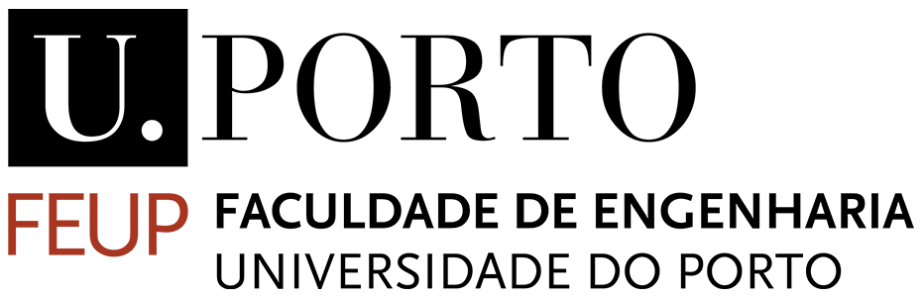
**2º Trabalho Laboratorial**

Relatório



Mestrado Integrado em Engenharia Informática e Computação

Redes de Computadores

**Trabalho realizado por:**

Maria Gonçalves Caldeira (up201704507)

Raul Manuel Fidalgo da Silva Teixeira Viana (up201208089)

18 de dezembro de 2020

Índice

[Sumário 3](#_Toc58059559)

[Introdução 3](#_Toc58059560)

[Parte 1: Aplicação de download 3](#_Toc58059561)

[Arquitetura 3](#_Toc58059562)

[Resultados 4](#_Toc58059563)

[Parte 2: Configuração da Rede 4](#_Toc58059564)

[Conclusões 4](#_Toc58059565)

# **Sumário**

Este relatório foi elaborado no âmbito da unidade curricular de Redes e Computadores, do curso Mestrado Integrado em Engenharia Informática e Computação, para descrever o segundo trabalho prático, que consistiu no desenvolvimento de uma aplicação capaz de realizar o *download* de um ficheiro através da utilização do protocolo FTP (*File Transfer Protocol*) e na montagem e configuração de uma rede privada.

Assim, é possível afirmar que o trabalho foi concluído com sucesso, visto que os objetivos estabelecidos foram cumpridos, tendo sido configurada uma rede e realizado o *download* de um ficheiro através desta.

# **Introdução**

Os objetivos deste trabalho foram o desenvolvimento de uma aplicação de *download* e a configuração de uma rede. Esta rede irá permitir o funcionamento correto da aplicação a partir da criação de duas VLAN’s dentro de um *switch*.

De forma a descrever o trabalho laboratorial realizado este relatório está divido na seguinte estrutura:

**Parte 1**: Aplicação de download: arquitetura e resultados;

**Parte 2**: configuração e análise da rede: análise e descrição de cada experiência;

**Conclusão**: resumo da informação apresentada nas secções anteriores e conclusões finais.

# **Parte 1: Aplicação de download**

A primeira parte deste trabalho consistiu no desenvolvimento de uma aplicação de *download*, escrita em C. Esta aplicação aceita como argumento um *link* do tipo:

ftp://<username>:<password>@<host>/<url-path>

Através do processamento deste argumento a aplicação será capaz de qualquer ficheiro que esteja alojado num servidor FTP.

## **Arquitetura**

A aplicação está dividida em dois ficheiros principais: *main.c* e *handlers.c.* No segundo estão definidas as funções necessárias à manipulação dos *url’s*, dos *sockets* e da ligação FTP.

O processamento é iniciado pelo *parse* do *url*. É criada a estrutura “url” que contém espaço para as variáveis *user, password*, *host*, *path*, *filename* e *ip\_adress*. Seguidamente é feito o *parsing* do argumento recebido de forma a preencher estas variáveis. É corrida também a função get\_ip() que converte o nome do *host* num endereço de *ip*. A porta utilizada é a 21.

Posteriormente é criado um *socket* através do qual serão transferidos os comandos e lidos as respostas do servidor. É feito o *login* enviando os comandos “USER” e “PASS” e posteriormente é pedido ao servidor que entre em modo passivo através do envio do comando “PASV”. A resposta do servidor a este comando é interpretada pela função ftp\_passive\_mode() que calcula o endereço de ip e a porta através dos quais se liga um novo *socket*, desta feita para a transferência do ficheiro. Assim é enviado o comando “RETR” através do primeiro *socket* e recebido o ficheiro através do segundo.

