Relatório do 2º Trabalho Laboratorial

Mestrado Integrado em Engenharia Informática e Computação

Redes de Computadores

Grupo:

Carolina Azevedo – up201506509
Nuno Oliveira – up201506487
Pedro Miranda – up201506574

Faculdade de Engenharia da Universidade do Porto

Rua Roberto Frias, sn, 4200-465 Porto, Portugal

Índice

Sumário

Este trabalho laboratorial está dividido em duas grandes parte. A primeira é implementar uma aplicação download com um cliente FTP (File Transfer Protocol) para fazer transferência de ficheiros, sendo a segunda, configurar e analisar uma rede. Estas experiências basearam-se na configuração de um IP de rede, de um router em Linux, de um router comercial e do DNS (Domain Name System), e na implementação de duas LAN's (Local Area Network) virtuais no switch e do NAT (Network Address Translation) e num teste com a aplicação de download desenvolvida para a verificação de um bom funcionamento nas ligações TCP (Transmission Control Protocol).

Aplicação Download

1. Arquitectura

A aplicação está dividida em duas partes, numa primeira parte iremos processar a string de input do utilizador e em seguida, iremos focar-nos na ligação com o servidor, login e download do ficheiro indicado pelo utilizador. Utilizamos uma estrutura chamada *url_info* onde guardamos toda a informação disponibilizada pelo o utilizador na string de input, ou seja, o username, a palavra-passe, o host, file-path e nome do ficheiro que o utilizador quer fazer download. Guardamos ainda a struct hostent que contém as informações sobre host.

```
13  typedef struct {
14    char user[256];
15    char password[256];
16    char host_url[256];
17    struct hostent* host_info;
18    char file_path[256];
19    char filename[256];
20  } url_info ;
```

Figura 1:Estrutura url

Ao correr o programa, primeiro é validado o input do utilizador. Após, a validação da string de input, é chamada a função *parse_url* que retira da string de input os vários dados necessário.

```
int parse_url(char url[], url_info* url_info);
```

Figura 2: Função parse_url

Após todas as variáveis estarem instanciadas, é chamada a função init_connection responsável por ligar o cliente FTP ao servidor através de um socket.

```
int init_connection(char* address, int p);
```

Figura 3: Função init_connection

Depois de conectados, o cliente irá fazer login no servidor com o username e a palavra passe disponibilizados pelo utilizador. Para isso, utiliza a função *login*. Função que envia dois comandos ao servidor, o comando ”USER ’user’\r\n”e ”PASS ’password’\r\n”aguardando pela respetiva resposta do servidor.

```
void login(int ctr_socket, url_info* url_info);
```

Figura 4:Função login

Após efetuado o login, iremos colocar o servidor em modo passivo, para o fazer chamamos a função *passive_mode* que envia o comando ”PASV \r\n”. Desta forma o cliente inicia ambas as conexões ao servidor, resolvendo o problema de as firewalls filtrarem os dados recebidos pelo cliente do servidor. O servidor responde, enviando uma string, com o endereço ip e a porta onde iremos, mais tarde fazer o download do ficheiro. Após, a receção da mensagem desmontamos a string e guardamos o ip e a porta. Com este endereço e porta chamamos novamente a função *init_connection*, que retorna o socket que mais tarde iremos utilizar para fazer download do ficheiro.

```
void passive_mode(int sockfd, char* ip, int* p);
```

Figura 5: Função passive_mode

Após estabelecida a nova conexão, iremos agora pedir ao servidor o ficheiro desejado, para isso utilizando a função *send_retrieve*, que envia o comando ”RETR ’file_path’\r\n”. Após enviar o comando, lemos a resposta do servidor e verificamos se o ficheiro existe.Após o pedido do ficheiro, chamamos a função *download* que lê o ficheiro pedido pelo o utilizador e o guarda em disco.

```
void send_retrieve(int ctr_socket, url_info* url_info);

int download(int data_socket, url_info* url_info);
```

Figura 6:Funções send_retrieve e download

Terminada a receção do ficheiro fecham-se os sockets para terminar o programa com a função *end_connection*.

```
int end_connection(int ctr_socket, int data_socket);
```

Figura 7: Função end_connection

2. Resultados

Esta aplicação foi testada com diversos ficheiros, tanto em modo anónimo como em modo não anónimo. Em caso de erro, para além da aplicação terminar é impresso na consola o erro em causa, de modo a que o utilizador tenha a informação sobre o que correu mal.

