U.PORTO

Projeto 2
Redes de Computadores

Orangel Azuaje Contreras (orangel.contreras@fe.up.pt)
Redes de Computadores

11/11/2022

Planeamento das aulas laboratorias

U.PORTO

Semana de 7/nov:

- Parte 1
- Experiências com telnet; ênfase no protocolo
- Programação com funções de sockets e gethostbyname
- Arquitetura do cliente de download e caracterização do seu caso de uso principal

Semana de 14/nov:

- Parte 2: Passos 1 e 2
- Linux, laboratório, cablagem, ifconfig, route, arp, wireshark

Semana de 21/nov:

- Parte 2: Passos 2, 3, 4, 5 e 6

Semana de 28/nov:

- Parte 2: Passos 2, 3, 4, 5 e 6.

Semana de 5/dez:

- Avaliação do 2º trabalho laboratorial
- Demonstração da aplicação desenvolvida sobre a rede configurada no laboratório

Semana de 12/dez:

- Avaliação do 2º trabalho laboratorial
- Demonstração da aplicação desenvolvida sobre a rede configurada no laboratório

U.PORTO

Evaluation of RCOM

Frequência

- L1 - grade of 1st lab
- L2 - grade of 2nd lab
- H - grade of homeworks
- FQ - grade of FREQUÊNCIA

- FQ = 0,4*L1 + 0,4*L2 + 0,2*H

- if (FQ < 8,0) FQ = "No Admission to Exams"

Classificação Final

- E - grade of final exam
- FQ - grade of FREQUÊNCIA
- AD - grade of distributed evaluation
- CF - final grade

- if (FQ > E + 5) AD = E + 5 else AD = FQ

- CF = 0.4 * AD + 0.6*E

- if (E < 8.0) CF = E

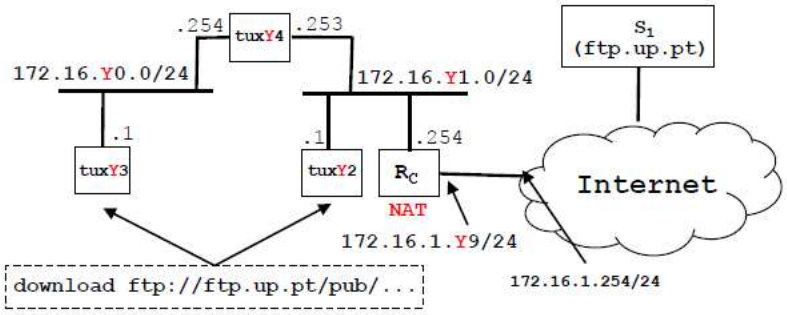
U.PORTO

Lab Work – Two parts

- Part 1 – Development of a download application

  download ftp://ftp.up.pt/pub/...

- Part 2 – Configuration and study of a computer network



U.PORTO

Development of an Application

- Develop application `download ftp://ftp.up.pt/pub/...`
  - » Application downloads a single file
  - » Implements FTP application protocol, as described in RFC959
  - » Adopts URL syntax, as described in RFC1738
    `ftp://[<user>:<password>@]<host>/<url-path>`
- Steps
  - » Experiments using Telnet application (Telnet, SMTP, POP, HTTP and FTP); focus on FTP
  - » Specification/design of a download application
    - unique use case: connect, login host, passive, get path, success (file saved in CWD) or un-success (indicating failing phase)
    - challenging programming aspects: gethostbyname, sockets, control connection, passive, data connection
  - » Implement a very simple FTP client at home
    - reuse existing programs: **`clientTCP.c, getIP.c`**
- Learning objectives
  - » Describe client – server concept and its peculiarities in TCP/IP
  - » Characterize application protocols in general, characterize URL, describe in detail the behaviour of FTP
  - » Locate and read RFCs
  - » Implement a simple FTP client in C language
  - » Use sockets and TCP in C language
  - » Understand service provided DNS and use it within a client program

Step Back

U.PORTO

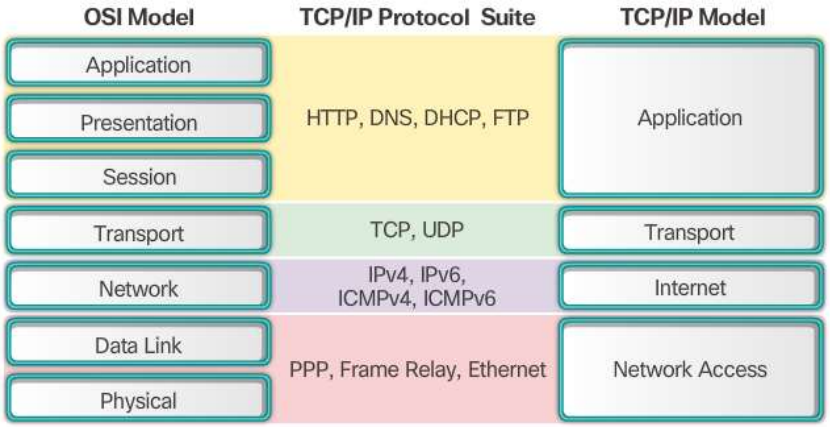
- What is a (network) app?

  A network application is any application running on one host providing communication to another application running on a different host.

- FTP application protocol?

  - What is a protocol?

    A network protocol is an established set of rules that determine how data is transmitted between different devices in the same network.



Step Back

U.PORTO

- FTP application protocol?
  - What is FTP?

    The File Transfer Protocol (FTP) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network.

  - What is RFC 959?

    Request for Comments is a publication in a series from the principal technical development and standards-setting bodies for the Internet.

- What is URL?
  - A Uniform Resource Locator (URL) is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.

U.PORTO

Experiência de FTP (transferência de ficheiro)

- **Objetivo:** Transferir um ficheiro
- **RFC:** STD 9/RFC 959 File Transfer Protocol

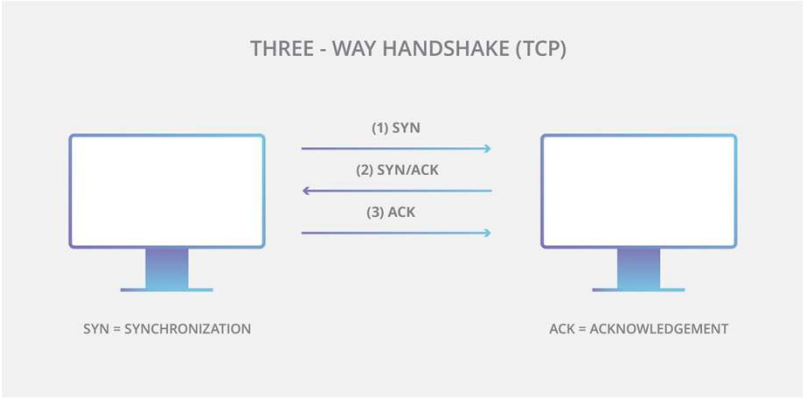
O FTP usa uma ligação TCP para controlo e, para cada ficheiro ou dados (ex.: listagem do diretório) a transferir, abre **uma nova ligação TCP**. Esta **ligação de dados** pode ser feita num modo *passivo* ou *ativo*. No modo:

- *ativo* o **servidor liga-se ao cliente**
- *passivo* o **cliente liga-se ao servidor**

- What is TCP?

