universidade de aveiro
theoria poiesis praxis

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA

MESTRADO INTEGRADO EM ENG. DE COMPUTADORES E TELEMÁTICA

ANO 2018/2019

FUNDAMENTOS DE REDES

LABORATORY GUIDE NO. 1

Objectives

- Familiarization with equipment configuration
- Familiarization with *Wireshark* protocol analyser
- Ethernet technology (hubs and switches)
- IP protocol (addressing, forwarding, fragmentation and reassembly)
- IP Address Resolution Protocol
- ICMP (*ping*, *arp* and *tracert* commands)
- Switching and Routing.

Duration

- 4 weeks

1. Evaluation

- This guide will be evaluated by a written test composed by multiple choice questions.
- In order to prepare for the written test, students should make during all laboratory classes an own report with the conclusions taken on all experiments including the captures showing all results supporting them.

2. Additional documents necessary for this guide

- "Configuration Commands of CISCO Router".
- "D-Link Switch Manual".
- "D-Link CLI Reference Manual".

3. Mandatory actions at the end of each class

At the end of each class, the network equipment must be reset to its default configuration before switching them off.

☞ Reset to default configuration of CISCO Router

To reset the CISCO Routers to its default configuration, first, run the following command:

```
router#write erase
```

and, then, switch off the Router.

☞ Reset to default configuration of D-Link Switch

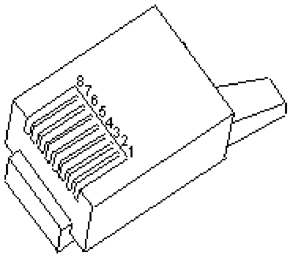
To reset the D-Link Switch to its default configuration, first, run the following command:

```
DES-3026:4#reset config
```

and, then, switch off the Switch.

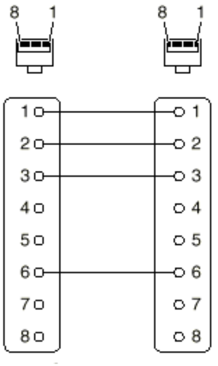
10BASET Standard Cables

On the 10BASET standard, cables are terminated in both sides with RJ-45 connectors:



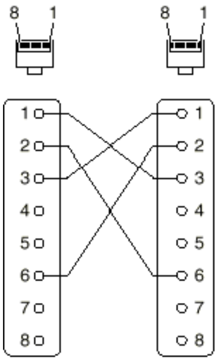
The standard specifies the following connections for direct cables:

1	White / Orange
2	Orange
3	White / Green
4	Blue
5	White / Blue
6	Green
7	White / Brown
8	Brown



and for crossover cables:

1	White / Green
2	Green
3	White / Orange
4	Blue
5	White / Blue
6	Orange
7	White / Brown
8	Brown



The following table specifies the cable type to be used between each pair of equipment types:

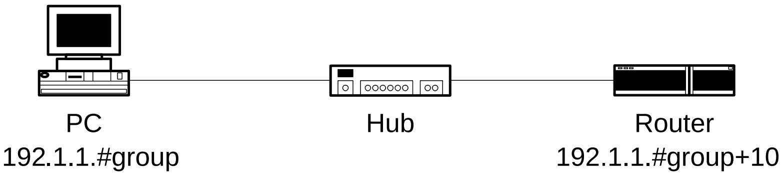
	PC	***Router***	***Switch***	***Hub***
PC	Crossover	Crossover	Direct	Direct
Router	Crossover	Crossover	Direct	Direct
Switch	Direct	Direct	Crossover	Crossover
Hub	Direct	Direct	Crossover	Crossover

NOTE: Recent switch equipment models have an auto-detect mode which let them work with only direct cables. In the lab, you need to use crossover cables only when you use the older switch models.