Durante o decorrer da aplicação vão sendo impressos na consola os códigos de retorno do servidor FTP, bem como outras mensagens de carácter informativo sobre o estado da operação.

Configuração Da Rede

1. Experiência 1: Configurar um IP de rede

A primeira experiência tem como objetivo configurar dois computadores numa só rede e estabelecer assim uma comunicação entre estes. Para isso foram configurados, com o comando *ifconfig*, os dois computadores tux51 e tux54 para que estes assumissem os endereços de IP de 172.16.50.1 e 172.16.50.254, respectivamente.

A chamada do comando Ping gera pacote ICMP. O comando Ping usa os pacotes ICMP para transferir mensagens de controlo entre endereços IP. Analisando o log do wireshark, podemos verificar que, o pacote ARP envia uma mensagem broadcast a todos os computadores com o objetivo de saber qual o MAC address correspondente a um certo IP.

MAC é o hardware address, único, da placa de rede. ARP é o protocolo que atribui a um IP um MAC. Os ARP Packets possuem informação de qual é o MAC address para um determinado IP.

A interface loopback é uma interface de rede virtual que o computador utiliza para comunicar com ele próprio, com o objectivo de realizar testes de diagnóstico, ou aceder a servidores na própria máquina, como se fosse um cliente. Assim, uma interface loopback, permite a existência de um endereço IP no router, que está sempre activo, em vez de ser dependente de uma interface física.

2. Experiência 2: Implementar duas LAN's virtuais no switch

Nesta experiência foram criadas duas LANs virtuais no switch: a primeira constituída pelos computadores tux51 e tux54, e a segunda pelo computador tux52. Com esta configuração, o computador 2 deixaria de ter acesso aos computadores tux51 e tux54, uma vez que se encontrariam em sub-redes diferentes.

Para configurar a vlan50 foi necessário através da consola do switch definir a vlan através do comando <*vlan 50*> e depois configurar as suas duas portas de forma a ligar uma ao tux51 e outra ao tux54.

De seguida, foi enviado o comando Ping do tux51 para o tux54 e depois para o tux52. De seguida foi também chamado o comando Ping Broadcast a partir do tux52.

Através do log do wireshark podemos concluir que existem duas sub-redes diferentes, logo, dois broadcasts diferentes. O comando Ping brodcast envia um ping para todos os computadores estao ligados a essa rede. Ou seja se o tux51 mandar um Ping brodcast, apenas o tux54 o vai receber pois estão ambos ligados a vlan50. Contudo o tux52 não o recebe pois esta ligado a uma rede diferente, neste caso a vlan51.

Desta forma, podemos concluir que existem dois domínios de Broadcast, vlan50 e vlan51.

3. Experiência 3: Configurar um router Linux

Esta experiência consiste na configuração do computador tux54 como router por forma a ligar as duas vlan's existentes, a Vlan 50: 172.16.50.0/24 e a Vlan 51: 172.16.51.0/24.

Para isto, foi ativada a porta eth1 do tux54. Além disso, foi ligada ao switch e, de seguida, à vlan51. Configura-se esta mesma porta com o endereço IP 172.16.51.253/24.

Após esta configuração, adicionou-se uma rota ao tux 51 utilizando o comando <route add –net 172.16.51.0/24 gw 172.16.50.254>. O primeiro endereço identifica a gama de endereços para a qual se quer adicionar a rota; o segundo endereço identifica o IP para o qual se deve reencaminhar o pacote (neste caso o IP do tux 54). Posteriormente, repetiu-se o mesmo procedimento para a máquina 2, mas utilizando os seguintes endereços <route add –net 172.16.50.0/24 gw 172.16.51.253>. Novamente, o IP 172.16.51.253,é o IP do tux 54 nesta sub-rede.