The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite. It allows us to deliver of a stream of octets (bytes) between applications running on hosts communicating via an IP network.

1. Reliable
2. Ordered
3. Error-checked
4. Connection-oriented



Serão necessárias dois terminais (`term_A` e `term_B`) para efetuar esta experiência.

No `term_A`

term_A		Servidor
`telnet ftp.up.pt 21`	⇒	
	⇐	`220-Welcome to the University of Porto's mirror archive (mirrors.up.pt)` [...]
`user anonymous`	⇒	
	⇐	`331 Please specify the password.`
`pass qualquer-password`	⇒	
	⇐	`230 Login successful.`
`pasv`	⇒	
	⇐	`227 Entering Passive Mode (193,136,28,12,19,91)`

Establish control channel

- Passive mode: clients connects to the server (data channel)
- IP of the server: 193.136.28.12
- Port: 19*256 + 91 = 4955

No `term_B`, faça então

term_B		Servidor
`telnet ftp.up.pt 4955`	⇒	
	⇐	[...]

Establish data channel

Na `term_A`, de novo

term_A		Servidor
`retr pub/kodi/timestamp.txt`	⇒	
	⇐	`150 Opening BINARY mode data connection for pub/kodi/apt/pre-release/ios/Release (179 bytes).` `226 Transfer complete.`

E observe o ficheiro recebido no `term_B`. Cada terminal funciona como a extremidade de uma ligação.

U.PORTO

Projeto 2
Redes de Computadores

Orangel Azuaje Contreras (orangel.contreras@fe.up.pt)
Redes de Computadores

18/11/2022

U.PORTO

Development of an Application

- Develop application `download ftp://ftp.up.pt/pub/...`
  - » Application downloads a single file
  - » Implements FTP application protocol, as described in RFC959
  - » Adopts URL syntax, as described in RFC1738
    `ftp://[<user>:<password>@]<host>/<url-path>`
- Steps
  - » Experiments using Telnet application (Telnet, SMTP, POP, HTTP and FTP); focus on FTP
  - » Specification/design of a download application
    - unique use case: connect, login host, passive, get path, success (file saved in CWD) or un-success (indicating failing phase)
    - challenging programming aspects: gethostbyname, sockets, control connection, passive, data connection
  - » Implement a very simple FTP client at home
    - reuse existing programs: **`clientTCP.c, getIP.c`**
- Learning objectives
  - » Describe client – server concept and its peculiarities in TCP/IP
  - » Characterize application protocols in general, characterize URL, describe in detail the behaviour of FTP
  - » Locate and read RFCs
  - » Implement a simple FTP client in C language
  - » Use sockets and TCP in C language
  - » Understand service provided DNS and use it within a client program

Sockets in C?

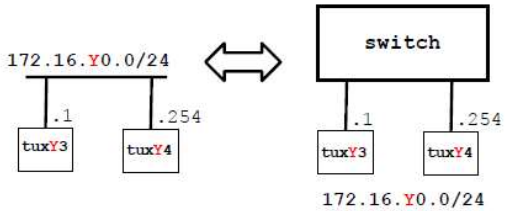
Control or data connection?

URL syntax?

So…you tested the program, right?

U.PORTO

Part 2 / Exp 1- Configure an IP Network



Steps

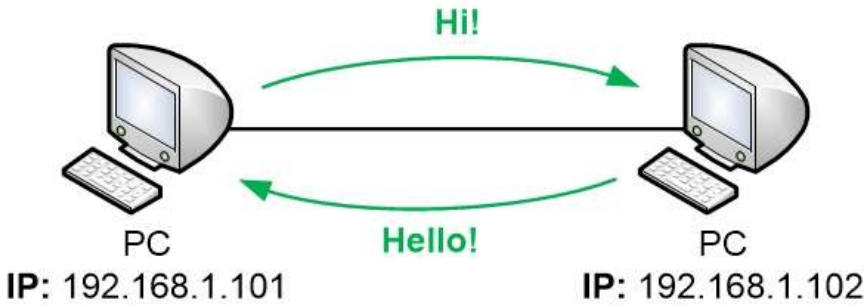
1. Disconnect the switch from **netlab** (PY.1). Connect tuxY3 and tuxY4 to the switch
2. Configure tuxY3 and tuxY4 using **ifconfig** and **route** commands
3. Register the IP and MAC addresses of network interfaces
4. Use **ping** command to verify connectivity between these computers
5. Inspect forwarding (***route –n***) and ARP (***arp –a***) tables
6. Delete ARP table entries in tuxY3 (***arp –d ipaddress***)
7. Start **Wireshark** in tuxY3.eth0 and start capturing packets
8. In tuxY3, **ping** tuxY4 for a few seconds
9. Stop capturing packets
10. Save the log and study it at home

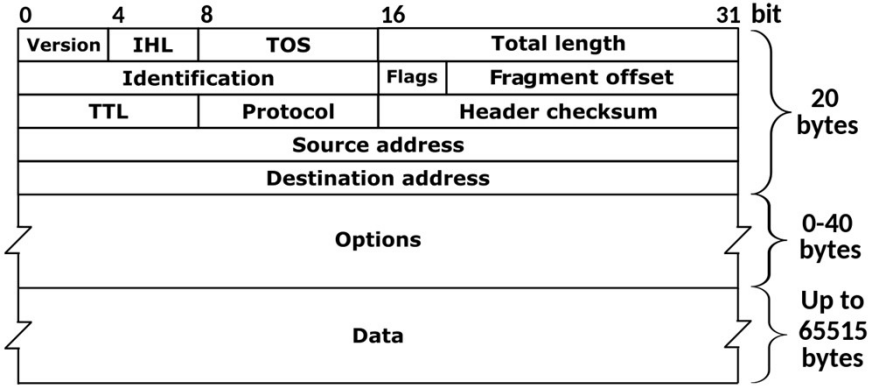
- What is IP?

  An Internet Protocol address (IP address) is a numerical label such as 192.0.2.1 that is connected to a computer network that uses the Internet Protocol for communication. An IP address serves two main functions: network interface identification and location addressing.

