

Trabalho Laboratorial 2

Relatório Final

Redes de Computadores

FEUP - 3º Ano

3MIEIC05 - GRUPO 206

Iohan Xavier Sardinha Dutra Soares - up201801011

Isla Patrícia Loureiro Cassamo - up201808549

Introdução	2
Parte 1 - Aplicação de Download	2
Parte 2	3
Configuração de uma rede IP	3
Implementação de duas Lan's virtuais em um switch	4
Configuração de um router em Linux	5
Configuração de um router comercial e implementação de NAT	6
Configuração de DNS	7
Conexões TCP	7
Conclusão	8
Referências	8

Introdução

Este relatório é relativo ao segundo trabalho laboratorial da disciplina Redes de Computadores o qual possui como objetivo estudar e compreender como funcionam os protocolos e configuração de uma rede de computadores.

O trabalho foi dividido em duas partes: na primeira foi desenvolvida uma aplicação em C para download de ficheiros com o uso do protocolo FTP, já a segunda parte constituía da configuração completa de uma rede desde da configuração de IP's a conexões FTP.

Ambas as partes foram desenvolvidas como entendido, cumprindo os objetivos do trabalho. A aplicação de download consegue realizar downloads com sucesso e a configuração de rede feita nos laboratórios da universidade foi realizada com sucesso, sendo feitas todas as medições necessárias e ajudando a entender as questões propostas.

Parte 1 - Aplicação de Download

A aplicação de download foi desenvolvida em C seguindo uma arquitetura simples. Dois ficheiros .c que possuem a função principal e as funções secundárias chamadas por ela, e três ficheiros .h com as definições necessárias para os dois ficheiros e algumas definições comuns em *utils.h*.

Primeiramente é realizado o processamento da *URL* que é passada como parâmetro obrigatória ao abrir a aplicação, fazendo a identificação do nome de usuário, palavra passe, caminho para o ficheiro e nome do ficheiro. Esta *URL* deve estar no formato '*ftp://[<user>:<password>@]<host>/<url-path>*' onde *user* e *password* são opcionais. Caso nem o campo *user*, nem *password* sejam dados pelo usuário, estes são substituídos por *anonymous*, ambos. Caso esteja faltando apenas um dos dois campos, o presente será interpretado como *password* e o *user* será *anonymous*.

Uma vez processada a *URL* será aberta uma conexão *FTP* usando a função *socket* por onde serão enviados os comandos de login e de ativação do modo passivo. Para cada comando enviado é lida e verificada sua resposta, é feita a interpretação da resposta recebida para saber se o código enviado é o esperado. Após entrar no modo passivo, é feito o cálculo do porto e ip respondidos ao entrar neste modo, então é aberto um novo *socket* para receber os dados, este ao qual é feito o comando *retr*. Os dados lidos são escritos em um ficheiro com o nome do ficheiro. Quando a transferência é finalizada todas as conexões feitas são fechadas.

Se em algum momento ocorrer algum erro abrindo uma conexão ou ficheiro, ou alguma resposta inesperada o programa interrompe sua execução retornando um código de erro e imprimindo o problema na tela.

```

~/Documentos/FEUP/RCOM/RCOM/LAB2 $ ls
download download.c download.h logs socket.c socket.h utils.h
~/Documentos/FEUP/RCOM/RCOM/LAB2 $ ./download ftp://ftp.up.pt/welcome.msg
User: anonymous
Password: anonymous
Host: ftp.up.pt
Path: welcome.msg
File name: welcome.msg
Host name: mirrors.up.pt
IP address: 193.137.29.15
< 220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)
220-----
220-
220-All connections and transfers are logged. The max number of connections is 200.
220-
220-For more information please visit our website: http://mirrors.up.pt/
220-Questions and comments can be sent to mirrors@uporto.pt
220-
220-
220-
> user anonymous
< 331 Please specify the password.

> pass anonymous
< 230 Login successful.

> pasv
< 227 Entering Passive Mode (193,137,29,15,210,210).

> retr welcome.msg
~/Documentos/FEUP/RCOM/RCOM/LAB2 $ ls
download download.c download.h logs socket.c socket.h utils.h welcome.msg
~/Documentos/FEUP/RCOM/RCOM/LAB2 $ cat welcome.msg
Welcome, archive user! This is an experimental FTP server. If have any
unusual problems, please report them via e-mail to Joao.Carvalho@fe.up.pt
If you do have problems, please try using a dash (-) as the first character
of your password -- this will turn off the continuation messages that may
be confusing your ftp client.
~/Documentos/FEUP/RCOM/RCOM/LAB2 $ █

```

Demonstração do download bem sucedido de um ficheiro

Parte 2

Configuração de uma rede IP

A conexão para esta experiência foi feita diretamente pela conexão entre os dois computadores *gnu43* e *gnu44* através de cabos de rede, sendo desligado o switch que normalmente os conecta ao laboratório. Então é feita a configuração do IP dos dois, testada a sua conexão e registrados os dados da conexão.

