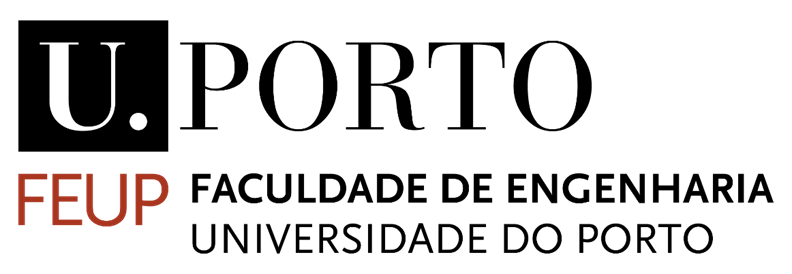
**Configuração de uma Rede**

**e Desenvolvimento de uma Aplicação de Download**

**Relatório do 2º Trabalho Laboratorial**

****

Mestrado integrado em Engenharia Informática e Computação

Redes de Computadores

Bruno Marques - 201405781

João Loureiro - 201405652

José Cruz - 201403526

José Costa - 201402717

Faculdade de Engenharia da Universidade do Porto

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

23 de dezembro de 2016

# Sumário

Este relatório foi elaborado no seguimento do segundo projeto da unidade curricular Redes de Computadores. Aqui é documentada a criação de um programa em linguagem C, que implementa o protocolo FTP (*File Transfer Protocol*) para transferir ficheiros através da Internet. É também descrita a configuração de duas redes virtuais, capazes de transferir dados entre si e com redes exteriores.

Conteúdo

[Sumário 2](#_Toc469755025)

[1. Introdução 4](#_Toc469755026)

[2. Parte 1 – Aplicação de Download 4](#_Toc469755027)

[2.1. url.c 4](#_Toc469755028)

[2.2. ftp.c 4](#_Toc469755029)

[2.3. main.c 4](#_Toc469755030)

[3. Parte 2 – Configuração de Análise da Rede 4](#_Toc469755031)

[4. Conclusões 4](#_Toc469755032)

# Introdução

O segundo projeto laboratorial de Redes de Computadores esmiúça protocolos de comunicação utilizados na comunicação entre computadores. O plano divide-se em duas partes com os seguintes objetivos:

* **Parte 1 – Aplicação de download**: recorrendo o *File Transfer Protocol* (FTP), criar uma aplicação em linguagem C capaz de transferir qualquer ficheiro de um servidor para um cliente.
* **Parte 2 – Configuração e análise de rede**: utilizando *switches* e *routers*, configurar duas redes virtuais de computadores capazes de comunicar entre si localmente (*Local Area Network –* LAN), com redes exteriores e mesmo com a Internet.

A segunda parte deste projeto segmenta-se num conjunto de seis experiências. Cada experiência indica como os computadores e as redes devem ser ajustadas e monitorizadas, com vista em responder a perguntas relativas aos protocolos que intervêm na comunicação e transmissão de dados.

# Parte 1 – Aplicação de Download

A aplicação de download divide-se em três módulos: ***main.c***, ***ftp.c*** e ***url.c***. O módulo *ftp.c* trata das funções necessárias para a interação servidor/cliente invocando comandos FTP para estabelecer a conexão e permitir a transmissão de ficheiros. O módulo *url.c* carrega numa estrutura as informações necessárias para a ligação FTP, obtidas a partir do URL que se usou como argumento. A conjunção destes dois módulos é feita em *main.c*.

## *url.c*

## *ftp.c*

## *main.c*

## Exemplo de um *download* bem-sucedido

# Parte 2 – Configuração de Análise da Rede

## Experiência 1: configurar uma rede IP

## Experiência 2: implementar duas redes virtuais num *switch*

## Experiência 3: configurar um *router* em Linux

Nesta experiência, o computador tuxy4 é programado para servir de *router*. Um *router* é um dispositivo capaz de transmitir dados entre **redes** de computadores, ao invés do *switch* que apenas maneja *packets* entre computadores pertencentes à mesma LAN.