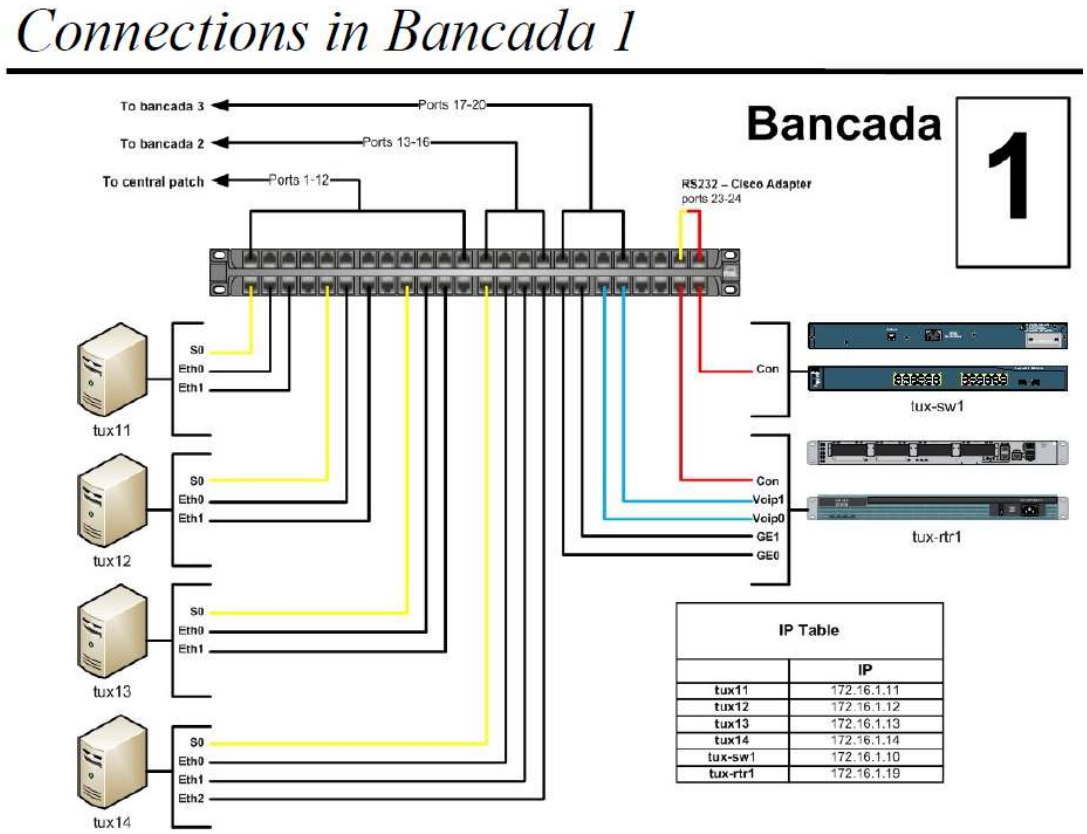
- What is an IP network?

  An IP network is a group of computers connected via their unique internet protocol (IP) addresses.





- Configuração nas bancadas



1. Disconnect the switch from netlab (PY.1). Connect tuxY3 and tuxY4 to the switch

Desligar todos os cabos do switch

Ligar tuxY3 e tuxY4 ao switch

2. Configure tuxY3 and tuxY4 using ifconfig and route commands

Configure tuxY3

ifconfig eth0 172.16.10.1/24
ifconfig eth0 172.16.10.1 netmask 255.255.255.0

What is netmask?

What is the other configuration?

Configure tuxY4

ifconfig eth0 172.16.10.254/24
ifconfig eth0 172.16.10.254 netmask 255.255.255.0

3. Register the IP and MAC addresses of network interfaces

ifconfig

4. Use ping command to verify connectivity between these computers

tuxY3 -> tuxY4: *ping 172.16.10.254*
tuxY4 -> tuxY3: *ping 172.16.10.1*

5. Inspect forwarding (route -n) and ARP (arp -a) tables

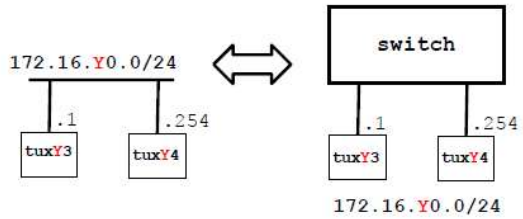
route -n
arp –a

6. Delete ARP table entries in tuxY3 (arp -d ipaddress)

tux13: *arp -d 172.16.10.254*
tux14: *arp -d 172.16.10.1*

U.PORTO

Part 2 / Exp 1- Configure an IP Network



Questions

- What are the ARP packets and what are they used for?
- What are the MAC and IP addresses of ARP packets and why?
- What packets does the ping command generate?
- What are the MAC and IP addresses of the ping packets?
- How to determine if a receiving Ethernet frame is ARP, IP, ICMP?
- How to determine the length of a receiving frame?
- What is the loopback interface and why is it important?

- WARNING: Não esqueçam

  Reset ethernet configuration

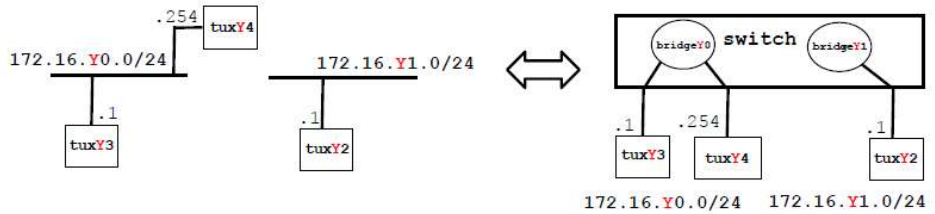
systemctl restart networking

  Reset switch configuration

/system reset-configuration

U.PORTO

Part 2 / Exp 2 – Implement two bridges in a switch



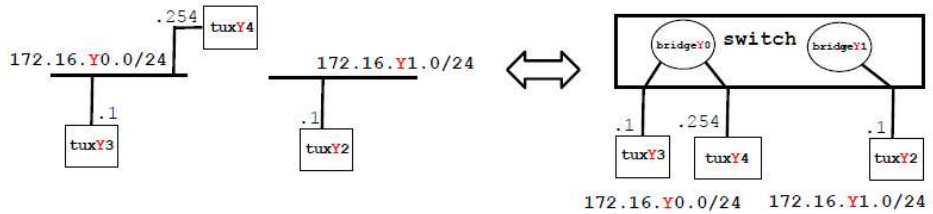
Steps

1. Connect and configure tuxY2 and register its IP and MAC addresses ← Vocês
2. Create two **bridges** in the switch: bridgeY0 and bridgeY1
3. 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
4. Start the capture at tuxY3.eth0
5. In tuxY3, ping tuxY4 and then ping tuxY2
6. Stop the capture and save the log
7. Start new captures in tuxY2.eth0, tuxY3.eth0, tuxY4.eth0
8. In tuxY3, do ping broadcast ***(ping –b 172.16.Y0.255)*** for a few seconds
9. Observe the results, stop the captures and save the logs
10. Repeat **steps 7, 8** and **9**, but now do
    - ping broadcast in tuxY2 ***(ping –b 172.16.Y1.255)***

