

Faculdade de Engenharia da Universidade do Porto

Configuração de rede e desenvolvimento de uma aplicação de download



2º Trabalho Laboratorial

Ana Sofia Teixeira

Rafael Cardoso Magalhães

L/M em Engenharia Electrotécnica e de Computadores
Redes de Computadores, 2021/2022, 2º Semestre

Índice

Sumário	3
1. Introdução	3
2. Parte 1 - Aplicação de download	3
2.1. Arquitetura	3
3. Parte 2 - Configuração e análise de rede	6
3.1. Experiência 1	6
3.2. Experiência 2	7
3.3. Experiência 3	8
3.4. Experiência 4	9
3.5. Experiência 5	10
3.6 Experiência 6	10
4. Conclusão	11

Sumário

Este relatório tem como objectivo demonstrar os conhecimentos adquiridos na Unidade

Curricular de Redes e Computadores ao longo da segunda metade do semestre. A realização deste trabalho laboratorial consiste na configuração de uma rede de computadores e no desenvolvimento de uma aplicação de download de um ficheiro.

Este documento está subdividido principalmente em duas partes: 1. A secção da aplicação de download onde é descrita a sua arquitectura e são apresentados os resultados. 2. A configuração da rede onde foram realizadas seis experiências tendo cada uma delas objetivos independentes.

1. Introdução

Este trabalho laboratorial tem como objetivos o desenvolvimento de uma aplicação de transferência de ficheiros, usando o protocolo FTP (*File Transfer Protocol*) - Parte 1 - e o estudo e configuração de uma rede de computadores - Parte 2.

Para a configuração da mesma foram usados três computadores, um Cisco Switch e um Router Cisco. As experiências acima referidas baseiam-se na configuração de um IP de rede, de um router em Linux, de um router comercial e do DNS (*Domain Name System*), e na implementação de duas LANs (*Local Area Network*) virtuais no switch e do NAT (*Network Address Translation*) e num teste com a aplicação de download desenvolvida para a verificação das ligações TCP.

2. Parte 1 - Aplicação de download

2.1. Arquitectura

Para uma melhor organização e estruturação do código, a aplicação de download está dividida em três partes: Main, Arguments e Connection.

☐ Connection

Neste módulo temos como primeira função `init()`, que é encarregada de, iniciar a conexão entre o cliente e o servidor ao criar um socket de conexão.

A função `sendCommand()` recebe um *file descriptor* de um socket e uma *string* com o comando a ser enviado (*command*). Com recurso à função `send()` de sockets, o comando é enviado para o servidor.

As funções `readResponse()` e `passiveMode()` usam `getline()` para esvaziar o buffer que contém a resposta do servidor. Em `readResponse()` há uma verificação do código de resposta para, se existirem erros, o programa termine. Em `passiveMode()` faz-se processamento da resposta com o envio do comando “pasv”. A função retira da mensagem de resposta um endereço IP e uma porta que retorna a partir dos argumentos.

A função `saveFile()` cria o ficheiro que irá ser transferido para a máquina e preenche o ficheiro com os dados que recebeu do servidor, os dados são adicionados até o buffer de leitura estar vazio.

□ Arguments

Neste módulo é feito o processamento dos argumentos da aplicação download.

A partir da função `parseArguments()` é guardada a informação da URL dos argumentos do download para uma struct “arguments”. No início da função são guardados os seguintes campos:

- Utilizador
- Palavra passe
- Host
- Path
- Nome do ficheiro
- Ip

No final são usadas as funções `getIP()` e `getFileName()` para a partir do URL saber o endereço IP do Host e o nome do Host, e para extrair o nome do ficheiro que se vai fazer download para no final do programa, ao criar o ficheiro transferido, saber que nome lhe colocar.

□ Main

O módulo main apenas contém a função `main()` com os passos necessários para fazer o download de um ficheiro.

O programa começa por verificar se apenas foi passado um argumento a partir da linha de comandos, se não tiver sido, apresenta uma mensagem de erro e o programa termina. Se o programa receber um URL, este é passado por argumento para a função `parseArguments()` que preenche a struct arguments com os componentes do URL. Uma lista de detalhes sobre a ligação aparece no terminal e o programa continua, chamando a função `init()`, que cria um socket para conseguir comunicar com o servidor. O programa procede com a chamada da função `readResponse()` para ler a mensagem de ligação e esvaziar o buffer de leitura.

