A black sign with white text

Description automatically generated

**Rede de Computadores**

(2º Trabalho Laboratorial)

Redes de Computadores **·** L.EIC025

2023/2024 **·** 1º Semestre

Turma: 3LEIC08

João Correia up202005015

Miguel Charchalis up201506074

# Sumário

No âmbito da unidade curricular Redes de Computadores, realizou-se o 2º projeto que nos foi proposto. O projeto “Computer Networks” está dividido em duas partes. A primeira parte consiste em desenvolver uma aplicação de *download* de um ficheiro utilizando o protocolo FTP (*File Transfer Protocol*). A segunda parte consiste em realizar várias experiências para configurar uma rede.

O projeto foi realizado com sucesso, pois criamos uma aplicação capaz de transferir ficheiros e configuramos a rede corretamente.

# Introdução

# Através deste relatório, pretendemos apresentar de maneira organizada o funcionamento da aplicação de *download* de um ficheiro, quando fornecido um URL no formato ftp://[<user>:<password>@]<host>/<url-path> e também o funcionamento da configuração da rede de computadores capaz de se conectar à internet e testar a aplicação. Organizamos o relatório com a seguinte estrutura:

* **Aplicação de *download***

Descrição da arquitetura e do caso de sucesso.

* **Configuração e análise da rede**

Descrição da arquitetura da rede, objetivos, principais comandos de configuração e análise dos principais *logs* capturados para as seis experiências realizadas.

* Experiência 1 – Configurar uma rede IP
* Experiência 2 – Implementação de duas *bridges* no *switch*
* Experiência 3 – Configuração de um *router* em Linux
* Experiência 4 – Configurar um *router* comercial e implementar NAT
* Experiência 5 – DNS
* Experiência 6 – Conexões TCP
* **Conclusões**  
  Síntese da informação apresentada nas secções anteriores e reflexão sobre os objetivos de aprendizagem alcançados.

# Aplicação *download*

# 3.1- Arquitetura

Para realizar a execução da aplicação, foram consultadas as normas RFC959 (funcionamento e arquitetura do protocolo FTP) e RFC1738 (tratamento do URL).

Inicialmente compila-se o programa e passamos-lhe como argumento o seguinte link:

* ftp://[<user>:<password>@]<host>/<url-path>

A aplicação faz um *parse* do URL, armazenando cada componente num array. Se o *user* e a *password* não forem especificados, são assumidos como “anonymous” e “password”.

Depois de interpretados os dados do URL, é feita a comunicação com o servidor através de um *socket* TCP. O passo seguinte é o programa tentar aceder ao servidor com as credenciais indicadas e, depois o login, o programa solicita ao servidor a resposta em modo passivo (PASV), pois o servidor é quem vai escolher para que porta o ficheiro será enviado. Outro *socket* é aberto para a transferência do ficheiro. Após concluída, os dois *sockets* são fechados.

# 3.2- Caso de sucesso

# Para testar a aplicação, realizamos os testes com dois ficheiros distintos, pic1.jpg e timestamp.txt, com e sem autenticação no servidor (Figura 3.2a e 3.2b).

# 

# Figura 3.2a - *Transferência do ficheiro pic1.jpg (340603 bytes) com autenticação*

# A computer screen with white text Description automatically generated

# Figura 3.2b - *Transferência do ficheiro timestamp.txt (11 bytes) sem autenticação*

# Diagram Description automatically generatedConfiguração e análise de redes

# 4.1- Experiência 1 - Configurar uma rede IP

O objetivo desta experiência é configurar duas máquinas, Tux53 e Tux54, sendo necessário a atribuição de um endereço IP para cada uma delas.

Foi ligado a entrada E0 de cada Tux ao switch e com o comando ifconfig, configurou-se os IPs de cada um. O Tux 53 foi configurado com o IP 172.16.50.1/24, enquanto o Tux 54 foi configurado com o IP 172.16.50.254/24.

O protocolo ARP (*Address Resolution Protocol*) é um procedimento para mapear endereços de IP dinâmicos para um endereço de máquina físico numa rede local. Assim, associa o endereço IP com o endereço MAC, traduzindo endereços 32-bit para 48-bit e vice-versa, trabalhando entre a *data link layer* e a *network layer* do OSI model. Depois de apagar a tabela ARP dos computadores, o protocolo ARP foi executado.