ARP é o protocolo usado para descobrir os endereços da camada de rede de uma rede, tais quais como o IP e MAC.

IP é um endereço que identifica uma máquina em uma rede, utilizado dentro de uma rede para localizar cada computador individualmente.

Assim como o IP, MAC também é um endereço de identificação, mas por sua vez está associado a um controlador de rede, aos quais uma máquina pode ter mais de um. O endereço MAC tem como objetivo identificar uma máquina fisicamente, um identificador único, enquanto o IP pretende identificar uma conexão em uma rede.

Uma vez configurados os IPs de uma rede pode ser feito o comando *ping* para testar sua conexão, este comando funciona fazendo pedidos a partir de uma máquina a outra, se comunicando através do protocolo ICMP, utilizado por dispositivos de rede para enviar mensagens de erro e informações operacionais, através do qual no ping são enviados os endereços IP e MAC da máquina que faz o pedido, e são recebidos os endereços da máquina que devolverá a resposta.

Quando um frame de Ethernet é recebido este possui um header no qual está contida a informação de que tipo de frame é esse, em um número hexadecimal, assim é determinado se este frame é um IP, ARP ou ICMP. Como o frame pode estar em diferentes formatos e tamanhos em seu header está associado o seu comprimento, de diferentes maneiras dependendo de qual protocolo está sendo usado.

A interface de loopback é um método utilizado principalmente para testar a comunicação em um canal de uma rede, onde é enviada uma mensagem que será devolvida sem qualquer tipo de alteração ou processamento, assim comprovando que a conexão está estabelecida e estável.

Principais comandos de configuração para este experimento:

```
$ ifconfig
$ ping
```

Envio de pedido por parte do gnu43 e resposta do gnu44:

5	7.669584090	HewlettP_01:2f:98	Broadcast	ARP	42 Who has 172.16.40.254? Tell 172.16.40.1
6	7.669721537	HewlettP_c3:78:76	HewlettP_61:2f:98	ARP	60 172.16.40.254 is at 00:21:5a:c3:78:76
7	7.669738368	172.16.40.1	172.16.40.254	ICMP	98 Echo (ping) request id=0x0c0b, seq=1/256, ttl=64 (reply in 8)
8	7.669873659	172.16.40.254	172.16.40.1	ICMP	98 Echo (ping) reply id=0x0c0b, seq=1/256, ttl=64 (request in...
9	8.019652791	Cisco_7c:8f:81	Spanning-tree-(for-...	STP	60 Conf. Root = 32768/1/00:1e:14:7c:8f:80 Cost = 0 Port = 0x80...
10	8.136944969	Cisco_7c:8f:81	Cisco_7c:8f:81	LOOP	60 Reply
11	8.688612385	172.16.40.1	172.16.40.254	ICMP	98 Echo (ping) request id=0x0c0b, seq=2/512, ttl=64 (reply in 1...
12	8.688744804	172.16.40.254	172.16.40.1	ICMP	98 Echo (ping) reply id=0x0c0b, seq=2/512, ttl=64 (request in...
13	9.712611816	172.16.40.1	172.16.40.254	ICMP	98 Echo (ping) request id=0x0c0b, seq=3/768, ttl=64 (reply in 1...
14	9.712775243	172.16.40.254	172.16.40.1	ICMP	98 Echo (ping) reply id=0x0c0b, seq=3/768, ttl=64 (request in...
15	10.024441236	Cisco_7c:8f:81	Spanning-tree-(for-...	STP	60 Conf. Root = 32768/1/00:1e:14:7c:8f:80 Cost = 0 Port = 0x80...
16	10.736608173	172.16.40.1	172.16.40.254	ICMP	98 Echo (ping) request id=0x0c0b, seq=4/1024, ttl=64 (reply in ...
17	10.736742058	172.16.40.254	172.16.40.1	ICMP	98 Echo (ping) reply id=0x0c0b, seq=4/1024, ttl=64 (request i...
18	11.764609206	172.16.40.1	172.16.40.254	ICMP	98 Echo (ping) request id=0x0c0b, seq=5/1280, ttl=64 (reply in ...
19	11.764742113	172.16.40.254	172.16.40.1	ICMP	98 Echo (ping) reply id=0x0c0b, seq=5/1280, ttl=64 (request i...

