2º Trabalho Laboratorial

Mestrado Integrado em Engenharia Informática e Computação

Redes de Computadores

Grupo:

Cláudia Margarida da Rocha Marinho - up201404493

Rui Miguel Oliveira - up201000619

Vinicius Ferretti da Silva – up201600721

Dezembro de 2016

Sumário

Neste relatório será apresentado o resultado obtido do desenvolvimento de uma aplicação de download e um sumário das várias experiências efetuadas ao longo das aulas práticas da unidade curricular de Redes de Computadores do Mestrado Integrado em Engenharia Informática e Computação.

O nosso trabalho foi terminado com sucesso, já que foi possível configurar uma rede e realizar a transferência de ficheiros sem qualquer tipo de erros, usando a aplicação desenvolvida.

Introdução

Este trabalho teve como objetivo desenvolver e implementar uma aplicação capaz de realizar download de um ficheiro através usando o *File Transfer Protocol (FTP)*. Também foi efetuada a configuração de uma rede local, configurando o router, switch, criando rotas e fazendo teste (ping) de cada aplicação, para isso foi utilizado o software Wireshark que captura os pings de resposta.

Assim, ao longo deste relatório vamo-nos focar em duas componentes essencialmente: a configuração de uma rede e o desenvolvimento de uma aplicação de download. Em relação à configuração de uma rede, vamos descrever as várias experiências efetuadas ao longo das aulas práticas seguindo o guião, explicando-as. Quanto à aplicação de download, vamos descrever a sua estrutura e aspetos mais gerais da aplicação.

Parte 1 - Aplicação de Download

1.1 Arquitectura

A aplicação está essencialmente dividida em duas partes: processamento da string dada como argumento ao utilizador e a implementação do protocolo FTP propriamente dito. Como a primeira parte é muito menos trabalhosa que a segunda, então apenas as funções relacionadas com a segunda parte (implementação do protocolo FTP) foram postas num outro bloco chamado “FTP.c”. O resto das funções estão em “main.c “.

As informações obtidas a partir do argumento são guardadas numa struct chamada url com os campos *user*, *password*, *host*, *url_path*, *filename* e *ip*. Todos estes campos exceto o último podem ser obtidos através da string dada como argumento que tem de ser do tipo ***ftp://[<user>:<password>@]<host>/<url-path>***. O campo user vai conter o nome *<user>*, *password* vai conter *<password>*, *host* vai conter *<host>* e *<url-path>* vai ser divido pelos campos *url_path* e *filename*. Isto foi feito de forma a tornar possível a mudança do diretório para *url_path* e a obtenção do ficheiro com o nome em *filename* através de retr (como vai ser explicado mais detalhadamente à frente). O ip vai ser obtido através da função gethostbyname que vai ter o host como argumento.

```
typedef struct URL {
        char user[MAX_SIZE];
        char password[MAX_SIZE];
        char host[MAX_SIZE];
        char url_path[MAX_SIZE];
        char filename[MAX_SIZE];
        char ip[MAX_SIZE];
} url;
```

Tendo o ip, torna-se possível a iniciação do protocolo FTP. Começa-se então por conectar a um servidor através de um socket TCP usando a função *connect_to_server(char * ip, int port)* cujo código foi fornecido no moodle. A porta usada na ligação de controlo foi a porta 21. Após a tentativa de estabelecimento de ligação, é recebida uma mensagem que que permite-nos testar se a conexão foi bem sucedida ou não.

Quando a ligação estiver estabelecida, faz-se login do utilizador com o user e password fornecidos anteriormente. Para efetuar login, o programa chama a função *login(int sockfd, char * user, char * password)* em que são mandadas e recebidas mensagens para o host. Para enviar e receber mensagens são usadas as funções *send_msg(int sockfd, char * msg)* e *receive_msg(int sockfd, char * msg)*. É verificado se o código recebido no fim da tentativa de autenticação está correto ou nao.

Posteriormente, é enviado o comando para que o servidor transfira os dados em modo passivo, sendo chamada a função *pasv(int sockfd)*. Nesta função, a mensagem “pasv\n” é enviada e a resposta recebida deve conter dados que permitem obter o ip e a porta da nova conexão a ser estabelecida. Após a obtenção do ip e da porta (que tem de

ser calculada), estabelece-se uma nova conexão com *connect_to_server*. Esta ligação vai ser usada para transferir o ficheiro pretendido.

Depois do estabelecimento da nova conexão de dados, deve-se mudar o diretório para a pasta em que se encontra o ficheiro a ser transferido através de *cwd_path(int sockfd, char * path)*. Com o diretório na pasta certa basta só pedir ao servidor para enviar um ficheiro através da conexão de dados estabelecida, chamado a função *retr(int sockfd, char * file)*. Esta função vai ser bem-sucedida se o ficheiro existir no diretório especificado.

Por último, o ficheiro vai ser transferido, sendo aberto um ficheiro para escrita e escrevendo neste os dados lidos de *datafd* a cada 256 bytes. No fim da transferência, este ficheiro é fechado e é recebida uma mensagem do servidor a indicar se o ficheiro foi transferido com sucesso. Por fim, as duas conexões estabelecidas também são fechadas e termina aqui a implementação do protocolo FTP.