4. Experiments with hubs

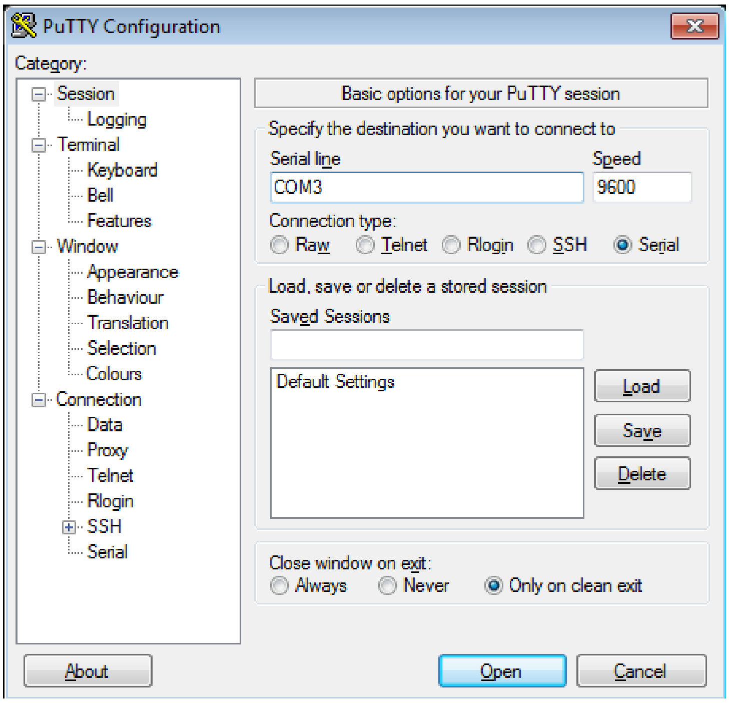
~~1.~~ Set-up the following 10BASET Ethernet, based on a single hub with two terminals (the Router is used only as a terminal). Configure the IP addresses following the specification of the figure where #group specifies your group number. Test the connectivity using *ping* command between the PC and the Router.

Consider the IP network address 192.1.1.0 of class C (subnet mask 255.255.255.0) where the host part of the address depends on the group number in the following way:



☞ Router configuration

To configure the Router, you must connect its console port to the PC serial port (with the appropriate cable) and use the *Putty* application. For that, you will have to select the right connection type (RAW, Telnet, Rlogin, SSH or Serial) and set the speed to 9600, as illustrated in the following figure.



Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM3

Speed: 9600

Connection type:
○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

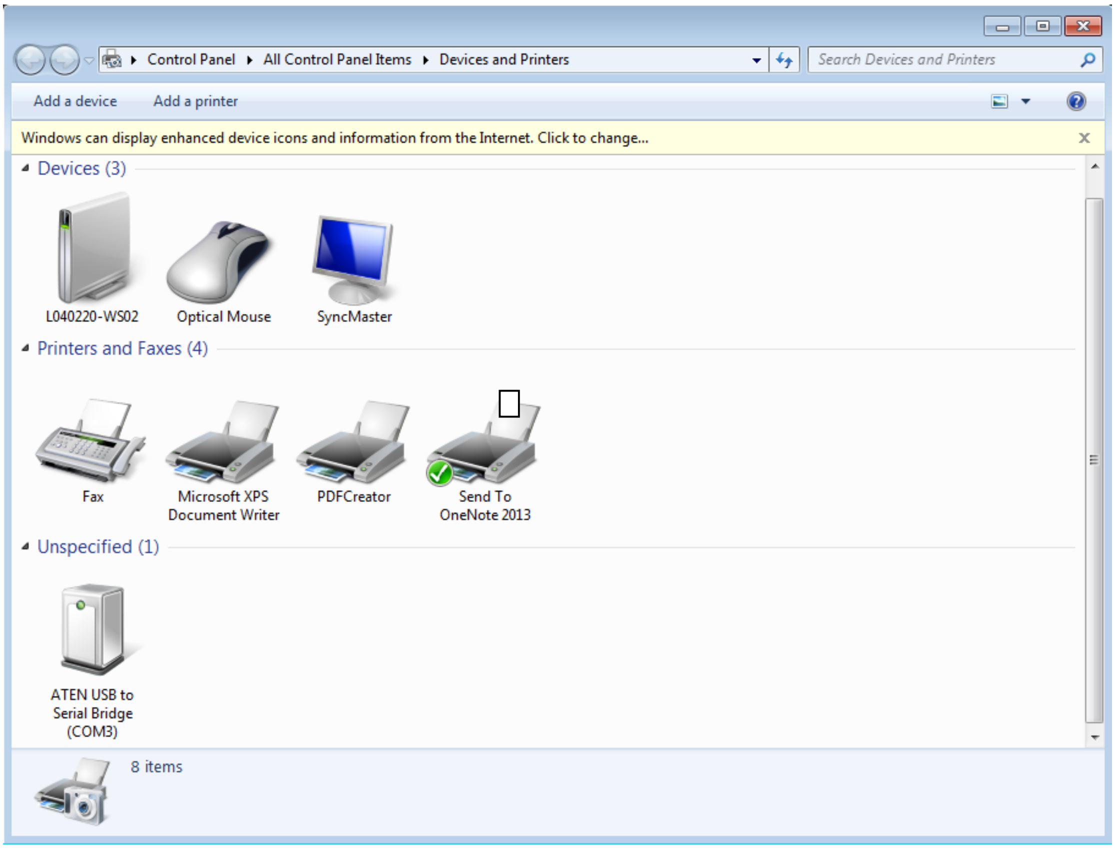
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:
○ Always ○ Never ● Only on clean exit

