Resumo

No âmbito da unidade curricular de Redes de Computadores, RCOM, do terceiro ano do Mestrado Integrado em Engenharia Informática e Computação, MIEIC, desenvolveu-se um projeto capaz de implementar um cliente FTP capaz de executar o download de um ficheiro, com auxílio a várias experiências laboratoriais executadas durantes as próprias aulas práticas da unidade curricular.

Através deste projeto conseguimos interiorizar alguns conceitos relativos à configuração e estudo de uma rede de computadores, principalmente conhecimentos relativos a configurações relativas a um switch e router comercial.

Introdução

No âmbito da unidade curricular de redes de computadores, foi proposto ao grupo a implementação de uma aplicação cliente de um servidor FTP, capaz de fazer o download de um ficheiro, a especificar pelo utilizador, com recurso a diversas experiências laboratoriais de forma a se conseguir perceber como é que se configura e como funciona uma rede de computadores.

As experiências laboratoriais executadas deram a conhecer os comandos necessários para configurar uma rede, aprofundando os conhecimentos teóricos abordados na unidade curricular.

Este relatório contém um capítulo referente a cada parte do trabalho, em que a primeira parte aborda a criação de um parser para que seja posteriormente efetuado o download do ficheiro e a segunda parte se destina às experiências laboratoriais, onde cada subsecção é uma experiência respetiva.

Cliente FTP

Foi desenvolvida uma aplicação em linguagem C, com o objetivo de fazer o download de um ficheiro de um servidor FTP. Após se ter compilado o makefile do ficheiro, apresentado no ficheiro [FTP.zip](ftp://FTP.zip) em anexo a este relatório), deve-se correr o seguinte comando, onde o URL segue a sintaxe descrita no RFC1738:

./download ftp://[<user>:<password>@]<host>/<url-path>

Inicialmente, a aplicação começa por interpretar o URL respetivo, através de parser.h e parser.c), extraindo o utilizador, password, host e o diretório do ficheiro a descarregar. Caso o utilizador queira fazer o download em modo anónimo basta chamar o URL com a sintaxe:

ftp:// <host>/<url-path>

Depois de todas estas informações serem armazenadas, a aplicação tenta converter o host num endereço IP, através da função getHostIP() pertencente a ClientFTP.h e ClientFTP.C. Esta função foi fornecida aquando da apresentação do guião referente ao segundo trabalho e tem como objetivo retornar todos os IP’s possíveis para criar um conexão ao servidor FTP.

De seguida, tenta-se estabelecer a conexão, utilizando a porta 21, e caso a conexão seja estabelecida com sucesso, é criado um socket para que seja possível comunicar com o servidor.

Caso a ligação seja estabelecida, uma mensagem de boas vindas proveniente do host é apresentado ao utilizador, através do terminal.

Assim, após todos estes passos é executado o login, utilizando o nome do user, através do comando USER, e a sua respetiva password, através do comando PASS, em que caso estas não existem, romam valores default onde o user seria anonymous e a sua password mail@domain.

Se o login for executado com sucesso, tenta-se entrar um modo passivo, usando o comando PASV. Caso a entrada em modo passivo seja feita com sucesso, o cliente FTP envia o comando RETR e por fim, tenta-se abrir o ficheiro a descarregar, iniciando-se assim o download propriamente dito.

Por fim, faz-se o logout, utilizando o comando QUIT, de forma a concluir o download e a finalizar a aplicação.

Experiências Laboratoriais

Experiência 1 – Configurar uma rede de IP’s

Esta primeira experiência tem como objetivo fazer com que saibamos distinguir diferentes pacotes de dados e as suas finalidades, respetivas.

Através da análise dos logs, verificámos que um dos pacotes de dados são os pacotes ARP. Estes pacotes permitem encontrar um endereço da camada de ligação (MAC) através de um IP.

O primeiro pacote ARP, é enviado para todos os computadores, contendo o IP para o qual é pretendido saber o MAC. Para que tal seja obtido, deve-se enviar o pacote em broadcast, a partir da máquina que deseja receber os dados.

