**✏️ Apontamentos internos**

# Experiências

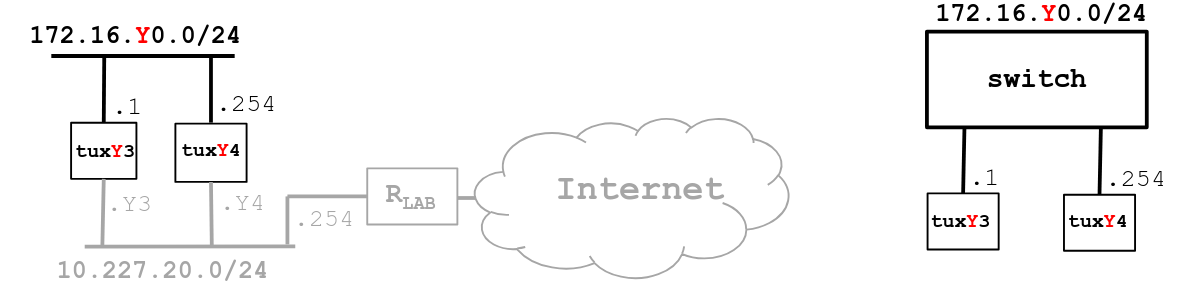
1. Configurar uma rede IP
2. Implementação de duas “bridges” num “switch”
3. Configurar um router em Linux
4. Configurar um router comercial e implementar NAT
5. DNS
6. Conexões TCP

# Limpar máquinas do laboratório

1. systemctl restart networking  
   (Executar em cada um dos 4 TUXs)
2. Conectar (usando os fios pretos) a consola do switch ao tux64S0 (ou tux63S0, …)
3. Abrir o GTKTerm e configurar a porta para baudrate de 115200
4. Realizar login (se necessário):  
   Utilizador: admin  
   Palavra-passe: <branco> (apenas clique Enter)
5. No switch, executar /system reset-configuration e confirmar com Y.
6. // TODO: Resetar router (ainda não aplicável)

**#1**

## #1: Configurar uma rede IP



### Instruções dadas:

* Connect E1 of tuxY3 and E1 of tuxY4 to the switch
* Configure eth1interface of tuxY3 and eth1 interface of tuxY4 using ifconfig and route commands
* Register the IP and MAC addresses of the network interfaces
* Use ping command to verify connectivity between these computers
* Inspect forwarding (route -n) and ARP (arp -a) tables
* Delete ARP table entries in tuxY3 (arp -d ipaddress)
* Start Wireshark in tuxY3.eth1 and start capturing packets
* In tuxY3, ping tuxY4 for a few seconds
* Stop capturing packets
* Save the log and study it at home

### Comandos executados:

// TODO



### Dados extra registados:

* // TODO

### Ficheiros Wireshark:

// TODO

**#2**

## #2: Implementação de duas “bridges” num “switch”

## 

### Instruções dadas:

* Connect and configure E1 of tuxY2 and register its IP and MAC addresses
* Create two bridges in the switch: bridgeY0 and bridgeY1
* Remove the ports where tuxY3, tuxY4 and tuxY2 are connected from the default bridge (bridge) and add them the corresponding ports to bridgeY0 and bridgeY1
* Start the capture at tuxY3.eth1
* In tuxY3, ping tuxY4 and then ping tuxY2
* Stop the capture and save the log
* Start new captures in tuxY2.eth1, tuxY3.eth1, tuxY4.eth1
* In tuxY3, do ping broadcast (ping -b 172.16.Y0.255) for a few seconds
* Observe the results, stop the captures and save the logs
* Repeat steps 7, 8 and 9, but now do ping broadcast in tuxY2 (ping -b 172.16.Y1.255)