U.PORTO

Part 2 / Exp 2 – Implement two bridges in a switch

Steps

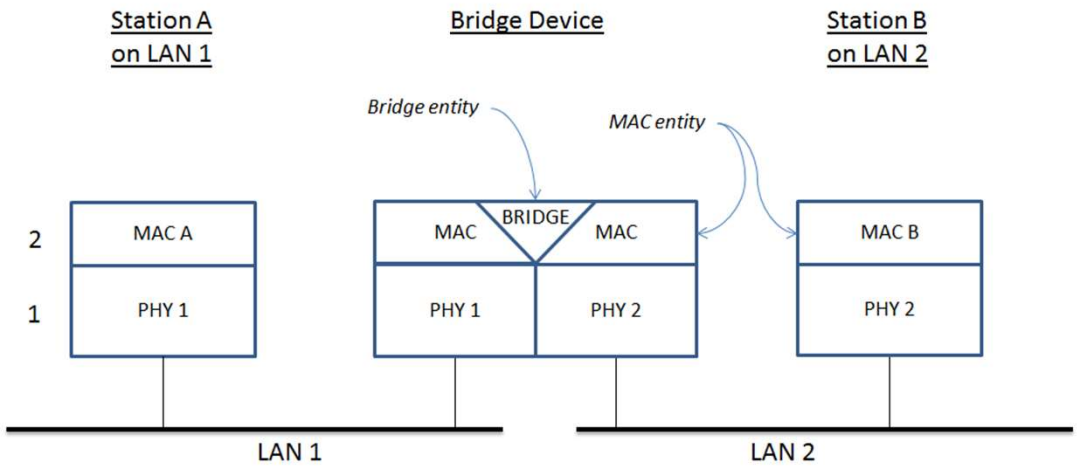


1. Connect and configure tuxY2 and register its IP and MAC addresses
2. Create two **bridges** in the switch: bridgeY0 and bridgeY1
3. Remove the ports where tuxY3, tuxY4 and tuxY2 are connected from the default bridge (***bridge***) and add them the corresponding ports to bridgeY0 and bridgeY1
4. Start the capture at tuxY3.eth0
5. In tuxY3, ping tuxY4 and then ping tuxY2
6. Stop the capture and save the log
7. Start new captures in tuxY2.eth0, tuxY3.eth0, tuxY4.eth0
8. In tuxY3, do ping broadcast ***(ping –b 172.16.Y0.255)*** for a few seconds
9. Observe the results, stop the captures and save the logs
10. Repeat **steps 7, 8** and **9**, but now do
    - ping broadcast in tuxY2 ***(ping –b 172.16.Y1.255)***

- Bridge: A network bridge is a computer networking device that creates a single, aggregate network from multiple communication networks or network segments.

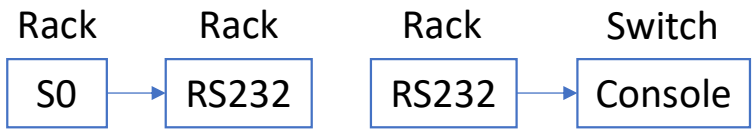
- Bridging is distinct from routing. Routing allows multiple networks to communicate independently and yet remain separate, whereas bridging connects two separate networks as if they were a single network.

A bridge connecting two LAN segments



2. Create two bridges in the switch: bridgeY0 and bridgeY1

- Connect to the switch



Abrir GTKterm@115200
Carregar ENTER
user: admin
pass: (blank)

- Create two bridges on the switch

/interface bridge add name=bridgeY0
/interface bridge add name=bridgeY1

3. 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

/interface bridge port remove [find interface=etherXX] ← How many times?

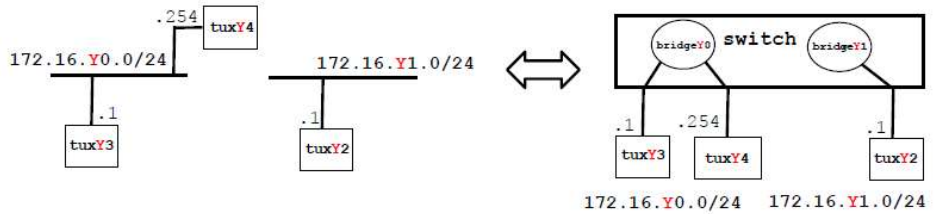
/interface bridge port add interface=etherXX bridge=bridgeYX ← How many times?

/interface bridge port print brief

U.PORTO

Part 2 / Exp 2 – Implement two bridges in a switch

Steps



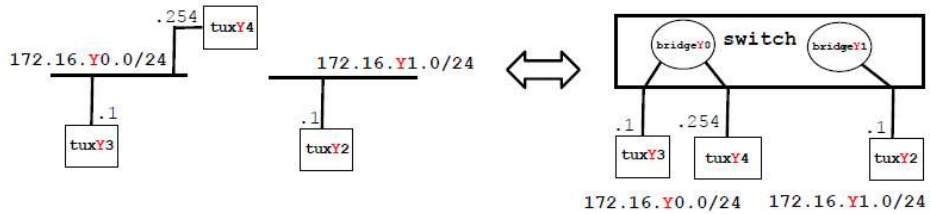
1. Connect and configure tuxY2 and register its IP and MAC addresses
2. Create two **bridges** in the switch: bridgeY0 and bridgeY1
3. Remove the ports where tuxY3, tuxY4 and tuxY2 are connected from the default bridge (***bridge***) and add them the corresponding ports to bridgeY0 and bridgeY1
4. Start the capture at tuxY3.eth0
5. In tuxY3, ping tuxY4 and then ping tuxY2
6. Stop the capture and save the log
7. Start new captures in tuxY2.eth0, tuxY3.eth0, tuxY4.eth0
8. In tuxY3, do ping broadcast ***(ping –b 172.16.Y0.255)*** for a few seconds
9. Observe the results, stop the captures and save the logs
10. Repeat **steps 7, 8** and **9**, but now do
    - ping broadcast in tuxY2 ***(ping –b 172.16.Y1.255)***

Vocês

U.PORTO

Part 2 / Exp 2 – Implement two bridges in a switch

Steps



1. Connect and configure tuxY2 and register its IP and MAC addresses
2. Create two **bridges** in the switch: bridgeY0 and bridgeY1
3. 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
4. Start the capture at tuxY3.eth0
5. In tuxY3, ping tuxY4 and then ping tuxY2
6. Stop the capture and save the log
7. Start new captures in tuxY2.eth0, tuxY3.eth0, tuxY4.eth0
8. In tuxY3, do ping broadcast ***(ping –b 172.16.Y0.255)*** for a few seconds
9. Observe the results, stop the captures and save the logs
10. Repeat **steps 7, 8** and **9**, but now do
    - ping broadcast in tuxY2 ***(ping –b 172.16.Y1.255)***