Podemos verificar que agora o tux51 e o tux52 podem comunicar através do tux54.

Uma entrada na tabela de encaminhamento tem uma NetMask e um Network Destination, para descrever o ID da rede, uma Gateway, uma Interface responsável por alcançar a gateway e um Metric.

Os pacotes ARP que se identificam são entre a interface eth0 do tux54 e o tux51, e entre a interface eth1 do tux54 e o tux52. Os endereços MAC são do tux54 para cada uma das interfaces e os respetivos para o tux51 ou tux52. Este pacote ARP serve para informar ao tux51 ou tux52, dependendo de quem está a tentar comunicar, que o endereço IP destino de que está à procura é acessível pela máquina tux54(endereço MAC), segundo a rota estabelecida.

Através dos logs, percebe-se que o pacote ICMP contém como endereço de destino o endereço MAC do tux54 e na resposta do tux52 o pacote contém também como endereço de origem o endereço MAC do tux54, o que era previsto, visto que este é que faz o redirecionamento da comunicação entre as duas VLANs.

4. Experiência 4: Configurar um router comercial e Implementar NAT

Nesta experiência pretendia-se a configuração de um router comercial com NAT devidamente implementado.

Na medida em que os IP's públicos são um recurso limitado e atualmente escasso, o NAT tem como objetivo poupar o espaço de endereçamento público, recorrendo a IP's privados. Os endereços públicos são pagos e permitem identificar univocamente uma máquina (PC, routers,etc) na Internet. Por outro lado os endereços privados apenas fazem sentido num domínio local e não são conhecidos na Internet, sendo que uma máquina configurada com um IP privado terá de sair para a Internet através de um IP público.

Essa tradução de um endereço privado num endereço público é conseguida através do NAT.

Configurou-se o router definindo as rotas internas e externas com o comando ip route na consola de configuração do router.

Desta forma, definiu-se o computador tux54 como default gateway do computador tux51 e o router como default gateway dos computadores tux52 e tux54. Assim, os pacotes enviados pelo computador tux51 seguem para o computador tux54 e depois para o router ou para o computador tux52.

Para testar, foi executado no tux51 um ping ao router da sala e verificou-se que os pacotes enviados pelo tux51, passavam pelo tux54, onde eram reencaminhados para o router no IP 172.16.51.254.

5. Experiência 5: DNS

Esta experiência consiste em adicionar um DNS aos tuxes.

O DNS é um sistema de gerenciamento de nomes hierárquico e distribuído para computadores, serviços ou qualquer recurso conectado à Internet ou numa rede privada. Funciona como um sistema de tradução de endereços IP em nomes de domínios. Para isso, editou-se o ficheiro "resolv.conf" situado no diretório /etc nos computadores com o sistema operativo em Linux, de modo a que fosse semelhante ao seguinte:

```
1  # Generated by NetworkManager
2  search netlab.fe.up.pt fe.up.pt
3  nameserver 172.16.1.1
4  nameserver 193.136.28.10
5
```

Figura 8:Configuração DNS

Para testar esta experiência, foi feito o teste de ping usando www.google.pt. Nos logs, consequentemente, verificou-se que o DNS pergunta a informação contida num dado domain name, e este responde com o tempo de vida e o tamanho do pacote de dados.