You will have to be careful when selecting the serial line. For you to know which COM should be used, search the environment *Devices and Printers* in your PC (Windows environment) and check which COM is active (for example, COM3 in the following figure).



Windows can display enhanced device icons and information from the Internet. Click to change...

Switch on the Router. After a while, the Router prompt will appear:

```
router>
```

To configure the IP address of the Router interface (assuming its name is `ethernet0`), execute the following commands (the following example refers to Group 1):

```
router>enable
router#
router#configure terminal
router(config)#
router(config)#interface ethernet0
router(config-if)#ip address 192.1.1.11 255.255.255.0
router(config-if)#no shutdown
router(config-if)#end
router#write
Building configuration...
[OK]
router#
```

Error while executing `interface ethernet0`? Why? How can I find out the right name of the interface?

☞ **Execution of command *ping***

At the Router:

```
router#ping 192.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4
ms
router#
```

At the PC:

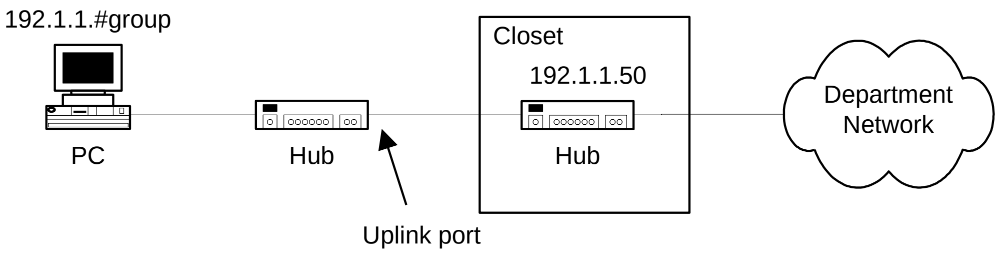
```
C:\ping 192.1.1.11

Pinging 192.1.1.11 with 32 bytes of data:

Reply from 192.1.1.11: bytes=32 time<10ms TTL=255
Reply from 192.1.1.11: bytes=32 time<10ms TTL=255
Reply from 192.1.1.11: bytes=32 time<10ms TTL=255
Reply from 192.1.1.11: bytes=32 time<10ms TTL=255

Ping statistics for 192.1.1.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

~~2.~~ Replace the connection from your hub to the Router by a connection to the lab closet hub as specified in the figure (this hub has been connected to the DETI network). Run the command *ping –t* for the closet hub address 192.1.1.50 and do not stop it until it is requested.



☞ ***ping* command**

```
ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
           [-r count] [-s count] [[-j host-list] | [-k host-list]]
           [-w timeout] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
```

