Redes de Computadores

*2º Trabalho Laboratorial*

FEUP-LEIC

*(20 de dezembro de 2022)*

Diogo Almeida – 202006059

Rafael Morgado – 201506449

Sumário

Este trabalho foi desenvolvido no âmbito da cadeira de Redes de Computadores na FEUP, que consistiu na elaboração de uma configuração de uma rede de computadores e execução de uma aplicação também esta desenvolvida por nós, com o objetivo de fazer download de um ficheiro.

Introdução

Este projeto foi desenvolvido ao longo de diversas aulas laboratoriais, estando ele dividido em duas partes principais, o cliente FTP e uma configuração de rede de computadores, num primeiro passo em trabalho autónomo, foi desenvolvido um cliente simples baseado no protocolo FTP e ligação TCP usando *sockets*, o qual se ligaria posteriormente aos servidores da FEUP para fazer download de um ficheiro.

Para esta componente tivemos de perceber oque é um cliente, um servidor e as suas particularidades em TCP/IP, saber como se caracterizam protocolos em aplicações no geral, como definir um URL, descrever o comportamento de um servidor FTP e perceber como o DNS será usado dentro da aplicação.

Presencialmente, nos laboratórios da FEUP, tivemos de configurar uma rede de computadores de acordo com o guião proposto nas aulas, que consistia em seis experiências com objetivos destintos, mas resultados acumulativos. O principal objetivo destas experiências foi a configuração de uma rede que permitisse a execução da aplicação previamente desenvolvida, através de duas LANS, ligadas por *bridges* dentro de um *switch*, numa das quais esta configurado um NAT, tendo a outra de se ligar à internet através dessa para a aplicação FTP funcionar.

Este relatório está subdividido em duas partes principais juntamente com a conclusão e segue a seguinte estrutura:

* Aplicação de download, onde é descrita a arquitetura da aplicação e demonstração dos resultados obtidos, junto com a sua análise.
* Configuração da rede de computadores, descrição de cada experiência, junto com os comandos mais pertinentes que foram usados para chegar ao resultado da configuração e análise dos *logs* que foram feitos nos laboratórios.
* Conclusões, resumo da informação apresentada nas secções anteriores e reflexão sobre os objetivos de aprendizagem alcançados.

Aplicação FTP

O primeiro componente deste projeto consiste no desenvolvimento de um cliente de *download* que siga o protocolo FTP com ligações TCP a partir de *sockets*, sendo então possível fazer a transferência de qualquer ficheiro de um servidor FTP. Recebe como argumento um link no formato *ftp://<user>:<password>@<host>/<url-path>,* escrito em C, que ira fazer o *download* de um ficheiro na última experiência na parte de configuração da rede.

Arquitetura

A nossa aplicação está toda contida num único ficheiro C, onde decorre toda a lógica do programa, ele está dividido em 3 partes principais sequenciais, Argumentos, Conexão e Download.

É criada uma *struct* *hostent* no início do código, logo de seguida na secção dos Argumentos é processado o URL que é recebido como argumento quando se corre a aplicação, eles são guardados para um *buffer* com a seguinte estrutura:

* Utilizador
* *Password*
* *Host*
* *Urlpath*

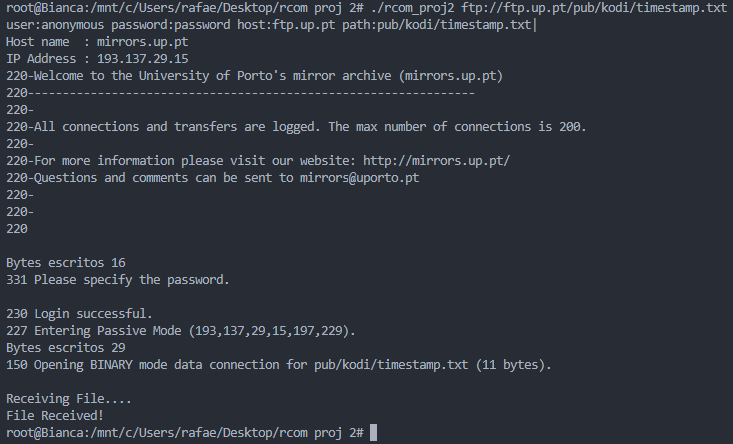
De seguida a função *gethostbymane()* é usada de forma a descobrir o endereço IP do *Host* e o nome do *Host*.

Na secção da conexão, é aberta uma *socket* para se conseguir fazer a primeira conexão, se seguida a ligação ao *server* é feita e são enviados as *strings* com o *user* e a *password*, não havendo erros, o cliente passa a modo passivo e volta a fazer uma conexão ao servidor ficando depois à espera de receber o ficheiro para *download*.