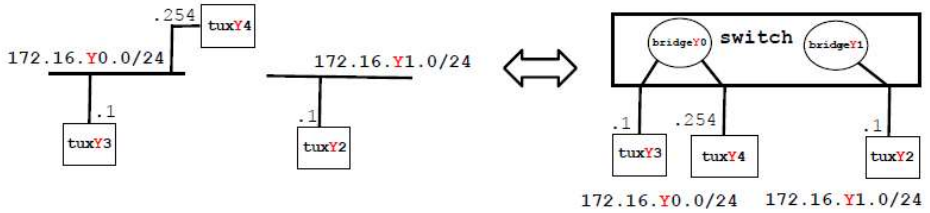
Nós (os três tux sff)

- Disable ICMP echo-ignore-broadcasts on tux13, tux14 and tux12

echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

U.PORTO

Part 2 / Exp 2 – Implement two bridges in a switch

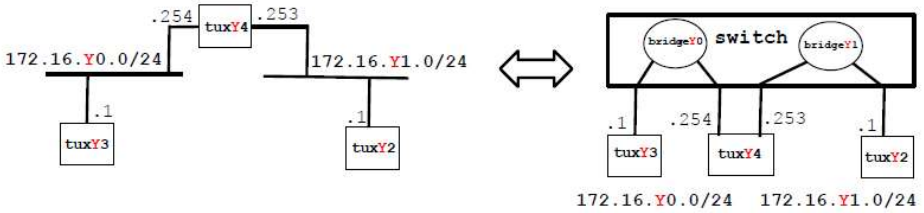


Questions

- How to configure bridgeY0?
- How many broadcast domains are there? How can you conclude it from the logs?

U.PORTO

Part 2 / Exp 3 – Configure a Router in Linux



Steps

1. Transform tuxY4 (Linux) into a router
   - Configure also tuxY4.eth1 and add it to bridgeY1
   - Enable IP forwarding
   - Disable ICMP echo-ignore-broadcast
2. Observe MAC addresses and IP addresses in tuxY4.eth0 and tuxY4.eth1
3. Reconfigure tuxY3 and tuxY2 so that each of them can reach the other
4. Observe the routes available at the 3 tuxes (***route –n***)
5. Start capture at tuxY3
6. 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
7. Stop the capture and save the logs

Nós

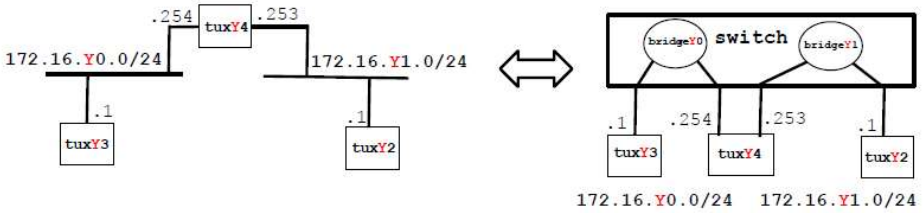
- Bla, bla, bla…
- Enable IP forwarding on tux14

echo 1 > /proc/sys/net/ipv4/ip-forward

- *Bla*

U.PORTO

Part 2 / Exp 3 – Configure a Router in Linux

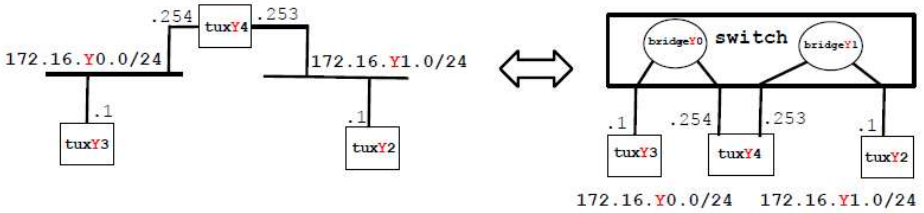


Steps

1. Transform tuxY4 (Linux) into a router
   - Configure also tuxY4.eth1 and add it to bridgeY1
   - Enable IP forwarding
   - Disable ICMP echo-ignore-broadcast
2. Observe MAC addresses and IP addresses in tuxY4.eth0 and tuxY4.eth1 ← Vocês
3. Reconfigure tuxY3 and tuxY2 so that each of them can reach the other
4. Observe the routes available at the 3 tuxes (***route –n***)
5. Start capture at tuxY3
6. 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
7. Stop the capture and save the logs

U.PORTO

Part 2 / Exp 3 – Configure a Router in Linux



Steps

1. Transform tuxY4 (Linux) into a router
   - Configure also tuxY4.eth1 and add it to bridgeY1
   - Enable IP forwarding
   - Disable ICMP echo-ignore-broadcast
2. Observe MAC addresses and IP addresses in tuxY4.eth0 and tuxY4.eth1
3. Reconfigure tuxY3 and tuxY2 so that each of them can reach the other
4. Observe the routes available at the 3 tuxes (***route –n***)
5. Start capture at tuxY3
6. 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
7. Stop the capture and save the logs

- Add route to 172.16.11.0/24 on tux13 (route –n)

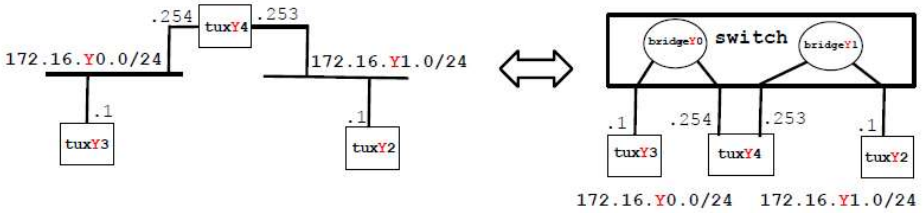
route add default gw 172.16.Y0.254

- Add route to 172.16.10.0/24 on tux12 (route –n)

route add default gw 172.16.Y1.253

U.PORTO

Part 2 / Exp 3 – Configure a Router in Linux



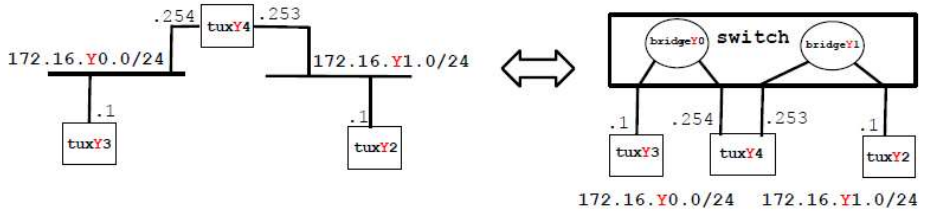
Steps

1. Transform tuxY4 (Linux) into a router
   - Configure also tuxY4.eth1 and add it to bridgeY1
   - Enable IP forwarding
   - Disable ICMP echo-ignore-broadcast
2. Observe MAC addresses and IP addresses in tuxY4.eth0 and tuxY4.eth1
3. Reconfigure tuxY3 and tuxY2 so that each of them can reach the other
4. Observe the routes available at the 3 tuxes (***route –n***)
5. Start capture at tuxY3
6. 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
7. Stop the capture and save the logs