6. Experiência 6: Ligações TCP

Na experiência 6, compilou-se e executou-se a aplicação desenvolvida e descrita na primeira parte do relatório. O teste foi feito com recurso à transferência de um ficheiro através de um servidor FTP. A transferência foi bem-sucedida, mostrando que a configuração da rede foi feita sem erros.A aplicação FTP abre 4 ligações sendo as duas primeiras para envio de comandas e recessão de respostas e os dois últimos para receber dados. A informação de controlo FTP é enviada pela primeira ligação FTP. O Transmission Control Protocol (TCP) utiliza o mecanismo Automatic Repeat Request (ARQ). Este método consiste no controlo de erros na transmissão de dados. Para isso utiliza ack (mensagens enviadas pelo receptor indicando que a trama de dados foi recebida corretamente) e timeouts (tempo permitido para esperar por um acknowledgment), de forma a garantir uma transmissão confiável através do serviço não confiável. Se não for recebido um ack antes do timeout, a trama é retransmitida até ser recebido um ack. Para fazer o controlo de congestão, o TCP mantém uma janela de congestão que consiste numa estimativa do número de octetos que a rede consegue encaminhar, não enviando mais octetos do que o mínimo da janela definida pelo recetor e pela janela de congestão. A transferência de dados em simultâneo pode levar a uma queda na taxa de transmissão, uma vez que a taxa de transferência é distribuída de igual forma para cada ligação.

Como podemos ver no gráfico de quando fizemos um download enquanto outro decorria, a taxa de transmissão cai no momento em que o segundo download é iniciado :

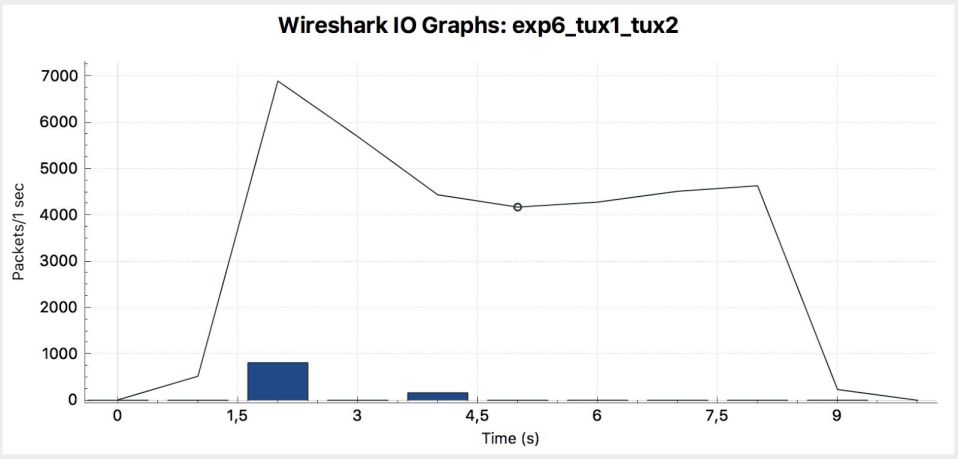

Figura 9: Download em simultâneo no tux 51 e tux 52

7. Experiência 7: Implementar NAT em Linux

Por fim na experiência 7 implementamos NAT no tux 54, através dos seguintes comandos aplicados na configuração deste:

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
iptables -A FORWARD -i eth1 -m state --state NEW,INVALID -j DROP
iptables -L
iptables -t nat -L
```

Figura 10: Configuração de NAT no tux 54

Com este procedimento, é aplicado NAT ao tux 54 que fará com que o tux 51 consiga pingar o tux 52 mas o reverso não seja possível, ou seja, o tux 52 não consegue pingar o tux 51 que serve para demonstrar a funcionalidade do NAT de servir como uma “firewall” não permitindo que passem para o tux 51 packets que não foram requisitados por este.

Conclusões

Após a conclusão do segundo projeto de Redes e Computadores, o grupo interiorizou grande parte dos conceitos necessários para uma implementação estável e coerente do que era pedido no guião. A implementação do cliente FTP (File Transfer Protocol) foi concluída com sucesso, o cliente era capaz se fazer download de diferentes tipos de ficheiros e tamanhos diferentes. Com esta implementação, o grupo entendeu melhor o funcionamento deste protocolo. Relativamente à configuração de rede, também foi concluída com sucesso, permitindo, aos elementos do grupo perceber, a um nível aprofundado, como funciona a configuração de uma rede, algo usado no dia-a-dia. Foi ainda concluída a experiência 7, experiência opcional, que permitiu uma melhor percepção do funcionamento do NAT. Portanto, podemos concluir que os objetivos delineados pelo guião do segundo projeto, foram atingidos com sucesso.