Por fim são fechadas as conexões e o ficheiro criado.

## Resultados

  A aplicação foi testada com ficheiros de diferentes tamanhos e tipos. É apresentado o estado do processamento na consola para mais fácil controlo por parte do utilizador.

# **Parte 2: Configuração da Rede**

## Experiência 1 - Configurar um endereço de IP de rede

O objetivo desta experiência foi ligar o *tuxy3* ao *tuxy4* utilizando um *switch* e adquirir conhecimento necessário para responder às seguintes perguntas:

1. **O que são pacotes ARP?**

O protocolo *Address Resolution Protocol* (ARP) é um protocolo de comunicação utilizado para um determinado computador descobrir o endereço da camada de ligação associado ao endereço de IP. Serve para mapear o endereço de rede a um endereço físico, por exemplo um endereço MAC.

1. **Quais são os endereços MAC e IP dos pacotes ARP e porquê?**

Ao usar o comando ***ifconfig*** no *tuxy3*, pudemos verificar que o seu IP tinha sido bem configurado, tendo ele sido configurado como *172.16.40.1*, e que o seu endereço MAC era *00:21:5a:61:2f:98*. No *tuxy4* o procedimento foi o mesmo, e o seu IP foi configurado como *172.16.40.254* e o seu endereço MAC era *00:21:5a:c3:78:76*.

1. Quais pacotes são gerados pelo comando *ping*?

O comando *ping* é utlizado para testar a conectividade entre o tuxy3 e o tuxy4 e gera tanto pacotes ARP (onde obtém o endereço MAC) como pacotes ICMP.

1. Quais são os endereços MAC e IP dos pacotes *ping*?

Os endereços de origem e destino dos pacotes vão ser os designados na tabela seguinte:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MAC | | IP | |
|  | Origem | Destino | Origem | Destino |
| Pacote Pedido | *00:21:5a:61:2f:98* | *00:21:5a:c3:78:76* | 172.16.40.1 | 172.16.40.254 |
| Pacote Resposta | 00:21:5a:c3:78:76 | 00:21:5a:61:2f:98 | 172.16.40.254 | 172.16.40.1 |

Tabela 1: endereços de origem e destino dos pacotes ARP no *ping*

1. Como determinar se a trama Ethernet recebida é do tripo ARP, IP ou ICMP?

É possível obter esta informação inspecionando o cabeçalho de uma trama Ethernet. Se o valor for 0x0800, representa um pacote do tipo IP, sendo que neste caso também é possível analisar o IP header. Se este tomar o valor de 1, quer dizer que se trata de um protocolo do tipo ICMP. Contudo, se o valor for 0x806, representa um pacote do tipo ARP.

1. Como determinar o comprimento de uma trama recebida?

Através da utilização do *Wireshark* é possível inspecionar a trama e observar o seu comprimento. Consultar imagem …(Arranjar imagem dos logs)

1. O que é a *interface loopback* e qual a sua importância?

A *interface loopback* é uma *interface* de rede virtual que permite que o computador comunique com ele mesmo. É um mecanismo utilizado para testar a correta configuração da rede, permitindo a existência de um IP sempre ativo no router, o que descarta a dependência numa interface física.

## **Experiência 2 – Implementação de duas LAN’s virtuais no switch**

Nesta experiência foram criadas duas LAN’s virtuais, *vlany0* e *vlany1*. Os computadores *tuxy3* e *tuxy4* foram adicionados à primeira, enquanto que o computador *tuxy2* foi adicionado à segunda.

1. **Como configurar *vlanyo*?**

A configuração física da *vlany0* passa por realizar as ligações corretas. Na régua 1 a porta *cisco->RS232* terá que ser ligada à porta do *switch* na régua 2. O *tux* que se pretende que esteja ligado ao *switch* tem de ter a sua porta S0 à porta *RS232->cisco* da régua 1. De seguida são introduzidos os seguintes comandos no *gtkterm* do *tux* a configurar:

* *conf t* (ou *configure terminal*)
* *vlan y0*
* *end*
* *show vlan brief* (para verificar se a vlan foi criada)

Adicionar portas (porta do tuxy3 e do tuxy4):

* *conf t* (ou *configure terminal*)
* *interface fastethernet 0/[nº da porta]*
* *switchport mode access*
* *switch access vlan y0*
* *end*