Nesta altura inicia-se a sequência de login, na qual o programa envia o comando "user" com um nome de utilizador, lê a resposta com `readResponse()` e executa as mesmas duas funções com o comando "pass1234". Caso não sejam especificadas credenciais de login, o nome de utilizador e a palavra passe usadas é "anonymous" e a palavra passe usada é "pass". O programa executa o envio da palavra passe mesmo que não seja necessário, isto não afeta a troca de mensagens e simplifica o código. o próximo comando a enviar é "pasv" para pedir ao servidor que transfira dados em modo passivo, o que faz com que seja necessário abrir outra ligação.

A resposta a este comando é lida na função `passiveMode()` e contém o endereço IP e a porta à qual se realizará a nova ligação. Esta é feita via `init()` e devolve um novo file descriptor do socket de onde se retira os dados. Para finalizar é enviado o comando "retr *path*" com o caminho para o ficheiro e o programa chama a função `saveFile()` para guardar num ficheiro os dados lidos a partir do novo socket.

```
sofiat@sofiat-GF65-Thin-95D:~/Documents/rcom2$ gcc *.c
sofiat@sofiat-GF65-Thin-95D:~/Documents/rcom2$ ./a.out ftp://:@ftp.up.pt/raspbian/snapshotindex.txt
host: ftp.up.pt
url path: raspbian/snapshotindex.txt
user: anonymous
password: pass1234
file name: snapshotindex.txt
host name: mirrors.up.pt
ip addr: 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
< user anonymous

331 Please specify the password.
< pass pass1234

230 Login successful.
< pasv

227 Entering Passive Mode (193,137,29,15,229,121).
IP: 193.137.29.15
Port Number: 58745
< retr raspbian/snapshotindex.txt

150 Opening BINARY mode data connection for raspbian/snapshotindex.txt (494843 bytes).
< quit

226 Transfer complete.
sofiat@sofiat-GF65-Thin-95D:~/Documents/rcom2$
```

Figura 1 - Resultado do teste da aplicação de download

3. Parte 2 - Configuração e análise de rede

3.1. Experiência 1

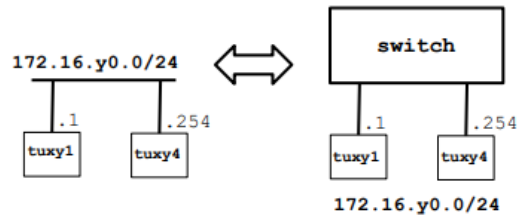


Figura 2 - Arquitetura da Experiência 1.

Esta experiência tem como objetivo dar a conhecer aos estudantes o ambiente de configuração e análise de rede, percebendo melhor os conceitos de IP, MAC e ARP.

Pacotes ARP são usados para pedir o endereço MAC de uma máquina, sabendo o seu endereço IP. Um endereço IP é um Identificador Público usado pela máquina numa rede específica. Uma máquina pode ter vários endereços IP. Por outro lado, apenas pode ter um endereço MAC, que é o identificador da placa de rede da máquina.

Iniciou-se a experiência ligando o tux43 e tux43 ao switch, através de cabos. Depois, configurou-se as saídas das máquinas, e os IPs das mesmas, como mostra a figura3. Após a configuração, fez-se um ping do tux43 para o tux44, e obteve-se resposta. Ao analisar os logs desta experiência, foi possível concluir que o ping gera pacotes ARP e ICMP.

Reparou-se também que os pacotes ARP, que têm como objetivo obter o endereço MAC da máquina, foram enviados antes dos pacotes ICMP.

Código configuração tux43:

```
//ip tux43
ifconfig eth0 up
ifconfig eth0 172.16.40.1/24
```

Código configuração tux44:

```
//ip tux44
ifconfig eth0 up
ifconfig eth0 172.16.40.254/24
```

3.2. Experiência 2

Esta experiência tem como objetivo a criação de duas VLAN's e a configuração das mesmas no switch, de forma a configurar e compreender melhor a conectividade entre os vários tux's.

Iniciou-se a experiência efetuando a configuração das VLAN's. Para isso, escolheu-se um tux, e liga-se a sua porta S0 à porta do switch. Através do GTKTerm, configuram-se as VLAN's, utilizando os comandos apresentados em baixo. De seguida, e mantendo-se as mesmas ligações da experiência anterior, acrescentou-se uma ligação: o tux42 a uma porta da direita do switch(vlan 41,porta 13), e configurou-se o IP do mesmo.