O endereço MAC é o endereço da máquina física e localiza-se na *data link layer*. Já endereço IP é um endereço exclusivo que identifica um dispositivo na Internet ou numa rede local, estando na *network layer*. Uma máquina pode possuir vários endereços IP, mas apenas um endereço MAC. Como o ARP é um protocolo de requisição e resposta, a requisição é feita via *broadcast*, solicitando um endereço MAC de uma máquina através do endereço IP. Por sua vez, a resposta é fornecida de forma *Unicast* pela máquina com o endereço lógico requisitado, contendo o endereço físico da mesma.

O protocolo ICMP (*Internet Control Message Protocol*) é utilizado para trocar mensagens de controlo, indicando sucesso ou erros que possam existir durante a comunicação com outro endereço.

Principais comandos:

Configuração de ip:

* *ifconfig ethX*
* *ifconfig ethX <ip-do-computador>*

Verificar conexão entre os dois Tuxs:

* *ping <ip-do-computador-que-pretende-verifcar>*

Diagram

Description automatically generated**4.2-** **Experiência 2 – Implementação de duas *bridges* no *switch***

O objetivo desta experiência é criar duas *bridges* no *switch*, e perceber a conectividade entre os computadores e como estas influenciam a troca de informação entre as máquinas.

Inicialmente repetiu-se o processo de configuração de IP para o Tux 52 (172.16.51.1).

Para configurarmos as *bridges*,criamos a bridge50 e bridge51 através do *GTkterm*, e associamos à primeira o tux53 e o tux54 e à segunda o tux52, tendo assim a arquitetura pretendida. Depois, podemos verificar a conectividade entre os computadores, fazendo um *ping* do tux33 até ao tux34, que foi executado com sucesso, pois estes encontram-se na mesma sub-rede. Fazendo um *ping* do tux53 até ao tux52, não obtivemos resposta, pois não existe nenhuma rota entre as duas *bridges*, tornando impossível o tux53 chegar à interface de rede do tux52. Portanto, podemos concluir que existem dois domínios de *broadcast* correspondentes às sub-redes bridge50 e bridge51.

Principais comandos:

Para criar as *bridges*, no *GTkTerm:*

* */interface bridge add name=bridgeY0*

Depois para adicionar as correspondentes *ports* às *bridges*:

* */interface bridge port add bridge=bridgeY0 interface=etherX*

Por fim utilizamos o comando *ping* pela mesma lógica referida na Experiência 1.

A diagram of a bridge

Description automatically generated**4.3-** **Experiência 3 – Configuração de um *router* em Linux**

O objetivo desta experiência é configurar o tux34 de modo a este funcionar como um *router*, para possibilitar a comunicação entre o tux33 e o tux32 através das *bridges* configuradas na experiência anterior. Para permitir a comunicação é necessário configurar os endereços IP’s das portas e *ethernet* dos computadores e as rotas que serão usadas.

Para conseguirmos ter ligação entre o tux53 e o tux52, é necessário adicionar uma rota ao tux53 para aceder aos endereços 172.16.51.0/24 a partir do IP 172.16.50.254. Assim, quando o tux53 quer enviar um *ping* para a bridge51 este vai utilizar o router, que neste caso é o tux54(172.16.50.254), como *gateway* e uma rota ao tux52 para aceder aos endereços 172.16.50.0/24 a partir do IP 172.16.51.253. Estas rotas podem ser vistas na *forwarding table* utilizando o comando *route -n*. A *forwarding table* é uma tabela onde cada entrada possui informação do tipo *Destino-Gateway-Interface*, em que o destino é o IP do computador de destino, o *gateway* é o IP do computador para o qual se vai enviar a mensagem e a interface é a placa de rede usada para enviar a mensagem. Assim, com estas rotas definidas, é possível fazer *ping*, a partir do tux33, a todas as interfaces dos outros computadores.