1. **Quantos domínios de transmissão existem?**

Nesta configuração existem dois domínios de transmissão. Quando os *tuxy3 e tuxy4* fazem *ping broadcast* apenas recebem resposta um do outro e não do *tuxy2*. O *tuxy3* recebe do *tuxy4* e vice-versa. Por outro lado, quando o *tuxy2* faz *ping broadcast* não recebe qualquer resposta. Assim é possível afirmar que existem dois domínios de transmissão e ainda que os *tuxy3 e tuxy4* pertencem a um e o *tuxy2* pertence a outro. Figura??(Arranjar imagem dos logs)

## **Experiência 3 – Configurar um router em Linux**

1. **Que rotas existem nos *tuxes*? Qual o seu significado?**

* No *tuxy3* há uma rota para a vlan y0 (*172.16.y0.0*) – *gateway 172.16.y0.1.* Ao longo da experiência foi criada outra rota para a vlan y1 (*172.16.y1.0*) - *gateway 172.16.y0.254.*
* No *tuxy4* há uma rota para a vlan y0 (*172.16.y0.0*) – *gateway 172.16.y0.254;* e outra rota para a vlan y1 (*172.16.y1.0*) – *gateway 172.16.y1.253.*
* No *tuxy2* há uma rota para a vlan y1 (*172.16.y1.0*) – *gateway 172.16.y1.1.* Ao longo da experiência foi criada outra rota para a vlan y0 (*172.16.y0.0*) - *gateway 172.16.y1.253.*

O destino de cada rota corresponde ao alcance de cada uma delas.

1. **Que informação é que uma entrada de uma tabela de forwarding contém?**

Estas tabelas são obtidas através do comando *“route -n”* e contém informação sobre o destino da rota (***Destination***), o ip do ponto em que a rota vai passar (***Gateway***), a máscara (***Netmask***), as ***Flags***, o “custo” de cada rota (***Metric***), o número de referências para a rota (***Ref***), o contador de pesquisas pelas rotas (***Use***) e a ***Interface***.

1. **Que mensagens ARP e endereços MAC associados são observados e porquê?**

Utilizando o comando *ping* é possível analisar a sequência existente entre mensagens ARP e endereços MAC. Caso um tux dê *ping* a outro tux e este segundo não reconhecer o endereço MAC do primeiro, irá enviar lhe uma mensagem ARP a perguntar-lhe qual é o endereço MAC que corresponde àquele ip. Esta mensagem terá o MAC do tux de origem e como o tux de destino ainda não é conhecido, terá o endereço *00:00:00:00:00:00.* Logo a seguir, é enviada outra mensagem ARP agora a partir do tux de destino com o seu endereço MAC e, como é obvio, esta mensagem também terá associado o MAC do tux de origem. Figura??(Arranjar imagem dos logs)

1. **Que pacotes ICMP são observados e porquê?**

São observados pacotes ICMP de *requests* e *replies* porque foram adicionadas durante a experiência rotas que permitem todos os tux’s conseguirem alcançar todos os outros.

1. **Quais os endereços IP e MAC associados a um pacote ICMP e porquê?**

Associados a um pacote ICMP estão os endereços IP e MAC do tux de origem e do de destino. Estes são atribuídos ao ICMP quando o tux de destino é alcançável e se o MAC estiver mapeado. Figura??(Arranjar imagem dos logs)

# **Conclusões**

O tema deste trabalho é o protocolo de ligação de dados, que consiste em fornecer um serviço de comunicação de dados fiável entre dois sistemas ligados por um meio de transmissão, neste caso, um cabo série.

Adicionalmente, foi dado a conhecer o termo **independência entre camadas**, e cada um dos blocos funcionais da arquitetura da aplicação desenvolvida, *writer* e *reader*, cumpre esta independência. Na camada de ligação de dados não é feito qualquer processamento que incida sobre o cabeçalho dos pacotes a transportar em tramas de Informação. Por outro lado, no que respeita à camada de aplicação, esta não conhece os detalhes do protocolo de ligação de dados, mas apenas a forma como este serviço é acedido.

Assim, o trabalho foi concluído com sucesso, tendo-se cumprido todos os objetivos propostos, sendo que a sua elaboração contribuiu para um aprofundamento do conhecimento teórico e prático acerca da temática de transferência de informação digital.