### Comandos executados:

// TODO



### Dados extra registados:

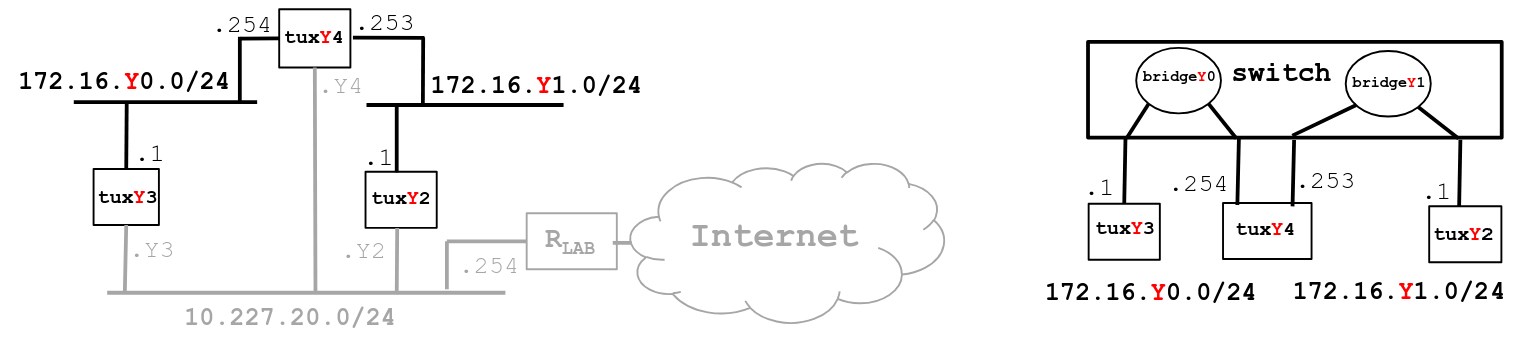
* // TODO

### Ficheiros Wireshark:

// TODO

**#3**

## #3: Configurar um router em Linux



### Instruções dadas:

* Transform tuxY4 (Linux) into a router
  + Configure also eth2 interface of tuxY4 and add it to bridgeY1
  + Enable IP forwarding
  + Disable ICMP echo-ignore-broadcast
* Observe MAC addresses and IP addresses in tuxY4.eth1 and tuxY4.eth2
* Reconfigure tuxY3 and tuxY2 so that each of them can reach the other
* Observe the routes available at the 3 tuxes (route -n)
* Start capture at tuxY3
* From tuxY3, ping the other network interfaces (172.16.Y0.254, 172.16.Y1.253, 172.16.Y1.1) and verify if there is connectivity
* Stop the capture and save the logs
* Start capture in tuxY4; use 2 instances of Wireshark, one per network interface
* Clean the ARP tables in the 3 tuxes
* In tuxY3, ping tuxY2 for a few seconds.
* Stop captures in tuxY4 and save logs

### Comandos executados:

// TODO



### Dados extra registados:

* // TODO

### Ficheiros Wireshark:

* // TODO

**#4**

## #4: Configurar um router comercial e implementar NAT

## 

### Instruções dadas:

* Connect ether1 of RC to the lab network on PY.12 (with NAT enabled by default) and ether2 of RC to a port on bridgeY1. Configure the IP addresses of RC through the router serial console
* Verify routes
  + tuxY4 as default router of tuxY3;
  + RC as default router for tuxY2 and tuxY4
  + in tuxY2 and RC add routes for 172.16.Y0.0/24
* Using ping commands and Wireshark, verify if tuxY3 can ping all the network interfaces of tuxY2, tuxY4 and RC
* In tuxY2, do the following:
  + sysctl net.ipv4.conf.eth0.accept\_redirects=0
  + sysctl net.ipv4.conf.all.accept\_redirects=0
  + remove the route to 172.16.Y0.0/24 via tuxY4
  + In tuxY2, ping tuxY3
  + Using capture at tuxY2, try to understand the path followed by ICMP ECHO and ECHO-REPLY packets (look at MAC addresses)
  + In tuxY2, do traceroute tuxY3
  + In tuxY2, add again the route to 172.16.Y0.0/24 via tuxY4 and do traceroute tuxY3
  + Activate the acceptance of ICMP redirect at tuxY2 when there is no route to 172.16.Y0.0/24 via tuxY4 and try to understand what happens
* In tuxY3, ping the FTP server (172.16.1.10) and try to understand what happens
* Disable NAT functionality in router RC
* In tuxY3 ping 172.16.1.10, verify if there is connectivity, and try to understand what happens