Implementação de duas Vlan's virtuais em um switch

As Vlans são pequenas áreas locais de internet que neste caso se instalam no Switch mas podendo ser também instaladas em routers.

Instalá-las implica simplesmente invocar o comando "**vlan yx**" dentro das configurações do terminal do switch. É importante notar que de seguida é preciso adicionar portas às vlans, de forma a representarem a rede pretendida. Na totalidade os seguintes comandos são de maior relevância:

- >>vlan yx : usado para configurar a vlan yx.
- >>interface fastethernet 0/x : x sendo a porta onde o gnu pretendido está ligado ao switch.
- >>switchport mode access.
- >>switchport access vlan xy : conecta a vlan a porta.

O broadcast abrange apenas as portas da mesma Vlan, e por isso existem apenas 2 domínios. Consegue-se concluir isto através dos logs, pois ao executar os comandos ping ou broadcast no gnu43 ele atinge apenas o gnu44, dando erro se for tentado o ping para o gnu42, e se os comandos forem executados no gnu44 ele chega aos dois outros gnu's.

Configuração de um router em Linux

Há 3 gnu's na instalação deste projeto. O gnu44 foi o escolhido para ser configurado como router. Para tal, as duas Vlans criadas na experiência anterior são conectadas a ele e rotas através dele são instaladas usando o comando "**route add -net 'destination' gw 'gateway'**". Isso quer dizer que para além das rotas default que chegam a qualquer lugar através do router 0.0.0.0 também, do tux3 chega-se a rede 172.16.41.0 e do tux2 chega-se a rede 172.16.40.0 através do tux4 usando a instalação.

Em cada entrada da tabela de forwarding encontra-se a seguinte informação:

- Destination: Endereço IP do gnu de destino.
- Gateway: Endereço IP da máquina usada como "middleman" entre o remetente e o destinatário.
- Interface: Placa da rede (eth0).
- Netmask: a máscara dos IP's.
- Flags: contém informação sobre as rotas.
- Metric: custo da transmissão por essa rota.
- Reference: referência da rota.
- Use: contador de pesquisas, podendo ser o contador de sucessos ou falhas dependendo da flag usada.

A cada transferência, como o emissor desconhece o endereço MAC do destinatário, é enviada uma mensagem ARP pelo emissor com o seu MAC e o campo para o MAC do receptor vazio, para o qual o receptor recebe e responde preenchendo com o seu respectivo MAC, sendo que no final, a mensagem ARP terá associada o endereço MAC de destino e de origem.

Observam-se durante as transferências pacotes ICMP do tipo **request e reply** pois as rotas foram adicionadas e como tal os gnu's se reconhecem. Estes pacotes têm associados os endereços IP e MAC dos gnus de origem e destino. Todavia, se este não fosse o caso, seriam enviados pacotes do tipo **Host Unreachable**.