← Vocês

U.PORTO

Part 2 / Exp 3 – Configure a Router in Linux



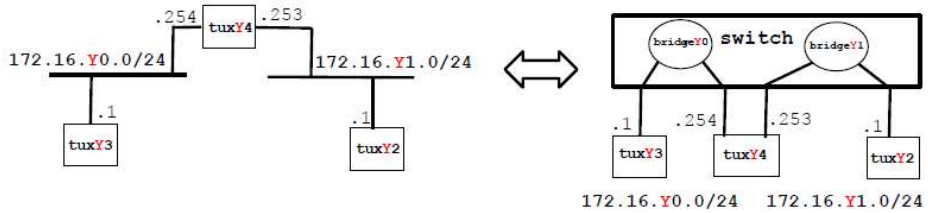
Steps

8. Start capture in tuxY4; use 2 instances of Wireshark, one per network interface
9. Clean the ARP tables in the 3 tuxes
10. In tuxY3, ping tuxY2 for a few seconds.
11. Stop captures in tuxY4 and save logs

← Vocês

U.PORTO

Part 2 / Exp 3 – Configure a Router in Linux

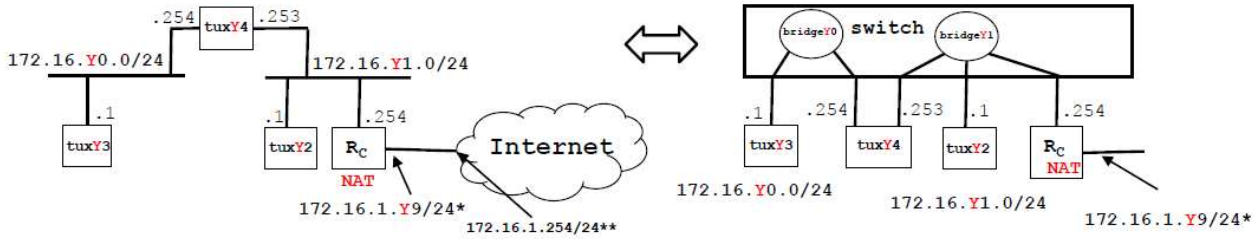


Questions

- What routes are there in the tuxes? What are their meaning?
- What information does an entry of the forwarding table contain?
- What ARP messages, and associated MAC addresses, are observed and why?
- What ICMP packets are observed and why?
- What are the IP and MAC addresses associated to ICMP packets and why?

U.PORTO

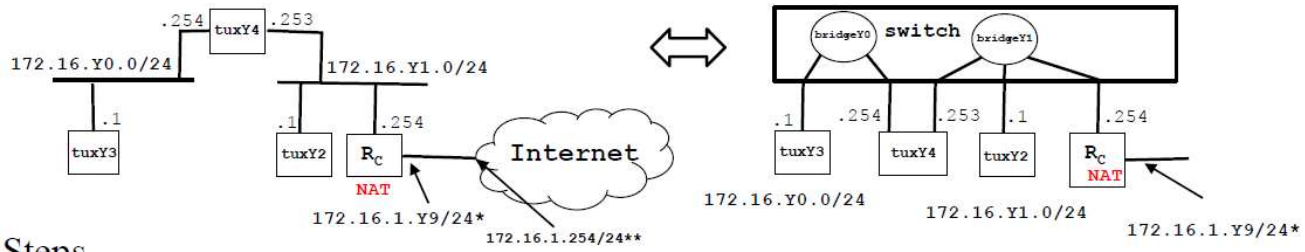
Part 2 / Exp 4 –
Configure a Commercial Router and Implement NAT



Steps

1. Connect $\mathbf{R_C}$ to the *lab network* (**with NAT enabled by default**) and configure ether1 and ether 2 interfaces
2. Verify routes
   - tuxY4 as **default router** of tuxY3;
   - $\mathbf{R_C}$ as **default router** for tuxY2 and tuxY4
   - in tuxY2 and $\mathbf{R_C}$ add routes for 172.16.Y0.0/24
3. Using **ping** commands and **Wireshark**, verify if tuxY3 can ping all the network interfaces of tuxY2, tuxY4 and $\mathbf{R_C}$

Part 2 / Exp 4 –
Configure a Commercial Router and Implement NAT

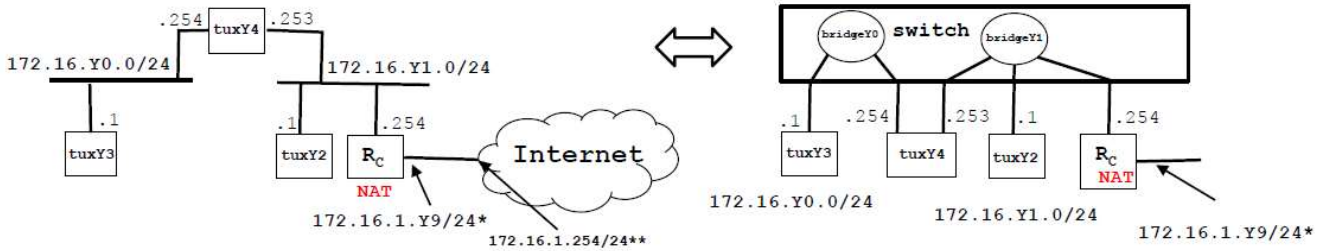


Steps

4. In tuxY2
   - Do the following:
     - *echo 0 > /proc/sys/net/ipv4/conf/eth0/accept_redirects*
     - *echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects*
   - 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