Com isto, ao fazer *ping* do tux53 para o tux52 são observados os seguintes pacotes ARP. Primeiro é observado um pacote que pede que o endereço MAC do IP 172.16.50.254 seja enviado para 172.16.50.1. Isto acontece porque o tux53 está a encontrar a interface do tux54 que encaminha o pacote enviado pelo *ping* para o tux52, o endereço MAC de resposta corresponde à interface eth0 de tux54. Em seguida, na mesma interface é observado o pacote ARP que pede que o endereço MAC de 172.16.50.1 seja enviado para 172.16.50.254. Isto acontece porque o tux54 necessita do endereço MAC da interface do tux53 para encaminhar a resposta do tux52 ao *ping*. Já na eth1 do tux54, o tux54 pede qual o endereço MAC do tux52 para lhe poder encaminhar o pacote de *ping* e este dá o endereço MAC de eth0 do tux52. Depois, o tux52 pede o endereço MAC de eth1 do tux54 para lhe poder enviar a resposta a *ping* para que esta seja encaminhada a tux53. O tux54 responde com o endereço MAC de eth1 do tux54. Esta troca de mensagens ARP ocorre sempre que uma mensagem é enviada de uma máquina para outra sendo que os endereços MAC não são conhecidos.

Ao analisar os pacotes ICMP, podemos identificar pacotes do tipo request e reply, indicando que todas as rotas estão devidamente configuradas. Caso não estivessem configuradas, veríamos pacotes ICMP do tipo Host Unreachable. Os endereços IP de destino nos pacotes ICMP sempre correspondem aos IPs de nossa máquina, enquanto os endereços IP associados aos pacotes ICMP de origem representam os IPs pelos quais os pacotes viajam para atingir o destino desejado. O endereço MAC de destino nos pacotes ICMP está associado à interface virtual, ao passo que o endereço MAC de origem nos pacotes ICMP é o endereço MAC da interface virtual do computador host.

**Diagram

Description automatically generated4.4-** **Experiência 4 – Configurar um *router* comercial e implementar NAT**

O objetivo desta experiência é estabelecer uma ligação com a rede dos laboratórios e implementar rotas num *router* comercial, configurando-lhe funcionalidade NAT para garantir a conexão entre as máquinas e a internet.

Inicialmente, é necessário configurar o router. Para realizar esta configuração, iniciamos a sessão no router por meio do GTkTerm e conectamos o cabo S0 de qualquer Tux à entrada de configuração do router. Essa abordagem permite o acesso ao router via GTkTerm, possibilitando a configuração do IP do router e das rotas usando o comando "ip route".

Existem dois cenários possíveis para possibilitar o acesso à internet a partir do computador. No primeiro cenário, uma rota é configurada entre as duas máquinas, e a informação segue essa rota. No segundo cenário, caso essa rota não exista, os pacotes são enviados pela rota padrão, e a máquina para a qual a rota padrão está definida encaminha os pacotes para o destino. Na experiência em questão, como a rota do Tux52 para o Tux53 foi apagada, os pacotes foram redirecionados para o router e, posteriormente, encaminhados para o Tux54.

O NAT (*Network Address Translation*) é um mecanismo implementado em routers que substitui os endereços IP locais nos pacotes por um endereço IP público, permitindo estabelecer conexões para fora da rede. Dessa forma, o router que implementa o NAT assume a responsabilidade de encaminhar todos os pacotes para os endereços corretos, seja dentro ou fora da rede local. Nesta experiência, considerando que o Tux33 deseja enviar um pacote para um endereço em uma rede pública, o pacote é inicialmente enviado para o router. Este router modifica o endereço de origem do pacote para o seu endereço externo, garantindo assim a privacidade e a segurança do remetente. O pacote é então enviado, e, ao receber uma resposta, o router a encaminha de volta para o Tux53, ajustando o destinatário do pacote para o seu próprio endereço. Esse processo possibilita a comunicação entre a rede privada local e a rede pública.

**4.4.2- Principais comandos**

Configurar *ip* e as rotas do *router*:

* */ip address add address=172.16.1.Y9/24 interface=ether1*
* */ip address add address=172.16.Y1.254/24 interface=ether2*
* */ip route add dst-address=172.16.Y0.0/24 gateway=172.16.Y1.253*
* */ip route add address=0.0.0.0/0 gateway=172.16.2.254*

No tuxy2 definir RC como *default* *router*:

* *route add default gw 172.16.y1.254*

No tuxy3 definir tuxy4 como *default* *router*:

* *route add default gw 172.16.y0.254*

No tuxy4 definir RC como *default* *router*:

* *route add default gw 172.16.y1.254*

**Diagram, schematic

Description automatically generated4.5-** **Experiência 5 – DNS**

O objetivo desta experiência é configurar o DNS responsável por traduzir endereços URL em endereços IP.

O serviço DNS é configurado no ficheiro *resolv.conf* bastando adicionar no ficheiro:

* search netlab.fe.up.pt
* nameserver 172.16.1.1