No.	Time	Source	Destination	Protocol	Length	Info
53	84.204835949	Cisco_7c:8f:87	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/41/00:1e:14:7c:8f:80 Cost = 0 Port = 0x8...
54	84.850708231	Cisco_7c:8f:87	Cisco_7c:8f:87	LOOP	60	Reply
55	86.289576837	Cisco_7c:8f:87	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/41/00:1e:14:7c:8f:80 Cost = 0 Port = 0x8...
56	87.083369149	Kye_02:55:95	Broadcast	ARP	42	Who has 172.16.41.1? Tell 172.16.41.253
57	87.083471676	HewlettP_61:2f:10	Kye_02:55:95	ARP	60	172.16.41.1 is at 00:21:5a:61:2f:10
58	87.083488438	172.16.40.1	172.16.41.1	ICMP	98	Echo (ping) request id=0x0d90, seq=1/256, ttl=63 (reply in 5...
59	87.083605352	172.16.41.1	172.16.40.1	ICMP	98	Echo (ping) reply id=0x0d90, seq=1/256, ttl=64 (request in...
60	88.094553799	172.16.40.1	172.16.41.1	ICMP	98	Echo (ping) request id=0x0d90, seq=2/512, ttl=63 (reply in 6...
61	88.094661983	172.16.41.1	172.16.40.1	ICMP	98	Echo (ping) reply id=0x0d90, seq=2/512, ttl=64 (request in...
62	88.219856609	Cisco_7c:8f:87	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/41/00:1e:14:7c:8f:80 Cost = 0 Port = 0x8...
63	89.118589791	172.16.40.1	172.16.41.1	ICMP	98	Echo (ping) request id=0x0d90, seq=3/768, ttl=63 (reply in 6...
64	89.118700559	172.16.41.1	172.16.40.1	ICMP	98	Echo (ping) reply id=0x0d90, seq=3/768, ttl=64 (request in...
65	90.142590373	172.16.40.1	172.16.41.1	ICMP	98	Echo (ping) request id=0x0d90, seq=4/1024, ttl=63 (reply in ...
66	90.142698278	172.16.41.1	172.16.40.1	ICMP	98	Echo (ping) reply id=0x0d90, seq=4/1024, ttl=64 (request i...
67	90.219321494	Cisco_7c:8f:87	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/41/00:1e:14:7c:8f:80 Cost = 0 Port = 0x8...
68	91.166603597	172.16.40.1	172.16.41.1	ICMP	98	Echo (ping) request id=0x0d90, seq=5/1280, ttl=63 (reply in ...
69	91.166736784	172.16.41.1	172.16.40.1	ICMP	98	Echo (ping) reply id=0x0d90, seq=5/1280, ttl=64 (request i...
70	92.168271124	HewlettP_61:2f:10	Kye_02:55:95	ARP	60	Who has 172.16.41.253? Tell 172.16.41.1
71	92.168291797	Kye_02:55:95	HewlettP_61:2f:10	ARP	42	172.16.41.253 is at 00:c0:df:02:55:95
72	92.190620662	172.16.40.1	172.16.41.1	ICMP	98	Echo (ping) request id=0x0d90, seq=6/1536, ttl=63 (reply in ...
73	92.190725354	172.16.41.1	172.16.40.1	ICMP	98	Echo (ping) reply id=0x0d90, seq=6/1536, ttl=64 (request i...
74	92.224232515	Cisco_7c:8f:87	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/41/00:1e:14:7c:8f:80 Cost = 0 Port = 0x8...
75	94.229197883	Cisco_7c:8f:87	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/41/00:1e:14:7c:8f:80 Cost = 0 Port = 0x8...
76	94.858396751	Cisco_7c:8f:87	Cisco_7c:8f:87	LOOP	60	Reply

Para configurar um router desta maneira usam-se os comandos **ifconfig** e **route** já usados na primeira experiência, apenas de uma maneira diferente que permite a conexão entre as duas Vlans configuradas.

Configuração de um router comercial e implementação de NAT

Esta experiência envolve configurar um router comercial da Cisco e posteriormente configurar a NAT (Network Address Translation) seguindo as instruções do slide 46 das instruções fornecidas com algumas pequenas alterações de interface e endereços.

O NAT permite traduzir IP's locais em IP's da rede pública, permitindo troca de informações entre computadores de redes públicas e privadas. De uma forma simples, permite que computadores dentro de uma rede privada acessem uma rede pública.

Para configurar uma rota estática no router, entra-se no configure terminal do router e usar o comando "**ip route destino gw gateway**". Os pacotes transmitidos, quando estas rotas estão configuradas, seguem esses caminhos. Caso não estejam, os pacotes seguirão a rota default através dos router.