### Comandos executados:

// TODO



### Dados extra registados:

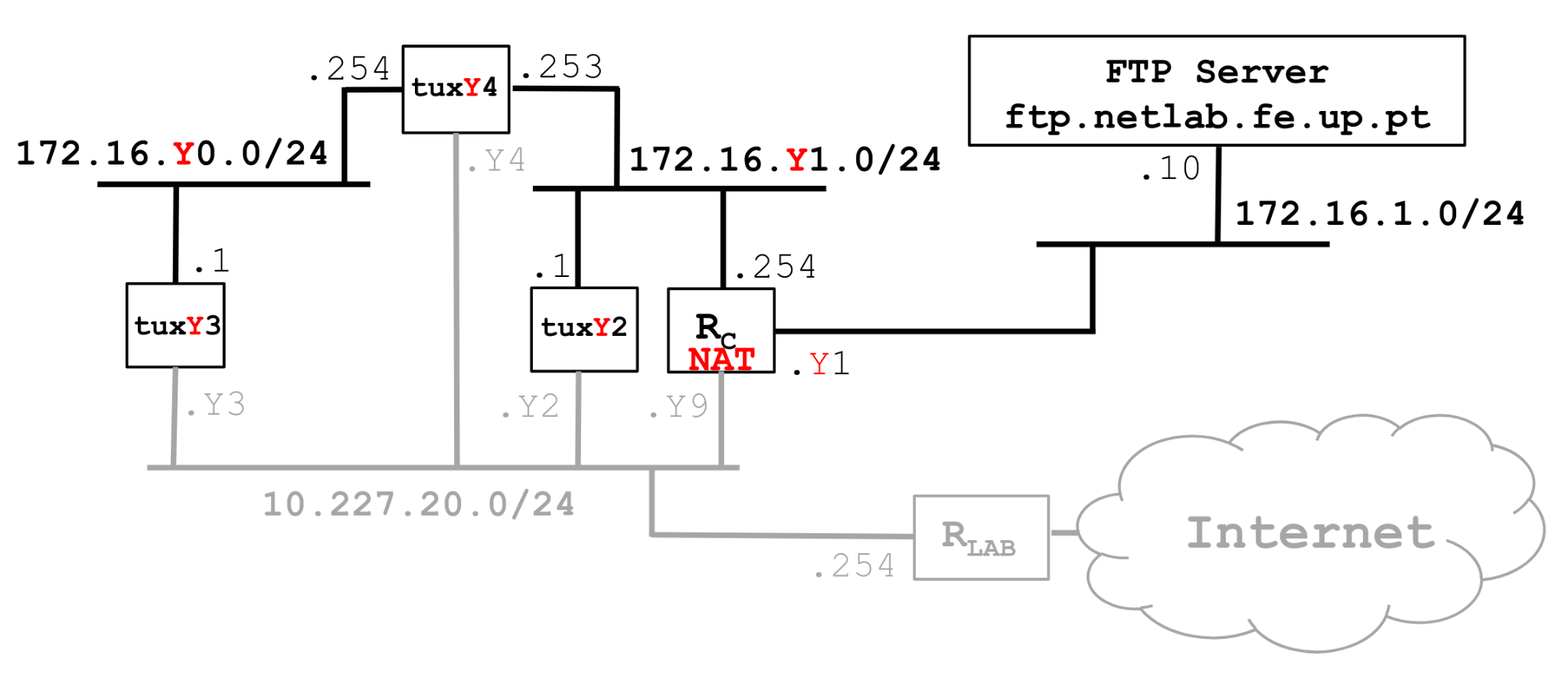
* // TODO

### Ficheiros Wireshark:

* // TODO

**#5**

## #5: DNS



### Instruções dadas:

* Configure DNS at tuxY3, tuxY4, tuxY2 (use DNS server services.netlab.fe.up.pt (10.227.20.3))
* Verify if names can be used in these hosts (e.g., ping hostname, use browser)
* Execute ping (new-hostname-in-the-Internet); observe DNS related packets in Wireshark

