figura 3 - download com utilizador

Os comandos enviados foram:

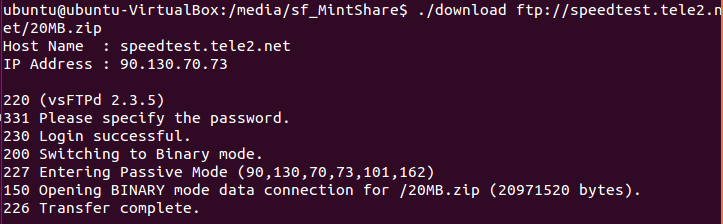
* ./download ftp://demo:password@test.rebex.net/pub/example/mail-editor.png
* ./download ftp://speedtest.tele2.net/20MB.zip

Figura 4 - download em modo anónimo

# Parte 2 – Configuração de rede

Nos seguintes subcapítulos são elucidados os objetivos de cada experiência, sendo que, a configuração exaustiva é apresentada em anexo no final do relatório, assim como, os respetivos *logs* de captura de pacotes.

## Experiência 1 – Configurar uma rede IP

Esta experiência teve por objetivo a configuração de uma rede local, na qual, se ligava um computador a outro. Para este efeito, ligaram-se os computadores diretamente através da porta eth0, e de seguida, configuraram-se os seus endereços IP de forma a situarem-se na mesma sub-rede (172.16.10.x).

**O que são pacotes ARP e para que servem?**

Os pacotes ARP servem para mapear um endereço de rede (IPv4) a um endereço físico (MAC).

**Qual é o MAC e endereço IP de pacotes ARP?**

Existem pacotes ARP de pedido e de resposta, num pacote de pedido, é enviado o endereço IP do computador que fez o pedido, e também, o endereço IP do computador do qual se quer saber o MAC. Num pacote de resposta, é enviado ao computador que fez o pedido, o endereço IP e MAC do computador destino.

Figura 5 - exemplo de pacotes ARP

**Que pacotes gera o comando *ping*?**

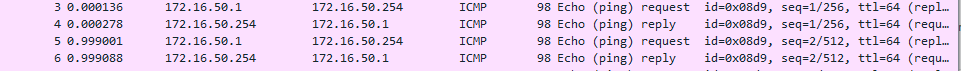
O comando *ping* gera pacotes ICMP, estes podem ser do tipo *echo request* ou do tipo *echo reply*.

Figura 6 - exemplo de pacotes ICMP

**Qual é o MAC e endereço IP de pacotes ping?**

Os pacotes ICMP contêm o endereço IP do computador origem e do computador destino, isto é válido tanto para *echo request*, como para *echo reply*.

**Como se determina se uma trama Ethernet é do tipo ARP, IP, ICMP?**

No cabeçalho da trama Ethernet, no campo *EtherType*, é possível verificar se se trata de um pacote ARP (0x0806) ou IP (0x0800). Para pacotes ICMP (tipo 1), esta informação, vem no cabeçalho do pacote IP no campo *Protocol*.

**Como se determina o tamanho de uma trama Ethernet?**

O tamanho da trama é indicado no campo *EtherType* da trama, para evitar ambiguidade com o outro uso deste campo, utilizam-se valores até 1500 para referir o tamanho, e valores acima de 1536 para o tipo.

**O que é a interface de *loopback* e para que serve?**

A interface de *loopback* é utilizada para um computador poder comunicar consigo mesmo, isto tem a utilidade de poder realizar testes de diagnóstico, e também, de poder aceder a servidores no próprio computador.

## Experiência 2 – Implementar duas LAN virtuais num *switch*

Esta experiência teve por objetivo a implementação de LAN virtual, dois computadores seriam ligados a uma VLAN e um terceiro a outra VLAN sem rota entre as duas VLAN.

Para este efeito, configurou-se um *switch* da Cisco e respetivas portas para ligar o computador 1 e 4 numa VLAN, e o computador 2 noutra VLAN.

**Quantos domínios de *broadcast* existem? Como se pode concluir isso a partir dos *logs*?**

Existem dois domínios de *broadcast* na rede configurada, pois a divisão em VLAN cria duas sub-redes. Nos *logs* pode-se ver isto pois ao fazer *ping* *broadcast* a partir do computador 1, chega-se apenas ao computador 4, e ao fazer ping broadcast do computador 2, não se chega a outro computador. Isto deve-se às redes criadas préviamente.