```
-l size        Send buffer size.
-f             Set Don't Fragment flag in packet.
-i TTL         Time To Live.
-v TOS         Type Of Service.
-r count       Record route for count hops.
-s count       Timestamp for count hops.
-j host-list   Loose source route along host-list.
-k host-list   Strict source route along host-list.
-w timeout     Timeout in milliseconds to wait for each reply.
```

~~2.1.~~ Run *Wireshark* and start a capture of all packets. Run the *Statistics* ➔ *Endpoints* tool and verify that your PC captures packets from/to other terminals, which illustrates how hubs work.

~~2.2.~~ Run the *Statistics* ➔ *Conversations* tool to visualise the communications among the different pairs of hosts.

~~2.3.~~ Analyse the packets that are being captured. Verify that the capture shows packets of different protocols and different hosts (once again illustrating how hubs work). Apply a display filter for ICMP packets and verify that the capture shows only packets of this protocol. Then, apply a display filter for ICMP packets to/from your PC address. Verify that, as before, the capture shows only packets following this new filter rule. Register the filter definitions used for each case. Stop the capture of packets and abort the command *ping –t* 192.1.1.50.

~~3.~~ Set-up again the network used in experiment 1. Start a capture with *Wireshark*. Execute the command *ping* from the PC to the Router. After the end of the *ping* execution, terminate the capture and save it with .cap extension.

~~3.1.~~ Analyse the saved capture. What do you conclude on the ICMP packet periodicity? Observe how the *Sequence Number* field of ICMP packets is used for round-trip-time (RTT) estimation done by the *ping* command.

~~3.2.~~ Observe now in the saved capture the different encapsulation levels: the ICMP packets are encapsulated on IP datagrams and the IP datagrams are encapsulated on Ethernet frames. Register the following information:

- PC Ethernet address: d0:17:c2:93:74:aa
- Router Ethernet address: 00:13:c3:d2:8e:d0
- Hexadecimal code (*Type* field of Ethernet header) that identifies an IP datagram: 0x0800
- Hexadecimal code (*Protocol* field of IP header) that identifies an ICMP packet: 0x01
- Hexadecimal code (*Type* field of ICMP header) that identifies the two ICMP packet types (*Echo Request* and *Echo Reply*): 0x08 (Echo request), 0x00 (Echo reply)

4. On a DOS window of your PC, first execute the command *arp –d* to delete all ARP table entries of your PC. Then, run the *ping* command to the Router. Finally, run the command *arp –a* to display the ARP table of your PC. Check that the IP address of the Router has an associated Ethernet address.

The router with the IP address 192.1.1.12 has a MAC address 00:22:55:18:a3:f8

***arp* command**

```
arp -d inet_addr [if_addr]
arp -a [inet_addr] [-N if_addr]