### Comandos executados:

// TODO



### Dados extra registados:

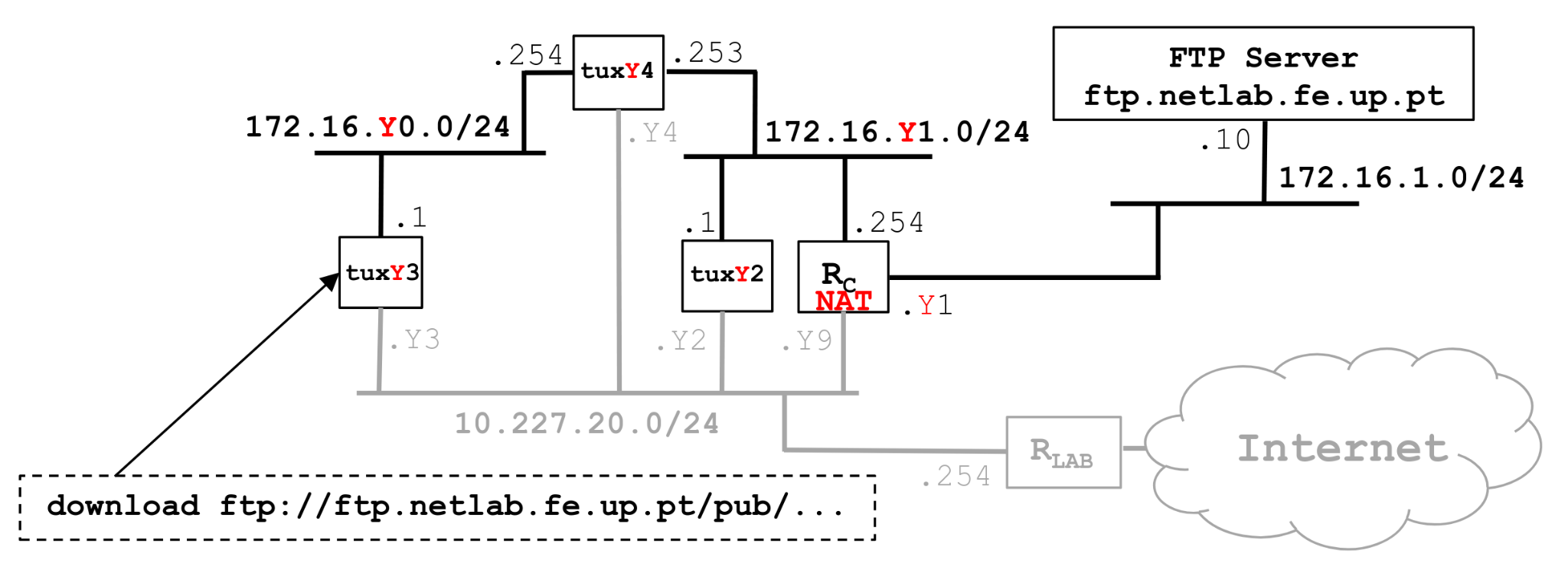
* // TODO

### Ficheiros Wireshark:

* // TODO

**#6**

## #6: Conexões TCP



### Instruções dadas:

* Compile your download application in tuxY3
* In tuxY3, restart capturing with Wireshark and run your application
* Verify if file has arrived correctly, stop capturing and save the log