1.2 Resultados

Para efeitos de debug e para o utilizador ter percepção das trocas de mensagens entre o programa e o servidor, todas as mensagens enviadas e recebidas são imprimidas na consola. Assim, no caso da transferência de ficheiro ter sido bem sucedida, o seguinte deve ser mostrado no ecrã:

```
Information given:
User name: anonymous
User password: abcd
Host: ftp.up.pt
URL path: /pub/
Filename: robots.txt
IP: 193.136.37.8

220 Bem-vindo à Universidade do Porto
user anonymous
331 Please specify the password.
pass abcd
230 Login successful.
pasv
227 Entering Passive Mode (193,136,37,8,181,161)

cwd /pub/
250 Directory successfully changed.
retr robots.txt
150 Opening BINARY mode data connection for robots.txt (23 bytes).
226 File send OK.
```

Para obter estes resultados, foram efetuados os comandos "*make*" e "*./main ftp://[anonymous:abcd@]ftp.up.pt/pub/robots.txt*". É possível verificar no diretório em que se encontra a aplicação que o ficheiro robots.txt foi transferido com sucesso após a execução destes comandos.

Parte 2 - Configuração de Rede e análise de tráfego

Part 2.1 Configurar um IP de rede

Nesta experiência configuramos o IP da porta eth0 do tux1 e tux4 utilizando o comando ***<ifconfig [porta] [ip]/[bits]>*** (ver IP's no Anexo 1). Depois, do tux1 pingamos o tux 4 com ***<ping [ip]>*** e verificámos os pacotes ARP a associar um IP a um MAC address, assim como o ping a enviar pacotes de ICMP. Conseguimos identificar o campo Ethertype da trama que identifica o tamanho da trama e o protocolo do pacote (ver anexo - RCOM - wireshark - relatório\Tux51\Part 1).

Part 2.2 Implementar duas LAN's virtuais num Switch

Nesta etapa foram implementadas as configurações da vlan50 e vlan 51, utilizando os comandos ***<configure terminal> <vlany0> <end>*** (ver anexo - RCOM - wireshark - relatório\Tux53\part 2).

Sendo que existem dois broadcast domains, um para vlan configurada (vlan 50, vlan 51), assim pode-se observar nos logs que quando se faz o ***<ping -b 172,16.xy.255>***, só é possível obter resposta dos tux ligados à rede corresponde.

Part 2.3 Configurar Router no Linux

Nesta fase, configuramos o tux4 de forma a servir como router entre os tux1 e tux2. Para tal, ligamos a porta eth1 com o comando ***<ifconfig eth1 up>*** e atribuímos-lhe um IP pertencente à vlan 51 (ver Anexo 1). A seguir, habilitamos o IP forwarding e desabilitamos o ICMP, para o tux4 poder reenviar os pacotes que recebe de uma vlan para a outra. Depois, no tux1 e tux 2, criámos uma rota para cada de forma a poderem comunicar entre si com o comando ***<route add -net [ip vlan destino] gw [ip tux de ligaçao]>*** (ver anexos - RCOM - wireshark realtorio\tux5y\part 3).

Por fim, do tux1 pingamos o tux2 e tux4 para testar se a configuração foi feita com sucesso. Reparámos na troca de pacotes ARP para associação de IP's e endereços MAC, foram capturados pacotes de ICMP com pedidos de respostas de *echos* entre os vários tux.

Part 2.4 Configurar Router comercial e implementar NAT

Neste passo foi configurado o router comercial e o NAT conforme descrito no guião. Primeiramente para o teste, foram efetuados pings com o router configurado para analisar se obtinha resposta do 172.16.1.254 a partir do tux1, sendo que não foi possível e posteriormente foram efetuados os mesmos procedimentos com o NAT configurado para comparar e analizar as respostas, sendo foi possível pingar um IP externo (www.google.com) conforme anexo.

Part 2.5 DNS

Nesta instância, configurou-se o DNS de forma a ser possível aceder a redes externas editando o ficheiro resolv.conf (ver anexo - RCOM - wireshark realtorio\tux51\part

6). Este ficheiro é lido cada vez que são invocadas rotinas que dão acesso à Internet. Para testar pingou-se o servidor da google (*www.google.com*), e verificou-se que são trocados pacotes de DNS, TCP e FTP durante a execução da aplicação (ver anexo - RCOM - wireshark realtorio\tux51\part 6).

Part 2.6 Ligações TCP

Por fim, nesta experiência foi compilado a aplicação desenvolvida, conforme descrição anteriormente e utilizando um servidor FTP foi efetuado o download de um ficheiro. O download efectuou-se com sucesso, o que demonstrou que a rede foi bem configurado, não demonstrando problemas no acesso por protocolo ftp, assim como a utilização de um servidor externo.

TCP utiliza *Selective Repeat* ARQ, com a funcionalidade de o receptor não deixar de processar os frames recebidos quando detecta um erro. Quando é detectado uma falha no grama o receptor continua um *acknowledgement* com o número de frames que falhou e continua a processar as seguintes, enviando no ack, o número da frame que falhou primeiro.

Conclusão

Com a conclusão do segundo trabalho laboratorial pode-se concluir que o desenvolvimento da aplicação de download e as configurações do router e switch foram efetuadas com sucesso, permitindo um melhor entendimento sobre como funcionam as transferências por FTP e o tráfego na rede, suas rotas, seu funcionamento, permitindo uma melhor “visualização” do que acontece com ficheiros transportados pela mesma rede.

Referências

Anexos

Anexo 1

```
TUX51:
MAC addrr: 00:0f:fe:8b:e4:a7
IP addrr: 172.16.50.1

TUX52
MAC addrr: 00:21:5a:61:2f:d6
IP addrr: 172.16.51.1

TUX54:
Eth0
MAC addrr: 00:21:5a:c3:78:70
IP addrr: 172.16.50.254 -- eth0
Eth1
MAC addrr: 00:60:df:08:d5:b0
IP addrr: 172.16.51.253 -- eth1
```