Figura 7 - computador 1 fazendo broadcast, resposta de computador 4

## Experiência 3 – Configurar um *router* em Linux

Esta experiência teve por objetivo utilizar o computador 4 como *router*, de forma a ligar as duas sub-redes criadas na experiência anterior.

Para este efeito, configurou-se a porta eth1 do computador 4 com um endereço IP da sub-rede do computador 2, e também, alteraram-se algumas configurações de forma a ativar *IP Forwarding* e desativar o *ICMP\_echo\_ignore\_broadcast*. Definiram-se também as rotas necessárias para cada sub-rede ter acesso à outra, usando o computador 4 como *gateway*.

**Que rotas existem nos computadores? Qual o seu significado?**

O computador 1 tem uma rota para a sub-rede do computador 2 e vice-versa, por exemplo, “172.16.11.1 172.16.10.254 255.255.255.0”. Esta rota permite ao computador 1 saber que para o destino 172.16.11.1 (computador 2), deve usar o endereço 172.16.10.254 como *gateway*, neste caso é o computador 4 (*router*).

**Que informação contém uma entrada na tabela de encaminhamento?**

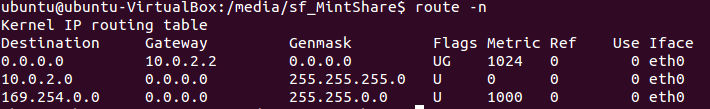
As tabelas de encaminhamento contém a informação necessária para fazer chegar pacotes ao endereço IP destino. Para isto contêm informação de destino e para onde deve ser encaminhado o pacote para chegar ao destino.

Figura 8 - exemplo de tabela de encaminhamento

**Qual o comportamento dos pacotes ARP e ICMP nesta rede?**

Estando no computador 1 e fazendo ping ao computador 2, verifica-se que o computador 1 envia para o computador 4 um pacote ICMP com o endereço IP do computador 2 no campo de destino. O computador 4, caso não tenha em cache o MAC do computador 2, envia um pacote ARP para descobrir o MAC associado ao endereço IP do ping recebido, após descobrir este MAC envia o ping original ao computador 2. O processo de resposta é idêntico com os valores invertidos, o computador 4 serve então de *router*, como é demonstrado por este comportamento.

## Experiência 4 – Configurar um *router* comercial e implementar NAT

Esta experiência teve por objetivo adicionar à rede um *router* comercial, este, configurado com NAT para permitir acesso à Internet.

Para este efeito, adicionou-se à VLAN do computador 2/4, o *router* comercial, de seguida, configuraram-se as rotas necessárias para ser possível receber pacotes de qualquer combinação de computadores. Finalmente, configurou-se NAT no *router*, para permitir acesso a redes externas.

**Qual o caminho seguido pelos pacotes na experiência?**

Inicialmente, o computador 2 tem apenas uma rota através do *router* comercial, fazendo *ping* a um endereço da outra sub-rede tem por resultado o pacote parar primeiro no *router* comercial, de seguida pára no computador 4 e daí é feita a ligação à outra sub-rede. Adicionando a rota à outra sub-rede, usando o computador 4 como *gateway*, faz com que o caminho anterior se torne mais curto, pois não tem de passar no *router* comercial primeiro.

**O que faz o NAT?**

NAT permite mapear um espaço de endereçamento IP a outro espaço, modificando o cabeçalho de datagramas, enquanto estes transitam num dispositivo de *routing*. O seu uso atual consiste em possibilitar que uma rede interna seja endereçada apenas por um único endereço IP público, sendo que esta utilização é denomeada de *IP masquerading*.

## Experiência 5 – DNS

Esta experiência teve por objetivo a configuração de DNS nos computadores da rede.

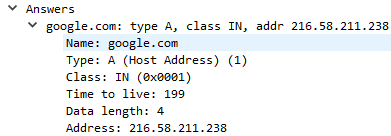
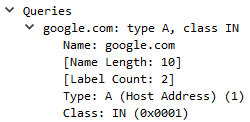
Para este efeito, bastou editar um ficheiro de configuração, para adicionar um *nameserver*, o qual traduz um *hostname* para um endereço IP.