  -a         Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and
                Physical addresses for only the specified computer are
                displayed. If more than one network interface uses ARP,
                entries for each ARP table are displayed.
   inet_addr    Specifies an internet address.
  -d        Deletes the host specified by inet_addr. inet_addr may
                be wildcarded with * to delete all hosts.
```

5. Start a new capture with *Wireshark*. Repeat experiment 4 and, then, stop the capture. Analysing the captured packets, explain how ARP protocol is used by the PC to discover the Ethernet address of the Router before exchanging the ICMP packets. Register the following information of the captured ARP packets:

ARP Request

Ethernet header
Origin address: d0:17:c2:93:74:aa
Destination Address: ff:ff:ff:ff:ff:ff ←
ARP packet
Origin MAC address: d0:17:c2:93:74:aa
Origin IP address: 192.1.1.2
Destination MAC address: 00:00:00:00:00:00 ←
Destination IP address: 192.1.1.12

ARP Response

Ethernet header
Origin address: 00:22:55:18:a3:f8
Destination Address: d0:17:c2:93:74:aa
ARP packet
Origin MAC address: 00:22:55:18:a3:f8
Origin IP address: 192.1.1.12
Destination MAC address: d0:17:c2:93:74:aa
Destination IP address: 192.1.1.2

6. On your PC, run the command *ping* to the Router. Then, estimate how long it takes the Router entry to disappear from the ARP table (if you need, use the Windows *Clock* applications). Remember from the theoretical classes the reasons for the fact that these ARP table entries are not permanent.

It would take 5 minutes! Way too long

7. In order to work properly, Ethernet requires a minimum size data field of 46 bytes. If the protocol running on top of Ethernet delivers a chunk of less than 46 bytes, Ethernet adds dummy bytes to guarantee its minimum size (this process is named *padding*). On a DOS window of your PC, execute the command *arp –d* to delete all ARP table entries of your PC. Start a new capture with *Wireshark*. Then, execute the command *ping –l 5* to the Router and stop the capture. Observe the padding process on the captured ARP and ICMP packets.

NOTE: *Wireshark* does not show the padding bytes in packets generated on its host; therefore, the padding process can be observed only in the packets received by the PC.

8. Run again experiment 3 to save a new capture from the PC to the Router. With the *Colasoft Packet Builder* application, import the saved capture.

8.1. Start a new capture with *Wireshark*. In *Colasoft Packet Builder* application, send the first ICMP *Echo Request* one single time to the network. Stop the capture, register and justify the captured packets.

8. Start a new capture with *Wireshark*. In *Colasoft Packet Builder* application, first, change to 192.1.1.20 the origin IP address of the first ICMP *Echo Request* and, then, send it one single time to the network. Register and justify the captured packets.

9. IP protocol includes a *fragmentation and reassembly* mechanism in order to transmit IP packets whose size is larger than the MTU (Maximum Transmission Unit) of the network (Ethernet MTU = 1500 bytes). Start a new capture with *Wireshark*. Execute on your PC the command *ping –l 3000* to the Router. Save the capture with .cap extension to be used in the following experiments. Analyse the captured packets and explain the fragmentation process. In particular, explain:

- why each packet is fragmented in 3 fragments;
- the content of the IP header fields that enable the recovery of the complete packet at the destination; Flags, Fragment Offset and Identification
- the packet size of each fragment.

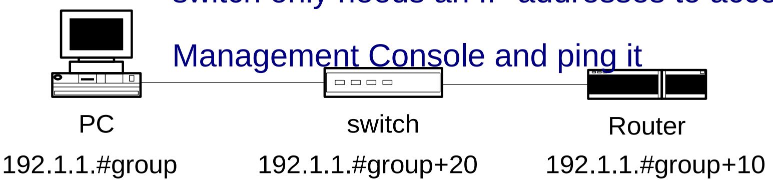
10.1. Start a new capture with *Wireshark*. With the *Colasoft Packet Builder* application, import the capture previously saved in experiment 9. Send the first two fragments of an ICMP *Echo Request* packet one single time to the network. Wait 2 minutes before stopping the capture. Analyse and justify the captured packets. Observe that the Router waits for some time to receive all fragments and discards the received fragments if at least one is missing.

10.2. Start a new capture with *Wireshark*. With the *Colasoft Packet Builder* application, send to the network the third fragment of an ICMP *Echo Request* packet and, after 10 seconds, send the first two fragments of the same ICMP *Echo Request* packet. Analyse and justify the captured packets. Observe that, although the fragments were not received in order, the Router has accepted all fragments, recovering the original ICMP packet and reacting accordingly.

5. Experiments with switches

11. Replace the hub by a D-Link Ethernet Switch to interconnecting the PC and the Router. Configure all equipment with the IP addresses specified in the figure below where #group specifies your group number and considering them as class C addresses (subnet mask 255.255.255.0). Test the connectivity between all equipment using the *ping* command.