No segundo pacote ARP, a máquina cujo MAC é o endereço alvo do primeiro pacote, envia o seu endereço MAC à máquina que pretendia esta informação.

Outros pacotes de dados verificados foram os pacotes ICMP, gerados através do comando ping. Estes pacotes têm dois IP’s associados, um relativo à fonte e outro relativo ao destino.

Para que seja, portanto, possível distinguir a trama recebida (ARP, IP ou ICMP), é necessário observar o cabeçalho da mesma e tentar analisá-lo. É, igualmente, através do cabeçalho que podemos descobrir o tamanho dos pacotes de dados.

O mecanismo de loopback permite detetar erros na transmissão de dados, uma vez que o emissor volta a receber o pacote de dados que tentou enviar. Este mecanismo permite, também, saber se os cabos de ligação se encontram em boas condições de uso.

Experiência 2 – Implementar duas LAN’s virtuais num Switch

A segunda experiência teve como objetivo ensinar a configurar o switch e, posteriormente, criar duas LAN’s virtuais.

Para se conseguir configurar um VLAN, primeiro tem que se abrir o modo de configuração, utilizando “configure terminal” e depois executa-se o comando “vlan x”, onde x é o número da vlan respetiva, na consola do switch.

De seguida, tem que se adicionar as respetivas portas em que os tuxs estão ligados no switch, selecionando a porta na mesma consola através do comando “interface fastethernet 0/y”, onde y é o número da porta a adicionar. Após o uso do comando referido, é preciso mudar o modo “switchport mode access” e por fim, adicionar a pota a uma VLAN com o comando “switchport access vlan x”, onde x é o número da VLAN.

Através da observação e análise dos logs, podemos concluir que há um broadcast domain para cada VLAN. Esta conclusão foi obtida, visto que fazendo broadcast no tux1 apenas os tuxs da mesma VLAN recebiam os pacotes de dados, verificando-se o mesmo para o tux2.

Experiência 3 – Configurar um router em Linux

A terceira experiência teve como objetivo converter o tux4 num router.