Figura 9 - resolv.conf

**Que pacotes são trocados pelo DNS e que informação é transportada?**

Quando se executa um *ping* a um *hostname*, é enviado um pacote “DNS *standard query*” que contém parâmetros *name*, *type* e *class*. A resposta a este pacote, é um pacote “DNS *standard query response*” que contém, além dos parâmetros mencionados, um campo *Time to live* e *Data length*.

Figura 10 - exemplo de pacotes DNS



## Experiência 6 – Ligações TCP

Esta experiência teve por objetivo testar a aplicação de *download* na rede configurada. Para este efeito, foram testados diversos servidores FTP, nos quais, se experimentou fazer transferência de diferentes tipos de ficheiro, como por exemplo, zip, png e txt.

**Quantas ligações TCP são criadas pela aplicação?**

O protocolo FTP funciona com uma ligação de dados e uma ligação de controlo, por isso, são abertas duas ligações TCP por transferência.

**Em que ligação é transportada a informação de controlo de FTP?**

A informação de controlo é transportada na ligação TCP da porta 21, que é a porta por defeito de serviços FTP.

**Quais são as fases de uma ligação TCP?**

Uma ligação TCP pode consistir de três fases, as quais são:

* Um processo de *handshake* com vários passos, SYN -> SYN-ACK -> ACK.
* Uma fase de transferência de dados.
* Finalmente, uma fase de término de ligação, que consiste noutro *handshake*.

**Como funciona o mecanismo ARQ de TPC? Quais sãos os campos relevantes a TCP?**

O mecanismo ARQ de TCP funciona com uma “janela”, o recetor envia o tamanho desta “janela” no pacote TCP, e o emissor envia no máximo, até este valor de bytes. Após enviar os bytes, espera por um ACK do recetor e nova “janela”. Os campos relevantes no pacote TCP a este mecanismo são *Sequence number*, *Acknowledgment number* e *Window Size*.

**Como funciona o mecanismo de controlo de congestão de TCP? Quais são os campos relevantes?**

O mecanismo de congestão de TCP usa o sistema *slow-start*, o qual começa por enviar dados em número reduzido. Este número é duplicado a cada ACK recebido até ocorrer uma falha na transmissão, ao fim da qual, a ligação entra em fase de *congestion avoidance*, na qual o algoritmo utilizado varia com a implementação,o mais simplista sendo reduzir o número de bytes enviados de volta ao ínicio do *slow-start*. Os campos relevantes no pacote TCP são as *flags* CWR, ECE e URG, juntamente com o campo *Urgent pointer*.

**A taxa de transferência de uma ligação de dados TCP é perturbada pela aparência de uma segunda ligação TCP? Em que sentido?**

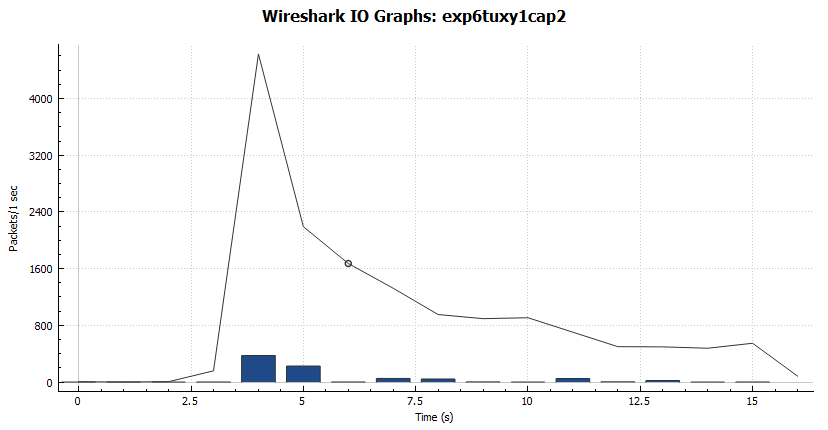
A aparência de uma segunda ligação reduz para cerca de metade a taxa de transferência, isto pode ser visualizado na captura seguinte.