Após a configuração, testaram-se as ligações através de pings. Como previsto, era possível efetuar ping broadcast entre os tux43 e tux44, mas não era possível efetuar qualquer tipo de comunicação com o tux42. Isto deve-se ao facto de o tux43 e tux44 estarem no mesmo domínio de transmissão.

Código de criação de VLAN's:

```
//Create VLAN 40 & 41
configure terminal
vlan 41
end
configure terminal
vlan 40
end
configure terminal
interface fastethernet 0/1
switchport mode access
switchport access vlan 40
end
configure terminal
interface fastethernet 0/3
switchport mode access
switchport access vlan 40
end
configure terminal
interface fastethernet 0/13
switchport mode access
switchport access vlan 41
end
```

Código configuração tux42:

```
//ip tux42
ifconfig eth0 up
ifconfig eth0 172.16.41.1/24
```

3.3. Experiência 3

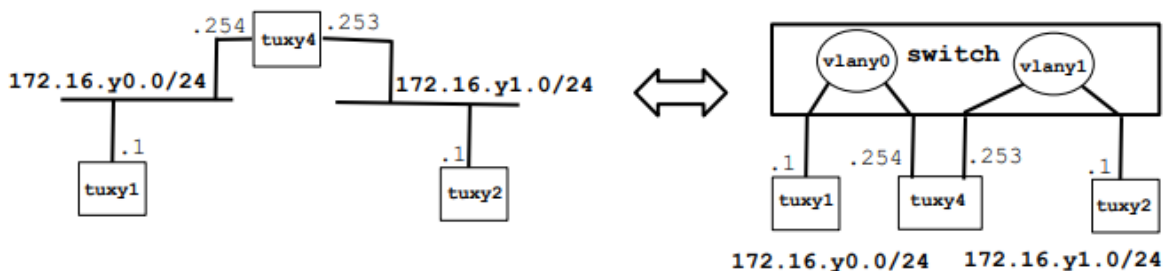


Figura 3 - Arquitetura da Experiência 3

Esta experiência tem como objetivo utilizar o tux44 como gateway para a comunicação entre o tux42 e tux43, através das VLAN's implementadas.

Começou-se por configurar um novo IP no tux44, que estaria conectado a uma porta da VLAN41 (porta 15). Efetuou-se a ligação dos cabos entre a porta eth1 do tux44 e a porta 15 da VLAN41. Para além disso, deu-se enable ao ip forwarding no tux44.

Após esta configuração, o tux42 tem uma rota para a VLAN40 pela gateway 172.16.41.253, e o tux43 tem uma rota para a VLAN41 através da gateway 172.16.40.254.

Como tal, é possível enviar pings do tux43 para o tux42, uma vez que o tux44 serve como gateway. O tux44, por sua vez, também consegue enviar e receber respostas para ambos (tux43 e tux42) uma vez que está ligado às duas VLAN's existentes.

Código configuração tux42:

```
//ip tux42
route add -net 172.16.40.0/24 gw 172.16.41.253
```

Código configuração tux44:

```
//ip tux44
ifconfig eth1 up
ifconfig eth1 172.16.41.253/24
echo 1 > /proc/sys/net/ipv4/ip_forward
```


3.4. Experiência 4

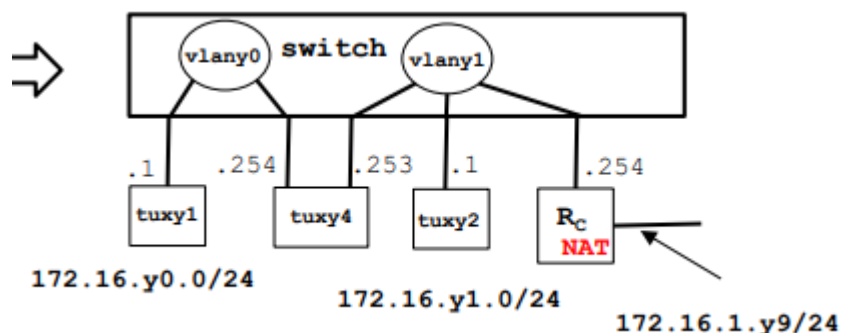


Figura 4 - Arquitetura da Experiência 4

Esta experiência tem como objetivo a configuração de um router comercial, estabelecer rotas no mesmo, e estabelecer uma ligação com a rede do laboratório, implementando a funcionalidade NAT.