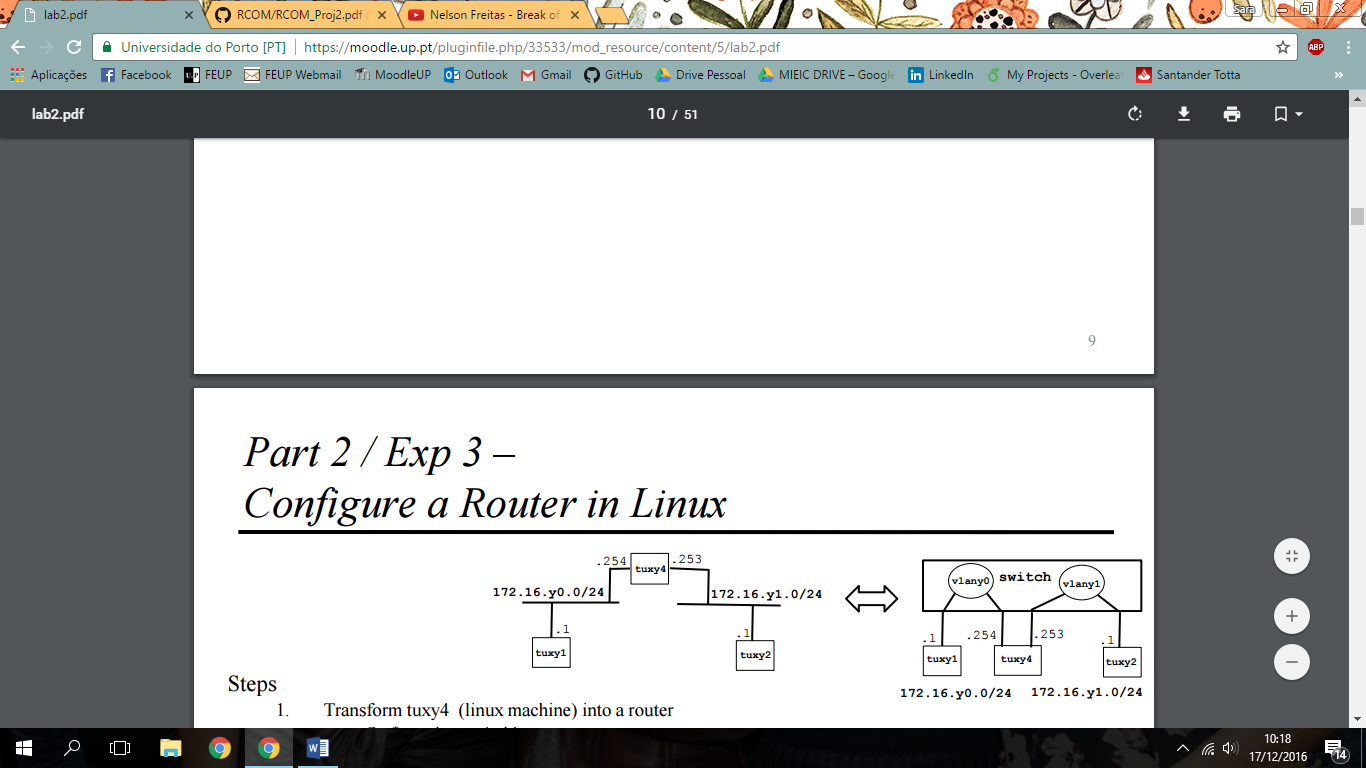
No tux1 temos a rota da interface eth0, ou seja, a rota que é automaticamente criada quando é atribuído o endereço IP a essa mesma interface e uma outra rota para a rede 172.16.Y1.0 (onde Y é o número da bancada, neste caso seria 172.16.41.0, visto que a bancada era a 4), através da gateway 172.16.Y0.254 (no caso deste grupo a gateway era 172.16.40.254).

No tux2 tems a rota da interface eth0, ou seja, a rota que é automaticamente criada quando é atribuído o endereço IP a essa mesma interface e uma outra rota para rede 172.16.Y0.0 (onde Y é, novamente, o número da bancada, neste caso seria 172.16.40.0), através da gateway 172.16.Y1.253 (no caso deste grupo a gateway era 172.16.41.253).

No tux4 temos as rotas das interfaces eth0 e eth1, sendo que as redes 172.16.Y0.0 e 172.16.Y1.0, ou seja, as redes 172.16.40.0 e 172.16.41.0 neste caso, são alcançáveis por este mesmo tux.

No passo 11 desta experiência, parou-se de capturar os logs respetivos, podendo observar que os pacotes de dados ARP têm como objetivo informar o tux4 quais os endereços MAC que estão associados ao IP de destino.

No início, o tux1 verifica quem tem o IP 172.16.Y0.254, neste caso 172.16.40.254. De seguida, o tux4 verifica quem tem o IP 172.16.Y1.1 e no caminho oposto, o tux2 verifica quem tem o IP 172.16.Y1.253 e o tux4, verifica numa outra vez, quem tem o IP 172.16.Y0.1.



Já os pacotes ICMP têm sempre o mesmo IP de fonte e destino para que ao passarem pelos diferentes tuxs saberem sempre onde têm de reencaminhar os pacotes.

Experiência 4 - Configurar um router comercial e implementar NAT

A experiência 4 consistia em configurar um router comercial e implementar NAT, percebendo melhor esta funcionalidade.

Para que se pudesse adicionar uma rota estática num router comercial, bastava escrever o seguinte comando “ip route [rede de destino] [máscara de rede] [gateway IP]”, na consola do router.