Figura 11 - taxa de transferência

# Conclusão

Com a conclusão do relatório e da análise da aplicação de *download*, como das experiências laboratoriais, considera-se que o grupo conseguiu alcançar os objetivos do segundo trabalho.

Como descrito na primeira seção, o grupo implementou uma aplicação que utiliza o protocolo FTP, de acordo com o RFC959. Isto permitiu consolidar o conhecimento sobre FTP e sobre ligações TCP.

Na segunda seção foi descrita a configuração de uma rede, sendo que, o grupo aprendeu, os conhecimentos base necessários, para conseguir configurar redes complexas. Estes conhecimentos passaram por configurar endereços IP de computadores, implementação de VLAN num *switch* e configuração de rotas e NAT num *router* comercial. Finalizando, com a utilização da aplicação para efetuar uma transferência na rede, confirmando assim, a apreensão dos conhecimentos do trabalho.

Posto isto, é considerado que o trabalho foi bem sucedido e os seus objetivos, maioritariamente, compreendidos.

# Anexos

## Anexo 1 – Comandos de configuração

Para todos os comandos seguintes, y é o número de bancada da sala I321. Os comandos assumem uma configuração incremental começando na primeira experiência.

### Experiência 1

**Computador 1**

ifconfig eth0 down

ifconfig eth1 down

ifconfig eth0 172.16.y0.1/24

**Computador 4**

ifconfig eth0 down

ifconfig eth1 down

ifconfig eth0 172.16.y0.254/24

### Experiência 2

**Computador 2**

ifconfig eth0 down

ifconfig eth1 down

ifconfig eth0 172.16.y1.1/24

***Switch***

conf t

vlan y0

exit

vlan y1

end

conf t

interface fastethernet 0/1

switchport mode access

switchport access vlan y0

end

conf t

interface fastethernet 0/4

switchport mode access

switchport access vlan y0

end

conf t

interface fastethernet 0/2

switchport mode access

switchport access vlan y1

end

### Experiência 3

**Computador 1**

route add -net 172.16.y1.0/24 gw 172.16.y0.254

**Computador 2**

route add -net 172.16.y0.0/24 gw 172.16.y1.253

**Computador 4**

ifconfig eth1 172.16.y1.253/24

echo 1 > /proc/sys/net/ipv4/ip\_forward

echo 0 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_broadcasts

***Switch***

conf t

interface fastethernet 0/14

switchport mode access

switchport access vlan y1

end

### Experiência 4 (sem NAT)

***Switch***

conf t

interface fastethernet 0/20

switchport mode access

switchport access vlan y1

end

**Computador 1**

route add -net default gw 172.16.y0.254

route del -net 172.16.y1.0/24

**Computador 2**

route add -net default gw 172.16.y1.254

**Computador 4**

route add -net default gw 172.16.y1.254

***Router***

conf t

interface gigabitethernet 0/0

ip address 172.16.y1.254 255.255.255.0

no shutdown

end

conf t

ip route 172.16.y0.0 255.255.255.0 172.16.y1.253

end

conf t

interface gigabitethernet 0/1

ip address 172.16.1.y9 255.255.255.0

no shutdown

end

conf t

ip route 0.0.0.0 0.0.0.0 172.16.1.254

end

### Experiência 4 (NAT)

***Router***

conf t

interface gigabitethernet 0/0

ip address 172.16.y1.254 255.255.255.0

no shutdown

ip nat inside

exit

interface gigabitethernet 0/1

ip address 172.16.1.y9 255.255.255.0

no shutdown

ip nat outside

end

ip nat pool ovrld 172.16.1.y9 172.16.1.y9 prefix 24

ip nat inside source list 1 pool ovrld overload

access-list 1 permit 172.16.y0.0 0.0.0.255

access-list 1 permit 172.16.y1.0.0 0.0.0.255

ip route 0.0.0.0 0.0.0.0 172.16.1.254

ip route 172.16.y0.0 255.255.255.0 172.16.y1.253

end

### Experiência 5

**Computador 1/2/4**

Editar /etc/resolv.conf para conter:

"search netlab.fe.up.pt

nameserver 172.16.1.1"

## Anexo 2 – Capturas

TODO

## Anexo 3 – Código da aplicação de download

TODO