### Comandos executados:

// TODO



### Dados extra registados:

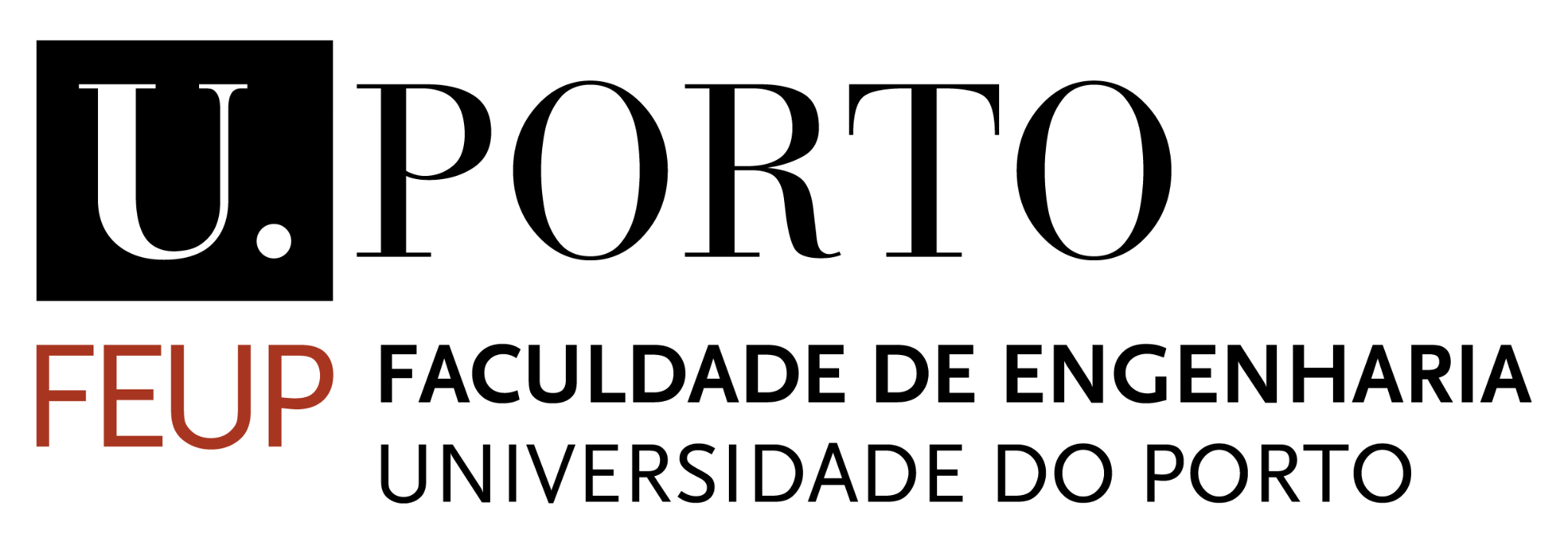
* // TODO

### Ficheiros Wireshark:

* // TODO

**📃 [RCOM] Relatório #2**

Faculdade de Engenharia da Universidade do Porto



**2.º projeto laboratorial**

// TODO

**Redes de Computadores (L.EIC025) 2024/2025  
Licenciatura em Engenharia Informática e Computação**

| Manuel Alberto Pereira Ricardo | (Co-regente do curso) |
| --- | --- |
| Rui Pedro de Magalhães Claro Prior | (Co-regente do curso) |
| Eduardo Nuno Moreira Soares de Almeida | (Professor das aulas laboratoriais) |

**Turma 2:**

| // TODO | upXXXXXXXXX@up.pt |
| --- | --- |
| // TODO | upXXXXXXXXX@up.pt |

# 

**Índice**

[Resumo 1](#_1ci93xb)

[1. Introdução 1](#_uw7aumoe4riv)

[2. Aplicação de transferência 1](#_9dkua7xm64ku)

[2.1. Arquitetura 1](#_uxiucwqpi22e)

[2.2. Exemplo de uma transferência com sucesso 1](#_1q5k4lc03h0l)

[3. Configuração e análise da rede 1](#_h9e7g0zbjczh)

[3.1. Configurar uma rede IP 1](#_aucc2oo04i3x)

[3.1.1. Arquitetura da rede 1](#_dzv7nblo62b8)

[3.1.2. Objetivos da experiência 1](#_8lyr1vicews9)

[3.1.3. Comandos principais de configuração 1](#_7fhsx6ad4etn)

[3.1.4. Análise dos registos 1](#_ltv3339w2ttg)

[3.1.5. Resolução das perguntas propostas 1](#_pb2cn2mmj92a)

[3.2. Implementação de duas “bridges” num “switch” 2](#_b3t2xng6713p)

[3.2.1. Arquitetura da rede 2](#_q9v070b7h2fw)

[3.2.2. Objetivos da experiência 2](#_rz3dsh4o1cu2)

[3.2.3. Comandos principais de configuração 2](#_au0e64hx6wi5)

[3.2.4. Análise dos registos 2](#_i21wm4orcx2w)

[3.2.5. Resolução das perguntas propostas 2](#_yzx6sf4n4r3x)

[3.3. Configurar um router em Linux 3](#_mc4oethz314s)

[3.3.1. Arquitetura da rede 3](#_u490l54k3ihl)

[3.3.2. Objetivos da experiência 3](#_kvrh8czcwne6)

[3.3.3. Comandos principais de configuração 3](#_t8bh0j7c0v0o)

[3.3.4. Análise dos registos 3](#_h3lhlrh635cm)