Neste experiência, caso o tux2 tenha uma rota para a rede 172.16.Y0.0, então os pacotes percorrem o seguinte trajeto tux2 -> tux4 -> tux1. Se essa rota for removida, os pacotes percorrem o caminho tux2 -> router comercial -> tux4 -> tux1. No entanto se ativarmos os “redirects” no tux2, na primeira vez o trajeto será tux2 -> router comercial -> tux4 -> tux1, contudo nas restantes vezes será tux2 -> tux4 -> tux1.

Nesta experiência é, igualmente, pedido para adicionar a funcionalidade NAT ao router e para isso basta selecionar, na consola do router, a interface pretendida através do comando “interface gigabitethernet 0/X”, onde X é o número da interface e posteriormente executar o comando “ip nat outsider” para que a interface esteja ligada ao router da sala. Com “ip nat inside” as rotas são primeiro encaminhadas e depois interpretadas, sendo que no caso se “nat outsider” se verifica o oposto.

O NAT é uma funcionalidade que permite reescrever os IP’s de origem e destino dos pacotes recebidos num router de forma a conectar uma rede local a uma rede externa. Assim, quando os pacotes de dados chegam ao router, provenientes da rede externa, ele sabe qual é o tux de destino, uma vez que este mantém uma hash table contendo essa informação.

Experiência 5 – DNS

A experiência cinco teve como objetivo configurar o servidor DNS, assim verificar qual é o seu papel nas redes de computadores.

Para configurar o servidor DNS tem-se que editar o ficheiro /etc/resolv.conf, especificando o search (lixa.netlab.fe.up.pt) e o nameserver (172.16.1.1). Estes valores podem ser alterados utilizando oo comando “echo” search lixa.fe.up.pt\nnameserver 172.16.1.1.

O servidor DNS é responsável por converter um endereço web, como por exemplo [www.google.pt](http://www.google.pt), num endereço IP do tipo 8.8.8.8.

Um primeiro pacote é enviado através do tux1 ao servidor DNS, com o domínio do website e este responde com o respetivo IP. Por sua vez, um segundo pacote envia uma lista contendo todos os servidores com aquele nome associado.

Experiência 6 – Conexões TCP

A experiência 6 tem como objetivo testar a aplicação FTP desenvolvida pelo grupo.

Esta aplicação abre duas ligações TCP. A primeira envia a informação do protocolo FTP, para se conseguir enviar pedidos ao servidor, como o username, password e ficheiro que pretendemos receber. A segunda ligação destina-se a transferir a informação do ficheiro que se escolheu para transferir.

A conexão TCP tem 4 fases recaindo sobre o estabelecimento da ligação, aceitação e rejeição de pacotes e término da ligação estabelecida.

O protocolo TCP utiliza um mecanismo ARQ bastante semelhante ao Go-Back-N, utilizando uma Sliding Window. Este é um mecanismo que em caso de erro na transmissão respetiva, tentar reenviar um pacote de cada vez, uma vez que assume que só um foi perdido. Os campos relevantes para este método são o ACK, tamanho da janela e número de sequência, sendo igualmente importante saber qual foi o último byte escrito, para se conseguir atualizar o tamanho da Sliding Window.

Para que o congestionamento de informação seja evitado, o protocolo TCP calcula se é possível enviar a informação sem ultrapassar o limite do buffer, o que caso aconteça faz com que bloqueie a aplicação até existir mais espaço disponível.

Conclusão

Durante a elaboração da aplicação de download, utilizando o protocolo FTP e da realização das experiências laboratoriais,a compreensão do guião respetivo tornou-se importante para que todos os objetivos proposto fossem cumpridos. Igualmente, foi fucral o apontamento de diversas informações e comandos para que cada fase das experiências fossem cumpridas com precisão de forma a que não se comprometesse as experiências seguintes.

No final da realização das duas partes do guião, o grupo conseguiu perceber como é que funcionava uma rede, neste caso, com um tamanho bastante reduzido, apercebendo-se da complexidade e trabalho que seria necessário para a criação de uma rede de elevada complexidade e dimensão.