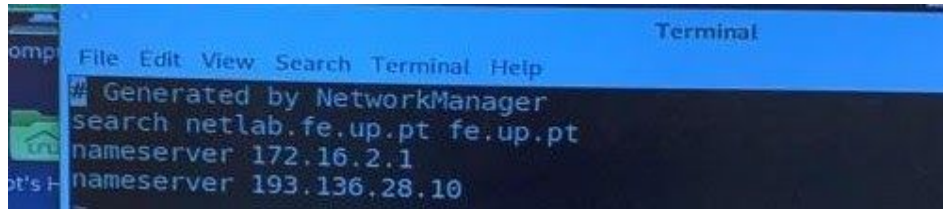
```

r1 min/avg/max/mdev = 1.130/1.146/1.165/0.033 ms
root@gnu42:~# traceroute 172.16.40.1
traceroute to 172.16.40.1 (172.16.40.1), 30 hops max, 60 byte packets
 1 172.16.41.254 (172.16.41.254) 1.225 ms 1.502 ms 1.825 ms
 2 172.16.41.253 (172.16.41.253) 2.999 ms 0.393 ms 0.388 ms
 3 172.16.40.1 (172.16.40.1) 0.595 ms 0.584 ms 0.570 ms
root@gnu42:~# route add -net 172.16.40.0/24 gw 172.16.41.253
root@gnu42:~# traceroute 172.16.40.1
traceroute to 172.16.40.1 (172.16.40.1), 30 hops max, 60 byte packets
 1 172.16.41.253 (172.16.41.253) 0.186 ms 0.164 ms 0.145 ms
 2 172.16.40.1 (172.16.40.1) 0.351 ms 0.334 ms 0.315 ms
root@gnu42:~#

```


Configuração de DNS

Um sistema de Nomes e Domínios (DNS) é um sistema responsável por localizar e traduzir para IP 's os nomes. Para configurá-los é necessário aceder ao ficheiro **/etc/resolv.conf** e colocar a informação do servidor (IP e nome). Neste projeto o exemplo concreto seria colocar **netlab1.fe.up.pt(nome)** e **nameserver 172.16.2.1(endereço)**.



O processo usado pelo DNS consiste em o host enviar um pacote com o hostname para o servidor e em retorno recebe de volta o IP correspondente do servidor.

Conexões TCP

Para a realização deste experimento não foram utilizados comandos específicos da linha de comando, apenas a aplicação desenvolvida na parte 1 do trabalho laboratorial e estudado seu comportamento.

O protocolo TCP pode ser dividido em três fases: estabelecimento da conexão, transferência de dados e terminação de conexão. Quando a aplicação de download FTP é utilizada são abertas duas conexões TCP, uma na qual se faz o login e se entra no modo anônimo, ou seja: as informações de controle e outra onde se faz a transferência dos dados.

Durante a fase de transferência de dados é usado o protocolo ARQ TCP de controle de erros, onde são enviados os dados e se aguarda por uma confirmação ou time-out, caso haja time-out o pacote de dados é reenviado, para garantir a confiança no canal de rede. No pacote de transferência do TCP são de destacável relevância os campos: porto remetente e destinatário, número de sequência e número de acknowledgment, usado para retransmitir os dados caso não cheguem corretamente. Nos logs de uma comunicação TCP é possível identificar os campos do header de acknowledgment, sequência, window size, e comprimento.

Para evitar a queda da conexão por excesso de congestionamento, o protocolo TCP usa vários mecanismos de controle de congestionamento, para controlar a taxa de transmissão de dados e evitar o congestionamento. Como aumentar o tempo entre as retransmissões não bem sucedidas, feito através do controle da janela de congestionamento. Quando uma

segunda conexão aparece é perceptível a diferença entre os tempos devido ao congestionamento.

Conclusão

O segundo trabalho de RCOM consistia em configurar uma rede IP , desenvolver uma aplicação de download por protocolos FTP e usá-las mutuamente para efetuar um download utilizando o citado protocolo.

Ao longo deste processo desenvolveram-se as habilidades necessárias para configurar devidamente uma rede de computadores, desde a camada mais simples à mais abstrata. Da configuração física através de cabos à digital através dos comandos em linux. Assim como se aprofundaram os conhecimentos sobre os protocolos de rede e seu funcionamento, nomeadamente ARP, ICMP, FTP e TCP, através de uma análise prática e de dados coletados.

Sendo estes os objetivos, é de afirmar que os tais foram concluídos com sucesso e que com tal a material teórico da cadeira foi aprofundada e a cadeira melhor percebida. Em suma, a realização do projeto foi positiva para os elementos do grupo.

Referências

- Slides das aulas teóricas
- https://en.wikipedia.org/wiki/Address_Resolution_Protocol
- [https://en.wikipedia.org/wiki/Ping_\(networking_utility\)](https://en.wikipedia.org/wiki/Ping_(networking_utility))
- https://en.wikipedia.org/wiki/MAC_address
- https://en.wikipedia.org/wiki/File_Transfer_Protocol
- https://en.wikipedia.org/wiki/Automatic_repeat_request
- https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
- Andrew Tanenbaum, David Wetherall, Computer Networks, 5/E, Prentice Hall, 2011