Na secção de *Download*, depois de ser ter ligado ao server pela segunda vez, se o ficheiro já existir, fecha o *stream* com erro, caso contrário, cria o ficheiro e começa a receber as tramas de informação até o ficheiro ser completamente descarregado. Por fim as duas *sockets* são fechadas.

Resultados

Independentemente de terem sido introduzidos as credenciais de *login* ou não, o nosso cliente funcionou sem erros, no caso de não serem introduzidos, a nossa aplicação assume um utilizador *anonymous* e uma password, *password*. Foi também possível fazer *download* de vários tipos diferentes de ficheiros com tamanhos diferentes sem gerar erros.



Configuração da rede de computadores

[Nota] As imagens representativas das capturas efetuadas durante as experiências laboratoriais, tanto como os comandos exatos usados, estão no diretório Anexo no ficheiro zip.

Experiência 1 – Configurar uma rede IP

O objetivo desta primeira tarefa foi a configuração de IP’s numa rede entre dois computadores utilizando um *switch* de forma que estes consigam comunicar entre si. Após ligarmos as portas *eth0* dos computadores TUX3 e TUX4 a um *switch,* foram utilizados os comandos *ifconfig* e *route* para criar essa ligação.

Foi feito um *ping* do TUX3 para o TUX4, que gera pacotes ICMP, estes pacotes são usados para mandar mensagens de controlo na camada 3, neste contexto são usados para solicitar pedidos de resposta ao IP alvo de forma a aferir se a ligação está bem estabelecida.

Os pacotes ARP são usados para, sabendo o IP de uma máquina, conseguirmos obter o seu MAC *address*, isto porque os MAC *addresses* são identificadores das próprias placas de rede, ao invés do IP que é o endereço publico de uma máquina. Uma máquina só tem um MAC *address*, mas pode ter vários IP’s.

Após se ter confirmado a correta configuração, apagamos as entradas da tabela ARP no TUX3 e usamos o software *wireshark* para capturar o tráfego que foi enviado do TUX3 para o TUX4 quando voltamos a correr o comando *ping*, foi então possível verificar o formato dos pacotes ARP juntamente com os pacotes ICMP que foram enviados durante o *ping.* Cada tipo de pacote tem uma cor diferente no *wireshark*. Para averiguar o tamanho de cada trama verificamos o valor em bytes passados *on wire.*

A interface *loopback* é uma interface de rede virtual que o computador utiliza para comunicar com sigo próprio, com o objetivo de realizar testes de diagnóstico ou aceder a servidores na própria máquina, como se fosse um cliente.

Experiência 2 – Implementar duas *bridges* num *switch*

Nesta experiência foram criadas duas *bridges* isoladas no *switch*, no nosso caso a *bridge* 40 e a 41, na primeira estão o TUX3 e TUX4, na segunda está o TUX2, como estão em bridges diferentes ficam isoladas uma da outra, ou seja, o TUX2 não consegue comunicar com a bridge 40.

Primeiro começámos por ligar as portas *eth0* dos 3 terminais ao *switch,* de seguida no *gtkterm* estando ele ligado a porta de série do *switch*, removemos todas as *bridges* que já pudessem existir no *switch,* de forma a não nos causar problemas, criamos então as duas *bridges* com os comandos */interface bridge port add bridge=bridge<bridge-number> interface=<port-number>* para ambas as *bridges*, alterando apenas os nomes das *bridges* e as portas que lhes correspondem.

Após terem sido criadas as *bridges*, corremos o comando *ping* do TUX3, que está na *bridge* 40, para o TUX2, que esta na *bridge* 41, como seria de esperar, estando as duas bridges isoladas uma da outra, os *pings* falharam. Foi também feito o mesmo para o IP 172.16.40.255, que não obteve nenhuma resposta, apesar de se encontra na mesma sub-rede que o TUX3, isto acontece porque *echo-ignore-broadcas* está ligado por omissão para prevenir amplificações de tráfico.

Um comando de *ping* no TUX2 para o IP 172.16.41.255 foi feito com mesmo resultado que no TUX3, mas desta vez a razão é que não existem mais terminais na *bridge* 41.

Com isto podemos concluir que existem dois domínios de *Broadcast* que correspondem as sub-redes da *bridge* 40 e *bridge* 41 com os endereços de 172.16.40.255 e 172.16.41.255, respetivamente.

Experiência 3 – configurar um router em Linux

Nesta experiência o objetivo é transformar o TUX4 num *router* de forma a possibilitar a comunicação entre os TUX2 e o TUX3.

Começamos por ligar a porta *eth1* do TUX4 ao *switch,* e configuramos o seu IP de forma que esteja integrado na sub-rede do TUX2, ou seja na *bridge* 41.