A porta *Eth0* do tuxy1 encontra-se ligada à porta *Eth0* do tuxy4, através da rede virtual *Vlan0* criada no *switch*, pelo que a porta *Eth0* do tuxy4 é o seu *gateway* para a rede Vlan0. Sobra configurar o *gateway* do tuxy4 para a rede *Vlan1*, da qual o tuxy2 faz parte, e permitir que o tuxy4 reencaminhe *packets* IP.

Assim, as *routes* que devem existir nos computadores são as seguintes:

**Tuxy1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Source | Destination | Gateway | Genmask | Interface |
| 172.16.y0.1 | 0.0.0.0 | 172.16.y0.254 | 0.0.0.0 | *Eth0* |

**Tuxy2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Source | Destination | Gateway | Genmask | Interface |
| 172.16.y1.1 | 172.16.y0.0 | 172.16.y1.253 | 255.255.255.0 | *Eth0* |

**Tuxy4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Source | Destination | Gateway | Genmask | Interface |
| 172.16.y0.254 | 172.16.y0.0 | 0.0.0.0 | 255.255.255.0 | *Eth0* |
| 172.16.y1.253 | 172.16.y1.0 | 0.0.0.0 | 255.255.255.0 | *Eth1* |

A tabela de *forwarding* possui os dados das *routes*, permitindo escolher o melhor caminho para um pacote de informação, segundo o endereço IP do pacote.

Para tornar possível que o tuxy4 redirija pacotes IP e permita mensagens de *broadcast*, é necessário executar os seguinte comandos:

# echo 1 > /proc/sys/net/ipv4/ip\_forward

# echo 0 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_broadcasts

### Resultados da experiência

Ao iniciar a comunicação entre o tuxy1 e o tuxy2, nenhum dos computadores consegue associar o outro ao endereço IP, pois não existem entradas na tabela ARP. Para resolver o endereço IP num endereço MAC, é enviado um pacote ARP em *broadcast* perguntando qual é a máquina que tem o endereço IP definido pelas rotas. Essa máquina responde de volta com o seu endereço MAC e quem fez o pedido atualiza a sua tabela ARP.

Por exemplo, quando o tuxy1 faz *ping* ao tuxy2 através do tuxy4, ele desconhece o endereço de *layer* 2 da porya *Eth0* do tuxy4. Assim, faz um *broadcast* para a sua LAN a perguntar quem tem o IP 172.16.y0.254, que é o seu *default gateway*. O tuxy4 responde, enviando o seu endereço MAC para o IP do tuxy1. O tuxy1 adiciona este endereço às suas entradas ARP e passa a poder comunicar com o tuxy4 ao nível local.



Figura 1 - Pedido ARP feito pelo tuxy1 capturado na porta Eth0 do tuxy4

Além de pacotes ARP, são também enviados pacotes ICMP derivados do comando “ping”. Quando se executa esta ordem, o computador envia um *echo request* para tentar obter resposta de um dispositivo com um determinado endereço IP. Caso essa mensagem chegue ao destino, é enviada uma *reply* para notificar a estabilidade da conexão. Caso o *host* ou *router* não tenha sido alcançado, reporta erro.

Neste caso, o *request* é enviado pelo tuxy1, que obtém a *reply* do tuxy2.

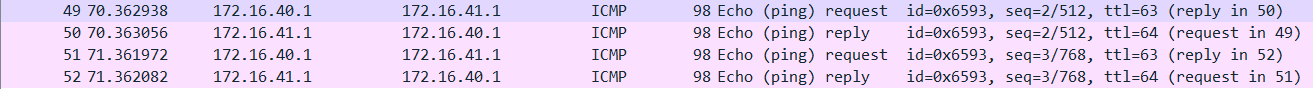


Figura 2 - Pacotes ICMP capturados na porta Eth1 do tuxy4 quando o tuxy1 "pinga" o tuxy2

(falar do IP ser igual em qualquer pacote ICMP e os MAC serem do tuxy2-eth1 e tuxy1-eth0)

## Experiência 4: configurar um *router* comercial e implementar NAT

## Experiência 5: DNS

## Experiência 6: conexões TCP

# Conclusões