**Redes de Computadores**

**2º Trabalho Laboratorial – Rede de computadores**

**Mestrado Integrado em Engenharia Informática e computação**

**21 de Dezembro de 2020**

****

**Diogo Guimarães do Rosário - up201806582**

**Henrique Melo Ribeiro - up201806529**

**Sumário**

Este projeto teve como objetivo criar uma rede de computadores e um programa de download de ficheiros por ftp.

Este relatório divide-se em duas partes:

- Descrição da aplicação de download;

- Configuração da rede em função das 6 experiências propostas.

As experiências acima referidas consistem em configurar um **IP de rede,** de um sistema de **router** em Linux, de um router comercial juntamente com um sistema de **DNS** e na implementação de duas **Virtual lans** através do switch e do NAT e num teste de download usando a configuração final e a aplicação de download criada.

**Introdução**

Este projeto foi realizado no âmbito da disciplina Redes de Computadores com o objetivo de obter um melhor entendimento sobre o desenvolvimento de uma aplicação de download em modo FTP e sobre a criação/análise de uma rede.

O trabalho baseou-se na criação de uma rede de computadores e posterior ligação da aplicação de download desenvolvida pelo nosso grupo. Desta forma o nosso relatório estará dividido da seguinte forma:

-Parte 1 - Aplicação de download:

- Arquitetura da aplicação de download;

- Analise de um download com sucesso:

- Parte 2 – Configuração da rede e análise:

- Arquitetura da configuração, objetivos da experiência, comandos das configurações principais;

- Análise dos logs obtidos através do *Wireshark*.

- Conclusões.

- Anexos.

**Parte 1 – Aplicação de download.**

Para testar o bom funcionamento da rede configurada foi preciso criar uma aplicação de download capaz de criar uma conexão FTP com um endereço fornecido pelo utilizador e transferir o ficheiro especificado utilizando as normas RFC959 (leitura e analise das respostas do servidor) e RFC1738 (tratamento e utilização dos endereços URL).

De seguida iremos descrever com mais detalhe a implementação do programa bem como algumas das suas funcionalidades e algumas analises dos resultados.

* 1. **– Arquitetura da aplicação de download.**

Para a melhor distribuição de funcionalidades o grupo decidiu dividir a aplicação em duas partes, uma em que faz o tratamento do input do utilizador e outra para realizar a conexão FTP com o servidor fornecido. O servidor é fornecido como único argumento de execução sendo que pode ou não conter também um utilizador e password para este.

O formato para execução da aplicação deve ser o seguinte:

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

Este argumento é tratado na função getURL(username, password,  returnHost,  file,  argv).

Esta função utilizada uma expressão regex para validar o formato do link fornecido e varias chamadas à função strtok para separar os vários campos deste.

Após o tratamento do input, os dados da função que são recebidos como pointer são preenchidos com os valores corretos de forma a conseguir realizar a conexão FTP.

Após este tratamento do input é feita a conexão com o servidor, utilizando funções fornecidas pelos professores ( **getHostName(char \*host)** que é responsável por obter a informação sobre o host, como o seu IP, e um conjunto de funções como **socket** e **connect** que criam um TCP socket e conectam-no com o servidor desejado).

Após esta conexão é feito um loop while que lê a mensagem enviada pelo servidor como resposta à ligação do socket com este, sendo esta mensagem processada na função **parseConnection(char\* buf**). Esta função verifica se os três primeiros bytes da mensagem correspondem ao código 220 que simboliza que a conexão foi estabelecida corretamente. Para alem disto, também verifica se, após o código 220 existe um ‘-‘, caso exista é sinal que o servidor ainda vai enviar mais mensagens, se não existir significa que o servidor está à espera de um input.

Quando a função **parseConnection** indicar que a resposta do servidor terminou é utilizada a função **sendUserPass(int sockfd, char \*user, char \*pass)** que se responsabiliza por enviar os credenciais ao servidor caso estes tenham sido indicados, caso não tenham sido indicados é indicado que a conexão é feita em modo anónimo.

Em primeiro lugar é enviado o utilizador para o servidor através da função **write** para o file descriptor do socket. Após a escrita do user é esperada uma resposta do servidor com o código 331. Caso este código esteja presente é enviada a password ao servidor, caso seja outro código a execução da aplicação é cancelada. Se todos os credenciais forem aceites é enviado o comando para o servidor entrar em modo passivo, retornando o **serverPort** para a segunda conexão.

Caso o servidor consiga entrar em modo passivo é feita uma nova conexão para o download do ficheiro ser realizado, utilizando o valor de retorno da função anterior. Este download é feito na função **getFile** na qual é enviado o comando **retr** para o servidor, juntamente com o path para o ficheiro a ser transferido. Caso o ficheiro exista, é iniciado o download do ficheiro através de varias chamadas à função **read,** utilizando o file descriptor originado pela segunda conexão como fonte da leitura.

* 1. **– Resultados do download.**

Caso o programa tenha conseguido executar sem problemas o terminal deve indicar todas as respostas obtidas do servidor tal como algumas mensagens da aplicação que servem como forma de seguir a execução do programa e saber que passo está a executar.