14

Part 2 / Exp 4 –
Configure a Commercial Router and Implement NAT

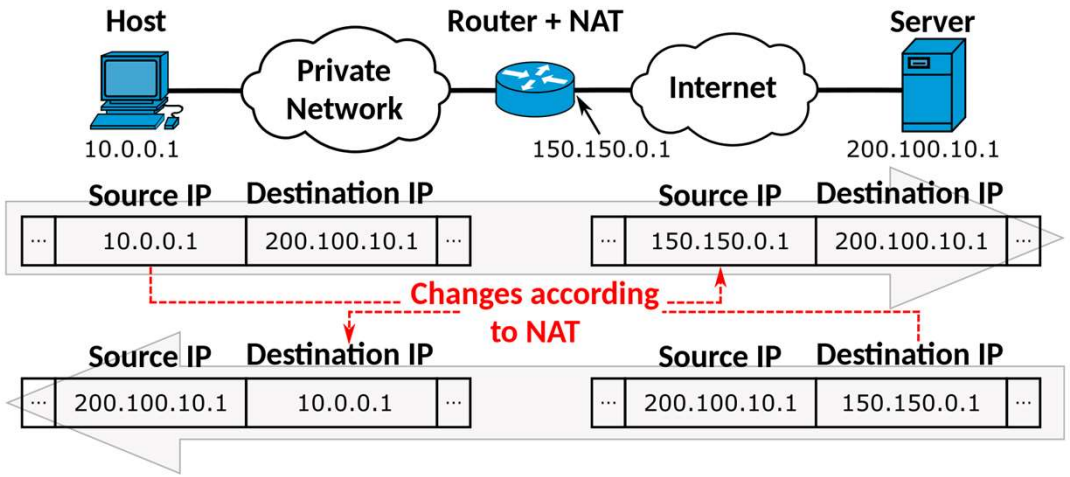


Steps

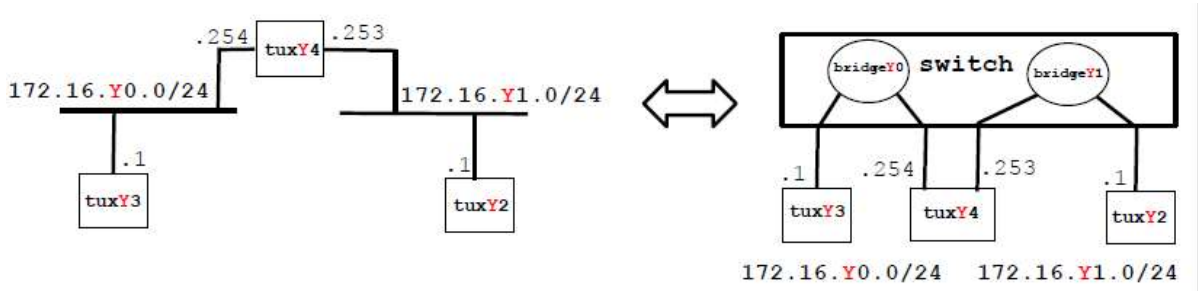
5. In tuxY3, ping the router of the lab I.321 (172.16.1.254)** and try to understand what happens
6. Disable **NAT** functionality to router $\mathbf{R_C}$
7. In tuxY3 ping 172.16.1.254**, verify if there is connectivity, and try to understand what happens

- What is NAT?

Network address translation (NAT) is a method of mapping an IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device.



- Vocês têm de configurar os computadores e o switch para ficarem com a rede da imagem (Part 2 – Exp 3)



- WARNING: Não esqueçam

Reset ethernet configuration

systemctl restart networking

Reset switch and router configuration

/system reset-configuration

Parte 2 – Experiência 4: Configurar um router comercial e implementar NAT

U.PORTO

1. Connect ether1 of Rc to the lab network on PY.1 (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

- Connect eth1 of the router to the netlab network (PY.1)
- Configure the IP address through console

(router) -> */ip address add address=172.16.2.Y9/24 interface=ether1*

- Connect eth2 of the router to a port on bridgeY1 and configure the address

(switch) -> /interface bridge port remove [find interface=etherXX]

(switch) -> */interface bridge port add interface=etherXX bridge=bridgeY1*

(switch) -> */interface bridge port print brief*

/ip address add address=172.16.Y1.254/24 interface=ether2

- Check IPs

(router) -> */ip address print*

2. Verify routes

- tuxY4 as default router of tuxY3 -> Vocês têm de configurar/confirmar isto

route add default gw 172.16.Y0.254

- Rc as default router for tuxY2 and tuxY4

route add default gw 172.16.Y1.254

- In tuxY2 and Rc add routes for 172.16.Y0.0/24

ip route add 172.16.Y0.0/24 via 172.16.Y1.253

route –n

(router) -> */ip route add dst-address=0.0.0.0/0 gateway=172.16.2.254* (ONLY FOR LAB I320)
(router) -> */ip route add dst-address=172.16.Y0.0/24 gateway=172.16.Y1.253*
(router) -> */ip route print*

3. Using ping commands and Wireshark , verify if tuxY3 can ping all the network interfaces of tuxY2 , tuxY4 and Rc

ping 172.16.Y0.254
ping 172.16.Y1.253
ping 172.16.Y1.1
ping 172.16.Y1.254
ping 172.16.2.Y9
ping 8.8.8.8

- Let's verify the routes

traceroute 172.16.Y0.254
traceroute 172.16.Y1.253
traceroute 172.16.Y1.1
traceroute 172.16.Y1.254
traceroute 172.16.2.Y9
traceroute 8.8.8.8

5. Disable NAT functionality to router Rc

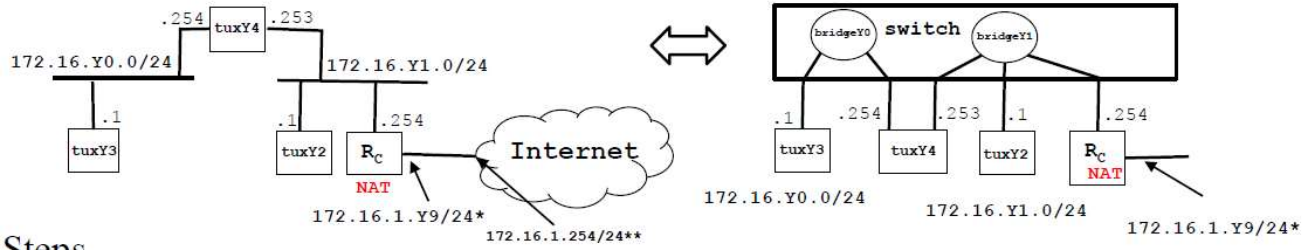
/ip firewall nat disable 0

- Run wireshark

ping 172.16.Y0.254
ping 172.16.Y1.253
ping 172.16.Y1.1
ping 172.16.Y1.254
ping 172.16.2.Y9
ping 8.8.8.8

- Do traceroutes

Part 2 / Exp 4 –
Configure a Commercial Router and Implement NAT



Steps

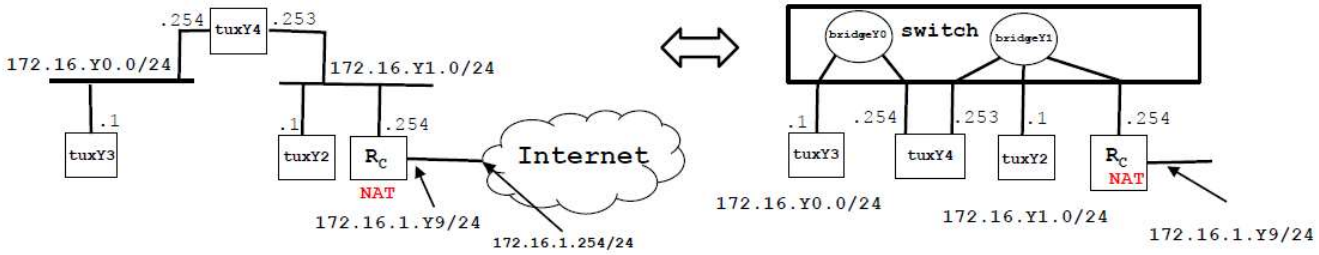
4. In tuxY2
   - Do the following:
     - *echo 0 > /proc/sys/net/ipv4/conf/eth0/accept_redirects*
     - *echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects*
   - remove the route to 172.16.Y0.0/24 via tuxY4
   - In tuxY2, ping tuxY3
   - Using capture at tuxY2, try to understand the path followed 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