De seguida reconfiguramos os TUX2 e TUX3 de forma que conseguissem comunicar entre si utilizando o TUX4 como *default gateway* para o TUX3, criando uma *route* no TUX3 que desse acesso ao IP 172.16.41.0/24 a partir do IP 172.16.40.254/24 (TUX4 *eth0*) e outra *route* no TUX2 que desse acesso ao IP 172.16.40.0/24 a partir do IP 172.16.41.253/24 (TUX4 *eth1*), foi também desligado o *echo-ignore-broadcast,* estas rotas podem ser confirmadas na *forwarding table* de cada um usando o comando *route -n.*

Assim feito, estando estas rotas definidas, com o comando *ping* no TUX3, todas as interfaces dos terminais da rede iram retornar *acknowledgements.* Utilizando o *wireshark* podemos verificar que a porta *eth0* do TUX4 envia dois pedidos ARP para averiguar os MAC *addresses* do TUX3 e que o TUX3 mandou outro pedido ARP para averiguar o MAC *address* da porta *eth0* do TUX4, o mesmo acontece simetricamente entre o TUX2 e o TUX4 em que as portas descobertas entre si serão o *eth0* do TUX2 pelo TUX4 e a porta *eth1* do TUX4 pelo TUX2. É então possível verificar que existe comunicação entre o TUX2 e o TUX3 tendo em consideração que os *pings* feitos têm resposta,

Experiência 4 – Configurar *Router* comercial e implementação do NAT

Nesta experiência é nos pretendido fazer uma ligação da rede que já tínhamos estabelecido até agora, à rede dos laboratórios via um *router*, adicionando-lhe rotas, e posteriormente uma configuração NAT.

O NAT, quando implementado num *router,* substitui os endereços IP em comunicações na rede por um endereço IP público. Com este mecanismo implementado, o router onde o NAT estiver configurado, passa a ser responsável por reencaminhar todo o tráfego que por ele passe para o endereço da rede privada correto.

Começando pela configuração física, ligamos a porta *eth1* do *router* à rede do laboratório, com o NAT ligado por omissão, a porta *eth2* foi ligada à bridge 41 de forma a estar na mesma sub-rede que o TUX2, com o IP 172.16.1.49/24. Utilizando a mesma configuração de rede que já se tinha feito na última experiência, em que o TUX4 servia de *router* para o TUX3, acrescentamos o *router* Rc como *default gateway* dentro da bridge 41 para o TUX2 e o TUX4.

Estando esta configuração feita, foi possível usar o w*ireshark* e o comando *ping* com sucesso no TUX3 para averiguar que as ligações estavam devidamente feitas. Uma vez que a comunicação entre o TUX3 e o TUX2 é efetuada através do TUX4 pela rota definida no TUX2, usando os comandos *echo 0 > /proc/sys/net/ipv4/conf/eth0/accept\_redirects, echo 0 > /proc/sys/net/ipv4/conf/all/accept\_redirects*  e de seguida apagando essa rota, ao usarmos o comando *traceroute tux43* no TUX2, verificamos que não havendo nenhuma rota definida para a sub-rede 40, o *router* Rc com o IP 172.16.21.254 que tinha sido configurado como *default gateway* do TUX2 passou a ser responsável por redirecionar o tráfego até ao seu destino pretendido. Quando estabelecido de novo a rota que utiliza o TUX4 para comunicar com o TUX3, podemos verificar, utilizando novamente o comando *traceroute,* que o tráfego deixa de passar pelo *router* Rc, passando ele agora pelo TUX4, pois este seria a rota mais eficiente.

No passo seguinte utilizamos o comando *ping* do TUX3 para o *router* do laboratório, com o IP 172.16.1.254 sem sucesso, visto que o NAT ainda não tinha sido configurado no *router*, aquando desta configuração, corremos o mesmo comando, desta vez com sucesso, visto que agora já era possível no NAT, que houvesse tráfego entre a rede local e a rede externa.

Experiência 5 – Configurar um DNS

Nesta experiência, o objetivo passa por configurar um serviço DNS nos TUX2, TUX3, TUX4 e também verificar que pacotes são trocados pelo DNS e qual a informação que contêm.

