**Trabalho 2**

Configuração de uma rede e desenvolvimento de uma aplicação de download

**Relatório Final**

Mestrado Integrado em Engenharia Informática Computação

Redes de Computadores

Turma 4:

André Rodrigues Barros – up201303567

Leonardo Ferreira – up2013

Luis Miguel Azevedo Duarte – up2013

Faculdade de Engenharia da Universidade do Porto Rua Roberto Frias, 4200-465 Porto, Portugal

**Resumo**

Este relatório serve como complemento do segundo projeto da unidade curricular de Redes de Computadores, do MIEIC, na FEUP. O objectivo deste projeto era de configurar uma rede de computadores e no desenvolvimento de uma aplicação de download de ficheiros, logo este relatório encontrar-se-á divido nessas duas respetivas metades, acompanhadas das suas respectivas subsecções.

A secção que refere a configuração de rede, foi abordada em seis experiências distintas propostas ao grupo. Tais experiências consistiam numa configuração de um **IP de rede**, de um **router em Linux**, de um **router comercial** e do **DNS**, e na implementação de duas **VLAN’s** (virtual Local Area Network)e do **NAT** (Network Address Translation), e de seguida, testar a aplicação de download desenvolvida para verificação de um bom funcionamento nas ligações **TCP** (Transmission Control Protocol), sendo que todos estes conceitos e protocolos irão ser explicados com mais detalhe ao longo do relatório.

**Introdução**

Este segundo projeto divide-se em duas componentes:

- Configuração de uma rede

- Desenvolvimento de uma aplicação de download

O grande objetivo da configurar uma rede é permitir a execução de uma aplicação, neste caso a aplicação de download desenvolvida, num switch partilhado por duas *VLAN’s*, estando o *NAT* ativo numa destas. No entanto, para o correto funcionameto da aplicação a *VLAN* quer se encontra sem *NAT* terá de possuir uma ligação à Internet.

Olhando agora para a aplicação de download, o objetivo passava por ganhar um conhecimento mais profundo sobre o que é um cliente, um servidor e as suas especificidades em TCP/IP, a caracterização de protocolos em aplicações, como definir um URL e descrever o comportamento de um servidor FTP. De seguida, o grupo teria de implementar um cliente FTP e ligação TCP utilizando sockets.

Parte 1 – Aplicação Download

* 1. Arquitetura
  2. Exemplo de resultados do download com a respetiva explicação

Aplicação do download

Uma das componentes a avaliar no segundo projeto de Redes de Computadores era o desenvolvimento de uma aplicação, em C, que permitisse fazer downloads de um servidor. Para o desenvolvimento desta aplicação, o grupo estudou vários links e documentos, entre os quais o RFC959 que explica detalhadamente o protocolo de transferência de dados, ou FTP.

Iremos abordar detalhadamente todos os aspetos que envolvem esta aplicação assim como o método para a sua implementação.

* 1. Arquitetura

A aplicação download está dividida em 2 blocos: um dos blocos trata de extrair toda a informação necessária de um URL; outro contém todas as funções necessárias para o FTP assim como as responsarias pela ligação dos sockets aos servidores.

Estes dois blocos são conjugados no project.c permitindo ao utilizador descarregar um ficheiro de um servidor.

*Primeiro Bloco:*

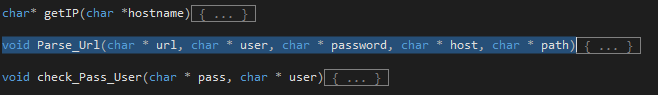
 É composto pelos ficheiros URL.c e URL.h e permite á aplicação aceder á informação passada como parâmetro aquando na execução do download na linha de comandos.

Figura 1: Funções responsáveis pela extração da informação do URL assim como do IP do host.

Funções:

**getIP** -> permite á aplicação aceder ao ip do servidor. Este processo dá-se graças à função gethostbyname que retorna uma estrutura do tipo hostent, que é usada na função inet\_ntoa através de um cast para a estrutura do tipo in\_ddr e é devolvida um char\* no formato de pontos e números relativos ao IP. Recebe como parâmetro o host;

**Parse\_Url** -> Faz o parse de toda a informação exposta pelo link passado na linha de comandos. Guarda a informação do user, password, host e do path do ficheiro que pretende fazer download. Nesta função, também é considerado o facto de o link poder não ter paswords ou não ter user e password (neste caso o utilizador entraria em modo anonymous). Caso falte argumentos como o host ou path do file no link, então é lançada uma mensagem de erro na consola. Recebe como parâmetros o user, a password, o host, o path assim como o url;

**Check\_Pass\_User** -> apenas faz print na consola para informar o utilizador se entrou em modo anonimo ou se não colocou a password no link passado na linha de comandos. Recebe como parâmetros o user e password do utilizador

*Segundo Bloco:*

Bloco que tem uma grande preponderância na aplicação do download e que é representado pelo clientFTP.c.