switch only needs an IP addresses to access its Web Management Console and ping it



12. Execute again the *ping* command between PC and Router. Access the management console of the Switch using the *Web Browser*. Analyse the *MAC Address Table* of the Switch and register its contents (MAC address and Ethernet address are equivalent terms). Observe that the Switch has learned on each port the MAC addresses of the equipment connected to the same port. Confirm on the PC and on the Router that their MAC address are the ones learned by the Switch.00-13-c3-8f-1d-66 : Port 1
d0-17-c2-93-74-aa : Port 5

13. Each entry of the *MAC Address Table* has a lifetime value that is set to zero whenever the Switch receives an incoming packet on the same input port with the same origin MAC address. During time, if an entry lifetime reaches the *Aging Time* value, the entry is eliminated (the *Aging Time* value can be configured on the Switch). Using the *Web Browser* access, check the default *Aging Time* value of the Switch10 seconds

13.1. Using the *Web Browser* access, configure an *Aging Time* value of 10 seconds. Then, wait for about 20 seconds and check if the PC MAC address entry has disappeared from the *MAC Address Table*. Observe that, apparently, this entry does not disappear.

NOTE: The Router MAC address does not disappear from the *MAC Address Table* due to the fact that routers send periodically (from 10 to 10 sec.) a LOOPBACK packet to check for physical connectivity; these packets are continuously validating the Router MAC address on the Switch.

13.2. Close the *Web Browser* and connect to the management console of the Switch through its console (using the *HyperTerminal* application, as explained for the case of the Router). Examine again the *MAC Address Table*. Check that, in this experiment, the PC MAC address disappears from the table. Justify the different behaviour observed in these two experiments (13.1 and 13.2).

In 13.2 we are no longer sending and receiving packets between the PC and the switch (we closed the web page) thus the entry port 5 -> PC MAC address reaches its aging time and disappear. The entry port 1 -> router disapear from time to time since the loopback packets ocurr less often than the aging time

14. Remember from the theoretical classes that, when a Switch receives a packet on an incoming port, it searches for an entry with the packet destination MAC address on its *MAC Address Table*. Then, the behaviour of the Switch is one of two possibilities:

Flooding process: no such entry exists and the Switch sends the packet to all its ports, except the incoming port.

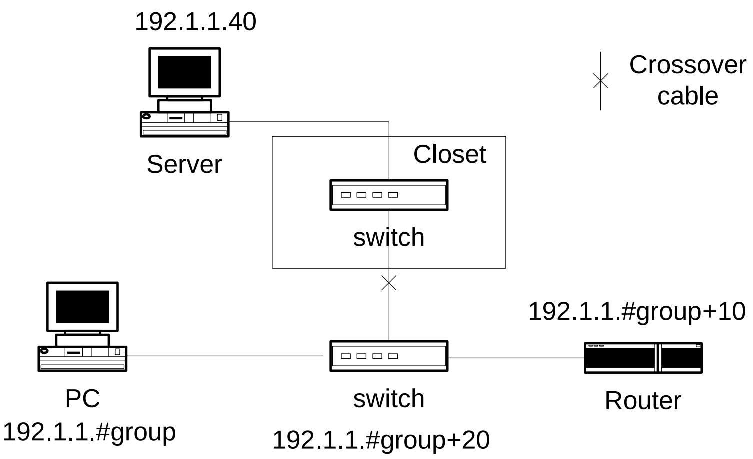
Forwarding process: the entry exists and the Switch sends the packet only for the port specified on the *MAC Address Table* entry, if it is not the incoming port.

The aim of the 2 next experiments is to verify the Switch basic *flooding* and *forwarding* processes.

14.1. Add to your network a connection (using a crossover cable) between your Switch and the Switch located in the lab closet (this Switch is connected to a Server host with the IP address depicted in the figure below). Test the connectivity by executing a *ping* command from the Router to the Server.

PING sends 5 ICMP Packets

x2 (Request and Reply)!!!



14.2. With *WireShark*, start a capture with a display filter for ICMP packets. Execute once again the *ping* command from the *router* to the Server. Register the captured packets. Note that the *ping* command has generated the exchange of 5 ICMP *Echo Request* and 5 ICMP *Echo Reply* packets between the Router and the Server. Nevertheless, the capture run on the PC has only one ICMP *Echo Request* packet. Explain these observations based on the Switch *flooding* and *forwarding* processes.

Only 1 packet is seen (because of flooding). Remember the Wireshark

is running on the PC and the router is the one where the ping is runs

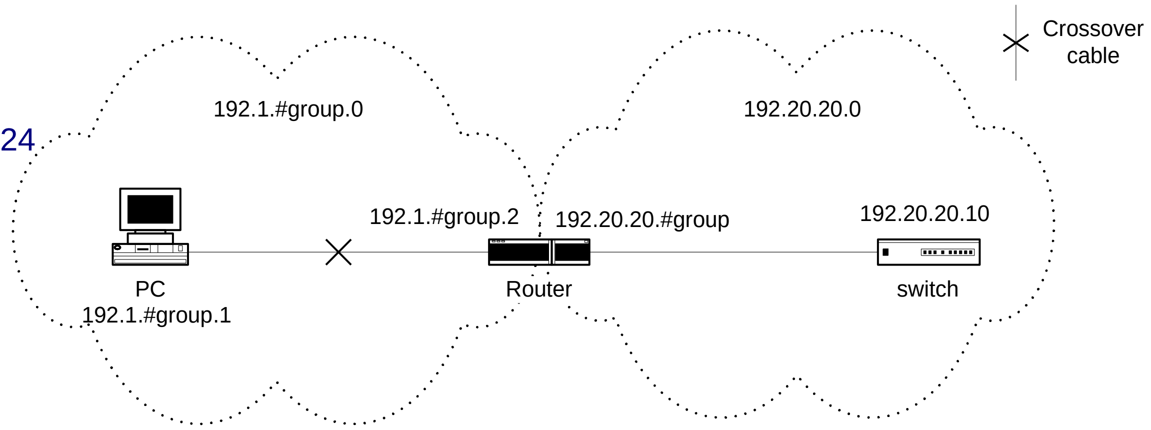
6. Experiments with routers

15. Configure two IP networks, interconnected by a Router, with the class C IP addresses specified in the figure below. Do not configure any *Default Gateway*. Register and justify the routing table of the Router.

show ip route

C : 192.20.20.0 /24

C : 192.1.2.0/24



16.1. Start a capture with a display filter for ICMP packets. Execute the *ping* command from the PC to the Switch. Repeat the experiment but now executing the *ping* command from the Switch (through its console) to the PC. Register and justify both the *ping* command results and the captured packets.

16.2. Configure the appropriate *Default Gateway* at the Switch. Start a new capture with a display filter for ICMP packets and execute the *ping* command from the Switch to the PC. Register and justify both the *ping* command result and the captured packets.