A configuração do DNS para qualquer um dos terminais fica concluída quando for introduzido uma linha de texto contendo *nameserver <ip-address>* no ficheiro *etc/resolv.conf.*

Ao utilizarmos o *wireshark* e o comando *ping,* serão enviados dois pacotes DNS com dois pedidos destintos, *Address Maping Record*, de forma a receber o endereço IPV4 do *host* e IP *Version 6 Address Record*, que visa receber o endereço IPV6. Ambos recebem resposta, para o qual, *ftp.up.pt,* o endereço IPV4 é 193.137.29.15 e o endereço IPV6 é 2001:690:2200:1200::15. De seguida, quando recebido uma resposta bem-sucedida do *ping,* é enviado outro pedido DNS, desta vez do tipo *Reverse-Lookup Pointer Record*, que utilizando o IP obtido anteriormente ira agora encontrar todos os seus *host names,* que neste caso seria *mirror.up.pt.*

Experiência 6 – conexão TCP

Nesta experiência, teremos de, utilizando toda a configuração de rede que foi feita até agora e utilizando a aplicação que foi previamente elaborado por nós, e explicada anteriormente neste relatório, conseguir fazer o download de um ficheiro no TUX3 e posteriormente, fazer o download de um ficheiro no TUX3 e de outro ficheiro noTUX2 ao mesmo tempo.

Quando executada a aplicação, duas conexões TCP são abertas, uma quando se entra em contacto com o servidor, e pela qual é feita o controlo da transferência de dados, e a outra que será utilizada para fazer toda a transferência de dados, inicialmente para dar início a conexão ao servidor, é aberta apenas uma porta, esta à qual, será usada para fazer a própria conexão.

A conexão foi feita no modo passivo, assim foi, pois, nos permite identificar a porta de retorno de maneira mais direta e desta forma iniciar uma nova ligação para a transferência binaria. No encalço da criação da nossa ligação, recebesse então o FTP-DATA, que indica o tamanho em bytes de cada pacote, no fim de cada um desses pacotes, é enviada uma mensagem de *acknowledgement* pelo cliente ao servidor com a finalidade de confirmar a receção de dados, quando chegar ao fim do ficheiro, é recebida uma mensagem com o código 226, de sucesso, vinda do servidor, para confirmar o fim da transmissão de dados.

O ARQ TCP funciona a partir do método da janela deslizante, consistindo ele, dessa forma, no controlo de erros durante a transmissão de dados, ao enviar pacotes de controlo *acknowledgements* numerados, a partir do cliente. O campo *sequence-number* identifica o *byte* que delimita o início da mensagem no pacote de dados, e o ACK *number field* identifica o número de sequência do próximo pacote que o emissor deve enviar para o cliente.

No controlo de congestionamento, as mensagens de *acknowledgement* são utilizadas pelo cliente, ao indicar em cada uma delas o tamanho da janela TCP, consiste este então, na quantidade de *bytes* que podem ser recebidos pelo cliente. Na eventualidade de o cliente não conseguir ler os dados de forma devidamente célere, esta janela pode ficar cheia, neste caso o cliente envia uma mensagem de *acknowledgement* indicando que a *bit window full está* ativada, aí, o recetor para de enviar dados temporariamente, no caso de não parar, os *bytes* excedentes são descartados, oque leva a uma quantidade elevada de retransmissões desnecessárias. No momento em que o recetor acabe de receber todos os *bytes* do pacote TCP, ele envia um TCP *window update,* com a indicação que esta pronto a receber mais dados.

Quando se iniciou os *downloads* de ficheiros no TUX2 e TUX3 ao mesmo tempo, era espectável que o *throughput* das ligações fossem menores do que quando existe só uma ligação, de forma a que a taxa de transferência seja distribuída de igual forma por ambos os terminais, apesar disto, nesta situação, não conseguimos verificar uma diminuição significativa do *throughput*, da mesma forma, quando um dos downloadsacabou estando o outro ainda a decorrer, não foi possível ver uma mudança notória do *throughput*, em que o espectável seria um restabelecimento do valor do *throughput* previamente verificado durante o *download* singular no TUX3.

Conclusão

Sentimos que todos os objetivos propostos neste trabalho foram concluídos com sucesso, desde a implementação de um cliente simples FTP à configuração de uma rede de computadores.

Na primeira parte deste trabalho, na implementação de um cliente FTP, apesar de ser mais simples do que no primeiro trabalho laboratorial desta cadeira, foi nos possível obter um aprofundamento dos conhecimentos sobre os protocolos FTP e TCP e a correta utilização de *sockets* e URL’s.

No que toca à configuração da rede de computadores, o facto de ter sido incremental foi nos bastante benéfico, já que nos permitiu perceber de uma forma mais granular todo o seu funcionamento, com isso foi possível laborar as nossas competências de arquitetura de redes e configuração dos dispositivos que as compõem, como os *switches* e *routers*, *configurações* NAT e DNS, estruturas de dados como as *forwarding* *tables*, ARP *tables*, e também dos protocolos que as usam, ICMP, ARP.

Com tudo isto podemos concluir que este trabalho foi bastante enriquecedor ás nossas competências na área, não só a um nível profissional, mas também a um mais pessoal.

Referências

Para este trabalho, foram utilizados os slides das aulas tanto práticas como teóricas, juntamente com algumas pesquisas espontâneas no google sobre alguns comandos a usar e protocolos.