O NAT tem como objetivo a conservação e proteção de endereços IP. Sendo uma funcionalidade do router, o NAT é um mecanismo que substitui os endereços IP's das diversas máquinas conectadas ao router por um IP público, de forma a possibilitar a ligação para fora da rede.

Iniciou-se esta experiência, ligando-se o router a uma das portas S0 de um dos tux, de forma a conseguir configurar o router comercial. Através do GTKTerm, inseriram-se os comandos escritos abaixo. Testou-se efetuar pings do tux43 para o tux42, tux44 e para o Rc, todos obtendo resposta com sucesso.

Código configuração tux44:

```
//ip tux44
route add default gw 172.16.41.254
```

Código de configuração do router:

```
//Router
conf t
interface gigabiteethernet 0/0
ip address 172.16.41.254 255.255.255.0
no shutdown
ip nat inside
exit
conf t
interface gigabiteethernet 0/1
ip address 172.16.1.49 255.255.255.0
no shutdown
ip nat outside
exit
ip nat pool ovrld 172.16.1.49 172.16.1.49 prefix 24
ip nat inside source list 1 pool ovrld overload
access-list 1 permit 172.16.40.0 0.0.0.7
access-list 1 permit 172.16.41.0 0.0.0.7
ip route 0.0.0.0 0.0.0.0 172.16.1.254
ip route 172.16.40.0 255.255.255.0 172.16.41.253
end
```

3.5. Experiência 5

O objectivo desta experiência era conseguir aceder a redes externas, conseguindo desta forma aceder à Internet, através da rede interna criada. Para isto, foi necessário configurar o serviço DNS.

Para configurar o serviço DNS no host basta editar o ficheiro `/etc/resolv.conf` através do comando no terminal:

```
nano /etc/resolv.conf
```

e colocar lá as linhas correspondentes aos servidores DNS, neste caso, basta adicionar: `nameserver 172.16.2.1`.

Os pacotes DNS (*Domain Name System*) utilizam pacotes UDP (*User Datagram Protocol*) para resolver nomes, ou seja, traduz um nome como `google.pt` para o seu endereço IP correspondente. Para isto são enviados dois pacotes: um para resolver o nome em IPv4 e outro para resolver o nome em IPv6. No caso do nome `ftp.up.pt` os pacotes de resposta informam, também, que o nome é um alias para `mirrors.up.pt`, ou seja, estes nomes correspondem ao mesmo endereço IP.

3.6 Experiência 6

Por fim, na experiência 6, compilou-se e executou-se a aplicação desenvolvida e descrita na primeira parte do relatório. Para testar a aplicação, foi usado um servidor ftp e efetuado o download de um ficheiro.

A aplicação FTP abre duas ligações TCP uma para enviar comandos para o servidor ftp e outra para receber dados. A informação de controle é transportada na primeira ligação feita pela aplicação FTP para a porta 21 do servidor.

Uma ligação TCP tem 3 fases: estabelecimento de ligação, comunicação entre os end hosts e encerramento da ligação. O ARQ TCP é um mecanismo de controlo de erros na transferência de dados, que utiliza respostas do tipo ACK, indicando que os dados enviados estão corretos, e timeouts, tempo especificado para que o emissor receba o ACK do receptor relativamente ao pacote enviado anteriormente. No caso do emissor não receber a resposta ACK antes do timeout ocorrer o pacote é dado como perdido ou corrompido e reenviado pelo emissor, este mecanismo garante que este processo é repetido até ser recebido uma resposta ACK antes do timeout ou seja atingido o máximo de retransmissões possíveis.

Os campos importantes para este mecanismo estão no cabeçalho dos pacotes TCP, são estes: a flag ACK; o campo checksum que garante a integridade dos dados; os campos de sequência e acknowledge que garantem que os dados são entregues na ordem certa.

4. Conclusão

O primeiro objetivo deste trabalho, a implementação de uma aplicação de transferência usando o protocolo FTP, foi um pouco mais desafiante do que o inicialmente expectado, uma vez que estávamos a experienciar algumas dificuldades em fazer download de outro tipo de ficheiros sem ser os do tipo “.txt”. No entanto, o segundo objetivo do trabalho, referente às experiências mencionadas anteriormente, foi realizado na sua totalidade durante as aulas atribuídas para a realização das mesmas.

A configuração passo a passo de uma rede fez com que fosse possível, de uma forma mais prática, perceber os mecanismos envolvidos e o impacto real que tem nas comunicações atualmente.