Após o parse do Url, é necessário ligar o cliente servidor FTP através de um socket TCP ao servidor FTP. Esta ligação é realizada pela função **connect\_server.**

De seguida, apresentaremos todas as funções necessárias assim como todos os passos a proceder para descarregar um ficheiro descrito no path inidicado pelo utilizador.

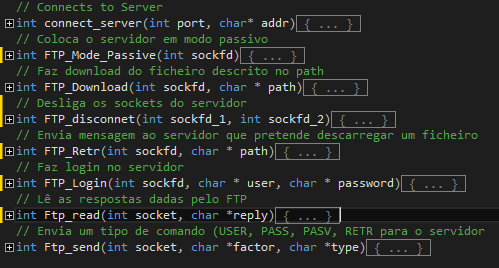


Figura 2: Funções que permitem aceder ao servidor e obter o ficheiro pretendido

Funções:

**connect\_server** -> responsável por estabelecer a ligação via sockets a um servidor. Recebe dois parâmetros: o endereço a que tem que se ligar assim como a porta (na primeira ligação que se estabelece com o servidor deve ser utilizada, por defeito, a porta 21 por se tratar de FTP). Se a ligação for bem-sucedida, é retornado o descritor da ligação que será necessária para os restantes processos;

**FTP\_Login** –> envia-se os comandos user e password ao servidor com a respetiva informação, através do Ftp\_Send, e recebe-se a resposta por parte do servidor, pela função Ftp\_read. Se após a transmissão do dados para autenticação, recebermos o código 530 (código que indica que indica que os dados de login estão errados) então o programa é terminado. Os parâmetros correspondem ao socket, ao user e á password;

**FTP\_Mode\_Passive** -> envia-se o comando PASV ao servidor. Uma vez feito este envio, recebemos uma resposta com o novo IP ao qual a aplicação download se deve associar para proceder á transferência do ficheiro assim como a nova porta a ligar. De seguida, repetimos o processo descrito na função connect\_server mas desta vez com o novo IP, com a nova porta associada a um novo socket (é calculada com base na resposta dada pelo servido após o envio do comando PASV). Recebe como parâmetro o socket;

**FTP\_Retr** -> envia-se o comando RETR ao servidor, através do primeiro primeiro socket com o path do ficheiro a descarrega. Toma como parâmetros o path do ficheiro e o socket;

**FTP\_Download** -> Cria-se um ficheiro com o mesmo nome que o ficheiro pretendido. De seguida, em cada iteração da leitura, é escrito para o novo ficheiro a informação contida no ficheiro original. Toma como parâmetros o path do ficheiro e o socket;

**FTP\_Disconnect** –> Após receber o o ficheiro corretamente (o utilizador recebe o código 226 que significa que recebeu o ficheiro corretamente), é enviado ao utlizador o comando QUIT que permite desconectar os sockets do servidor e fecha-se os sockets. Recebe os sockets envolvidos nas operações efetuadas pela aplicação.

* 1. Exemplo dos resultados da aplicação download

Testou-se a aplicação donwload com vários ficheiros de diferentes tipos e tamanhos e tudo correu como esperado.

Executou-se na consola -> ./proj <ftp://ftp:pass@ftp.up.pt/pub/robots.txt>

Output esperado na consola após execução da aplicação.

Info Aplication Download

User: ftp

Password: pass

Host: [ftp.up.pt](ftp://ftp.up.pt)

Path to file: pub/robots.txt

Start Connection…

IP Address: 193.136.37.8

Login…

Reply: 220 Bem-vindo à Universidade do Porto

Element sent: USER ftp

Reply: 331 Please specify the password.

Element sent: PASS pass

Reply: 230 Login successful.

Enter Pasv mode…

Element sent: PASV pasv

Reply: 227 Entering Passive Mode (193,136,37,8,180,67)

New IP Address: 193.136.37.8

Starts a new Connection…

Element sent: RETR pub/robots.txt

Reply: 10 Opening BINARY mode data connection for pb/robots.txt (23 bytes).

Filename described by path: robots.txt

New File received successfuly!!!

Reply: 226 File send OK.

Element sent: QUIT quit

Reply: 221 Goodbye

Disconnected from server

**Parte 2 – Configuração de rede**