14

Do it yourself

U.PORTO

Part 2 / Exp 4 –
Configure a Commercial Router and Implement NAT

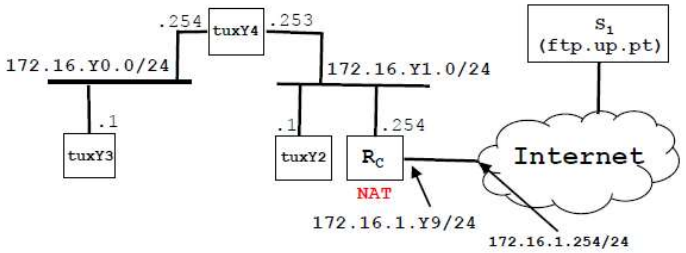


Questions

- How to configure a static route in a commercial router?
- What are the paths followed by the packets in the experiments carried out and why?
- How to configure NAT in a commercial router?
- What does NAT do?

U.PORTO

Part 2 / Exp 5 – DNS



Steps

1. Configure DNS at tuxY3, tuxY4, tuxY2
   (use DNS server *services.netlab.fe.up.pt (172.16.1.1*)*)
2. Verify if names can be used in these hosts (e.g **ping** hostname, use browser)
3. Execute **ping (new-hostname-in-the-Internet)**; observe DNS related packets in **Wireshark**

Questions

» How to configure the DNS service at an host?
» What packets are exchanged by DNS and what information is transported

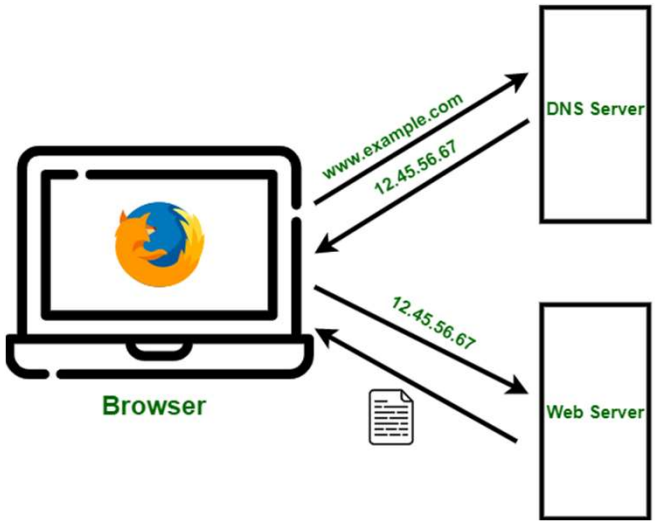
* In room I320 the DNS server address is 172.16.2.1

Parte 2 – Experiência 5: DNS

- What is DNS?

  The Domain Name System (DNS) is a hierarchical and distributed naming system for computers, services, and other resources in the Internet or other Internet Protocol (IP) networks.

  The Domain Name System (DNS) is the phonebook of the Internet…. DNS translates domain names to IP addresses so browsers can load Internet resources.



Parte 2 – Experiência 5: DNS

1. Configure DNS at tuxY3 , tuxY4 , tuxY2

- Open the resolv.con file

sudo nano /etc/resolv.conf

- Add the server

nameserver 172.16.2.1 (ONLY FOR LAB I320)

2. Verify if names can be used in these hosts

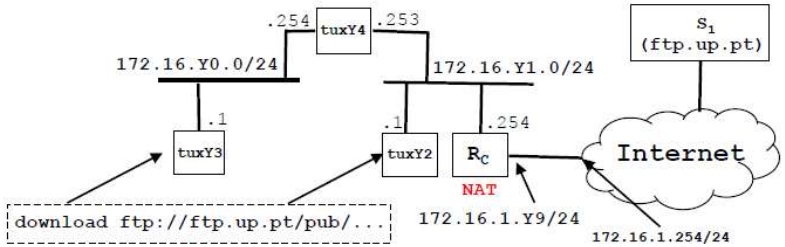
ping www.google.com
ping www.ojogo.pt
Open browser and open www.fcporto.pt

3. Execute ping (new hostname on the Internet); observe DNS related packets in Wireshark

Open Wireshark
ping www.nottingham.ac.uk

U.PORTO

Part 2 / Exp 6 – TCP connections



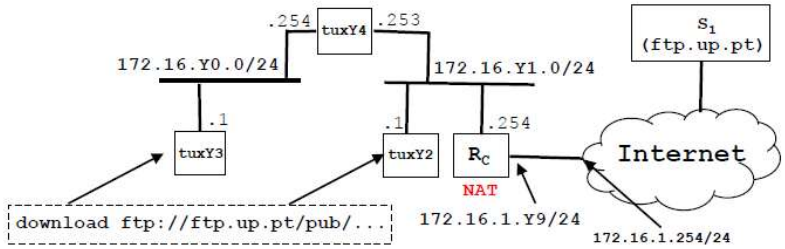
Steps

1. Compile your dowoload application in tuxY3
2. In tuxY3, restart capturing with Wireshark and run your
3. Verify if file has arrived correctly, stop capturing and save the log

U.PORTO

Part 2 / Exp 6 – TCP connections

Steps



4. Using **Wireshark** observe packets exchanged including:
   - TCP control and data connections, and its phases (establishment, data, termination)
   - Data transferred through the FTP control connection
   - TCP ARQ mechanism
   - TCP congestion control mechanism in action
   - Note: use also Wireshark Statistics tools (menu) to study TCP phases, ARQ and congestion control mechanism
5. Repeat download in tuxY3 but now, in the middle of the transference, start a new download in tuxY2
   - Use the Wireshark statistics tools to understand how the throughput of a TCP connection varies along the time

U.PORTO

Part 2 / Exp 6 – TCP connections

Questions

- How many TCP connections are opened by your ftp application?
- In what connection is transported the FTP control information?
- What are the phases of a TCP connection?
- How does the ARQ TCP mechanism work? What are the relevant TCP fields? What relevant information can be observed in the logs?
- How does the TCP congestion control mechanism work? What are the relevant fields. How did the throughput of the data connection evolve along the time? Is it according the TCP congestion control mechanism?
- Is the throughput of a TCP data connections disturbed by the appearance of a second TCP connection? How?