16.3. Configure the appropriate *Default Gateway* at your PC. Start a new capture with a display filter for ICMP packets and execute the *ping* command from the PC to the Switch. Register and justify the *ping* command result. Register also the following addresses of the ICMP *Echo Request* and *Echo Reply* packets and identify to which equipment interfaces each one of them belong.

ICMP Echo Request

Ethernet packet header	Source MAC Address: D0-17-C2-93-74-AA (PC) Destination MAC Address: 00-22-55-18-A3-F8 (Router)
IP packet header	Source IP Address: 192.1.2.1 (PC) Destination IP Address: 192.20.20.10 (Switch)

ICMP Echo Reply

Ethernet packet header	Source MAC Address: 00-22-55-18-A3-F8 (Router) NOT Switch!! Destination MAC Address: D0-17-C2-93-74-AA (PC)
IP packet header	Source IP Address: 192.20.20.10 Destination IP Address: 192.1.2.1

(seeing
from PC's
Point-of-
-view)

16.4. Remember from the theoretical classes that Routers forward IP packets based on the IP addresses of their IP headers (routers do not change the packet IP addresses). Nevertheless, routers are clients of each Ethernet segment. Therefore, the MAC addresses of the Ethernet header are specified with the MAC addresses of the communicating hosts on each Ethernet segment.

Having in mind this behaviour, and without making any capture, predict what were the following addresses of the ICMP packets exchanged between the Router and the Switch on the previous experiment (if needed, check the addresses on the equipment):

ICMP Echo Request

Ethernet packet header	Source MAC Address:	Router
	Destination MAC Address:	Switch (00-19-53-85-08-01)
IP packet header	Source IP Address:	PC
	Destination IP Address:	Switch

ICMP Echo Reply

Ethernet packet header	Source MAC Address:	Switch
	Destination MAC Address:	Router
IP packet header	Source IP Address:	Switch
	Destination IP Address:	PC

17. Register and justify the ARP tables of the Router. show arp

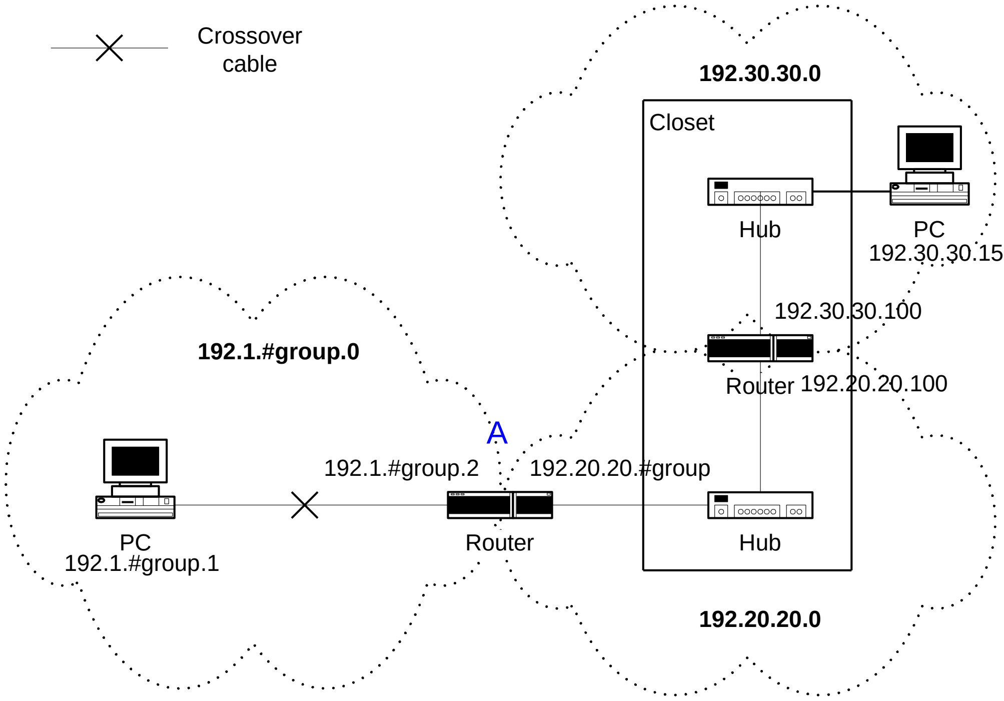
18.1. Start a new capture with a display filter for ICMP and ARP packets and execute the *ping* command from the *switch* to the IP address 192.1.#group.10 (an inexistent IP address of your network). Register the captured packets and explain the obtained results.

18.2. Start a capture with a display filter for ICMP and ARP packets. Execute the *ping* command from the Switch to the IP address 194.100.1.1. Register the captured packets. Justify the observed packets taking in mind that the Router has no entry for this IP address. What do you conclude about the difference between the Switch forwarding process (experiment 14.2) and the Router forwarding processes when the destination address is not known?

19. Substitute the connection from the Router to the Switch by a connection from the Router to the hub located in the lab closet (this hub is connected to another Router, as shown in the figure of the next page).

19.1. Register the routing table of your Router and compare it with the one of experiment 15. Observe that the routing table is the same, which means that the Router must be configured with something else (a routing protocol) to be able to reach the new IP network.

192.20.20.0/24 -> FastEthernet 0/1
192.1.2.0/24 -> FastEthernet 0/0



20.1. Start a capture with a display filter for ICMP packets. Execute the *ping* command from the PC to the IP address 192.20.20.150 (an inexistent address of an existing network). Register and justify the captured packets. Predict what has happened in this experiment in the other side of your Router (in the network 192.20.20.0) taking into consideration the results of experiment 18.1. ARP request is made but not answered but the PC can't see (unlike in 18.1) since it's not in the network where the request is done.

20.2. Start a capture with a display filter for ICMP packets. Execute the *ping* command from the PC to the IP address 192.30.30.100 (an existing address of a network that is not known yet by your Router). Register and justify the captured packets. Predict what has happened in this experiment in the other side of your Router (in the network 192.20.20.0) taking into consideration the results of experiment 18.2.

21. Configure a static route in your Router (the Router located at the lab closet has routes conveniently configured in both interfaces). Register the routing table of your Router. Observe that, now, the routing protocol enabled your Router to add information on its routing table concerning the new network. Then, execute the *ping* command from the PC to the IP address 192.30.30.150 to verify the connectivity between your PC and the new network.

Added line

192.30.30.0/24 via 192.20.20.100

to IP Route table of router A

It works now! :)

☞ Configuration of Static route in Cisco *routers*

In order to configure the static route, use the following commands (these IP addresses refer to the Group no. x):