[3.3.5. Resolução das perguntas propostas 3](#_arvv8yxqzt3x)

[3.4. Configurar um router comercial e implementar NAT 4](#_bl8217sbu08)

[3.4.1. Arquitetura da rede 4](#_mdsiqxeorrtt)

[3.4.2. Objetivos da experiência 4](#_hq9c34rx9k67)

[3.4.3. Comandos principais de configuração 4](#_aevn1y22cnip)

[3.4.4. Análise dos registos 4](#_v2u9qeb8vmxg)

[3.4.5. Resolução das perguntas propostas 4](#_lm1ho1220066)

[3.5. DNS 4](#_h9kr7lef2hnu)

[3.5.1. Arquitetura da rede 4](#_prnqv6ra8gj8)

[3.5.2. Objetivos da experiência 5](#_ueeuqektq2lj)

[3.5.3. Comandos principais de configuração 5](#_36wzqz8abcah)

[3.5.4. Análise dos registos 5](#_kenmmol22941)

[3.5.5. Resolução das perguntas propostas 5](#_4mlrplu7uiz4)

[3.6. Conexões TCP 5](#_khmy3mr08887)

[3.6.1. Arquitetura da rede 5](#_u0if76gqb9r0)

[3.6.2. Objetivos da experiência 5](#_hxc2xlxzk7g4)

[3.6.3. Comandos principais de configuração 5](#_ya8ruakql199)

[3.6.4. Análise dos registos 5](#_4zpkp89f4by)

[3.6.5. Resolução das perguntas propostas 6](#_wuw9mv809ucl)

[4. Conclusões 6](#_4yt2092biqv1)

[Apêndices 7](#_qb8m3zct79pj)

[A. Comandos de configuração 7](#_pqr9873rkc5w)

[A.1. Configurar uma rede IP 7](#_gk0d57l5gx20)

[A.2. Implementação de duas “bridges” num “switch” 7](#_7vf5pai8di6g)

[A.3. Configurar um router em Linux 7](#_jju8niqjcyh4)

[A.4. Configurar um router comercial e implementar NAT 7](#_biduxie0tv5h)

[A.5. DNS 7](#_16hufwn0r4z5)

[A.6. Conexões TCP 7](#_drvamxe7y325)

[B. Registos capturados 8](#_1vinkal6km4s)

[B.1. Configurar uma rede IP 8](#_c78r78k3chue)

[B.2. Implementação de duas “bridges” num “switch” 8](#_y1u0umain5z9)

[B.3. Configurar um router em Linux 8](#_ud7oqkncsdkq)

[B.4. Configurar um router comercial e implementar NAT 8](#_cvubxlpa2sna)

[B.5. DNS 8](#_3snfzz4vw61e)

[B.6. Conexões TCP 8](#_cl01ko6ldg1u)

[C. Código-fonte 9](#_zdsjp8o5kzw5)

[file (// TODO) 9](#_25y2mgh9qqrl)

# 

# Resumo

// TODO

# 1. Introdução

// TODO

# 2. Aplicação de transferência

## 2.1. Arquitetura

// TODO

## 2.2. Exemplo de uma transferência com sucesso

// TODO

# 3. Configuração e análise da rede

## 3.1. Configurar uma rede IP

### 3.1.1. Arquitetura da rede

// TODO

### 3.1.2. Objetivos da experiência

// TODO

### 3.1.3. Comandos principais de configuração

// TODO

### 3.1.4. Análise dos registos

// TODO

### 3.1.5. Resolução das perguntas propostas

* **O que são os pacotes ARP e para que são utilizados?**// TODO
* **Quais são os endereços MAC e IP dos pacotes ARP e por quê?**  
  // TODO
* **Que pacotes o comando ping gera?**  
  // TODO
* **Quais são os endereços MAC e IP dos pacotes gerados pelo ping?**  
  // TODO
* **Como determinar se uma trama Ethernet a receber é ARP, IP, ICMP?**  
  // TODO
* **Como determinar o comprimento de uma trama a receber?**  
  // TODO
* **O que é a “interface” de “loopback” e por que é importante?**// TODO

## 3.2. Implementação de duas “bridges” num “switch”

### 3.2.1. Arquitetura da rede

// TODO

### 3.2.2. Objetivos da experiência

// TODO

### 3.2.3. Comandos principais de configuração

// TODO

### 3.2.4. Análise dos registos

// TODO

### 3.2.5. Resolução das perguntas propostas

* **Como configurar o bridgeY0?**  
  // TODO
* **Quantos domínios de transmissão existem? Como pode concluir isso a partir dos registos?**

// TODO

## 3.3. Configurar um router em Linux

### 3.3.1. Arquitetura da rede

// TODO

### 3.3.2. Objetivos da experiência

// TODO

### 3.3.3. Comandos principais de configuração

// TODO

### 3.3.4. Análise dos registos

// TODO

### 3.3.5. Resolução das perguntas propostas

* **Que rotas existem nos ”tuxes”? Qual o seu significado?**

// TODO

* **Que informação contém uma entrada da tabela de encaminhamento?**

// TODO

* **Que mensagens ARP e endereços MAC associados são observados e por quê?**

// TODO

* **Que pacotes ICMP são observados e por quê?**

// TODO

* **Quais são os endereços IP e MAC associados aos pacotes ICMP e por quê?**

// TODO

## 3.4. Configurar um router comercial e implementar NAT

### 3.4.1. Arquitetura da rede

// TODO

### 3.4.2. Objetivos da experiência

// TODO

### 3.4.3. Comandos principais de configuração

// TODO

### 3.4.4. Análise dos registos

// TODO

### 3.4.5. Resolução das perguntas propostas

* **Como configurar uma rota estática num router comercial?**

// TODO

* **Quais os caminhos seguidos pelos pacotes, com e sem redirecionamento ICMP habilitado, nas experiências realizadas e por quê?**

// TODO

* **Como configurar o NAT num router comercial?**

// TODO

* **O que faz o NAT?**

// TODO

* **O que acontece quando o tuxY3 faz ping ao servidor FTP com o NAT desativado? Por quê?**

// TODO

## 3.5. DNS

### 3.5.1. Arquitetura da rede

// TODO

### 3.5.2. Objetivos da experiência

// TODO

### 3.5.3. Comandos principais de configuração

// TODO

### 3.5.4. Análise dos registos

// TODO

### 3.5.5. Resolução das perguntas propostas

* **Como configurar o serviço DNS num “host”?**

// TODO

* **Que pacotes são trocados pelo DNS e que informação é transportada?**

// TODO

## 3.6. Conexões TCP

### 3.6.1. Arquitetura da rede

// TODO

### 3.6.2. Objetivos da experiência

// TODO

### 3.6.3. Comandos principais de configuração

// TODO

### 3.6.4. Análise dos registos

// TODO

### 3.6.5. Resolução das perguntas propostas

* **Quantas ligações TCP são abertas pela sua aplicação FTP?**

// TODO

* **Em que conexão é transportada a informação de controlo do FTP?**

// TODO

* **Quais as fases de uma conexão TCP?**

// TODO

* **Como funciona o mecanismo ARQ TCP? Quais são os campos TCP relevantes? Que informação relevante pode ser observada nos logs?**

// TODO

* **Como funciona o mecanismo de controlo de congestionamento do TCP? Quais são os relevantes campos? Como evoluiu o rendimento da ligação de dados ao longo do tempo? Está conforme o mecanismo de controlo de congestionamento TCP?**

// TODO

* **A taxa de transferência das ligações de dados TCP é perturbada pelo aparecimento de uma segunda ligação TCP? Como?**

// TODO

# 4. Conclusões

// TODO

# Apêndices

## A. Comandos de configuração

### A.1. Configurar uma rede IP

// TODO

### A.2. Implementação de duas “bridges” num “switch”

// TODO

### A.3. Configurar um router em Linux

// TODO

### A.4. Configurar um router comercial e implementar NAT

// TODO

### A.5. DNS

// TODO

### A.6. Conexões TCP

// TODO

## B. Registos capturados

### B.1. Configurar uma rede IP

// TODO

### B.2. Implementação de duas “bridges” num “switch”

// TODO

### B.3. Configurar um router em Linux

// TODO

### B.4. Configurar um router comercial e implementar NAT

// TODO

### B.5. DNS

// TODO

### B.6. Conexões TCP

// TODO

## 

## C. Código-fonte

### file (// TODO)

// TODO