```
Router#configure terminal
Router(config)#ip route 192.30.30.0 255.255.255.0 192.20.20.100
Define the path to network 192.30.30.0 through router 1
```

22. Start a capture with a display filter for ICMP packets. Then, run on your PC the following *ping* commands:

```
        ping -i 1 192.30.30.150
        ping -i 2 192.30.30.150
        ping -i 3 192.30.30.150
```

Based on the analysis of the captured packets for each case, explain the behaviour of the routers with the different TTL (Time-To-Live) values sent by the PC.

number of jumps

Analyzing the source IP of ICMP TTL exceeded packets, we can discover and list the routers in the routing path

23.1. The *tracert* command is a tool to discover the routers of the routing path from an origin IP host to a destination IP host. At your PC, start a capture with a display filter for ICMP packets and execute the command *tracert –d 192.30.30.150*. Based on the analysis of the captured packets, explain how *tracert* command works. In particular:

(i) identify how the PC identifies each router in the path;

(ii) observe that the PC sends three ICMP *Echo Request* packets for each growing value of TTL in order to obtain a better estimation of the round trip time;

(iii) determine how the PC stops the process.

30 steps (TTL = 30)
or finds the device

☞ **Options of the DOS *tracert* command**

```
C:\ >tracert

Usage: tracert [-d] [-h max_hops] [-j host-list] [-w timeout] target

Options:

   -d                         Do not resolve addresses to hostnames.
   -h maximum_hops            Maximum number of hops to search for target.
   -j host_list               Loose source route along host_list.
   -w timeout                 Wait timeout milliseconds for each reply.
```

23.2. Verify and justify the differences obtained when executing in your PC the command *tracert –d* for the IP addresses *192.30.30.150* and *192.30.30.100*.