universidade de aveiro
theoria poiesis praxis

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA

LICENCIATURA EM ENG. DE COMPUTADORES E INFORMÁTICA

REDES DE COMUNICAÇÕES

LABORATORY GUIDE NO. 2

Objectives

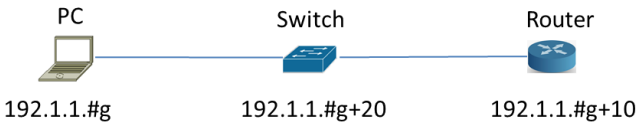
Physical Interfaces and Ethernet Addresses
IPv4 protocol (addressing, forwarding, fragmentation and reassembly)
IPv4 Address Resolution Protocol
ICMP (ping, arp and traceroute commands)
Familiarization with Wireshark protocol analyzer
Familiarization with equipment configuration
Ethernet technology (Switching)
Introduction to IP Routing
IP Subnetting

Duration

- 4 weeks

1.Initial Experiments

1. Build and configure the following network with the equipments in the lab (you can use your PC). Run the command *ping –t* (pings without stopping) for the router. All configurations are in the annex in the end of the document.



2. Run *Wireshark* in the PC and start a capture of all packets. Run the *Statistics* ➔ *Endpoints* tool and verify that the PC captures packets from/to another equipment.

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

4. Terminate the capture and save it with .cap extension.

5. 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.

6. 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:
- Router Ethernet address:
- Hexadecimal code (*Type* field of Ethernet header) that identifies an IP datagram:
- Hexadecimal code (*Protocol* field of IP header) that identifies an ICMP packet:
- Hexadecimal code (*Type* field of ICMP header) that identifies the two ICMP packet types (*Echo Request* and *Echo Reply*):

7. On a command 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.

***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.
```

8. 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:
- Destination Address:

ARP packet
- Origin MAC address:
- Origin IP address:
- Destination MAC address:
- Destination IP address:

ARP Response

Ethernet header
- Origin address:
- Destination Address:

ARP packet
- Origin MAC address:
- Origin IP address:
- Destination MAC address:
- Destination IP address:

9. 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.

10. 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.

<u>*NOTE*</u>: *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.

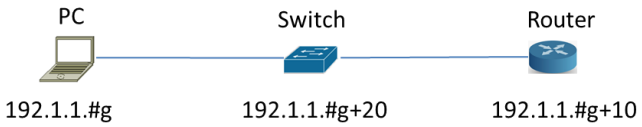
11. IP protocol includes a *fragmentation and reassembly* mechanism in order to transmit IP packets whose size is larger than the MTU (Maximum Transmission Unit)

of the network (Ethernet MTU = 1500 bytes). Start a new capture with *Wireshark*. Execute on your PC the following commands to the Router:
ping –l 2000 Router
ping –l 3100 Router
Repeat the ping commands from the Router do the PC using 2000 and 3000 bytes of data:
ping PC size 2028
ping PC size 3128
Analyze 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;
- the packet size of each fragment.

2. Experiments with switches

1. Consider the same network as before. Test the connectivity between all equipment using the *ping* command.



2. 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.

3. 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 Switch.

4. 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.

5. Close the *Web Browser* and connect to the management console of the Switch through its console (using the serial interface). 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 (4 and 5).

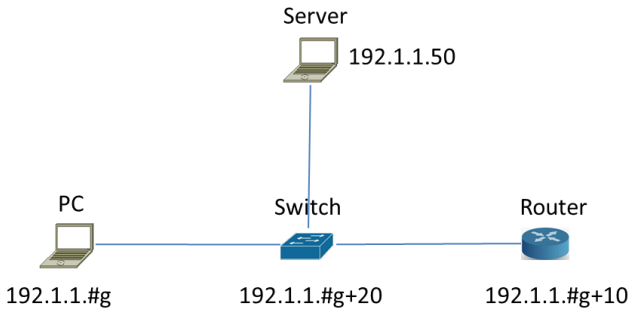
6. Remember from the theoretical classes that, when a Switch receives a packet on an incoming port, it searches for an entry with the packet destination MAC address on its *MAC Address Table*. Then, the behaviour of the Switch is one of two possibilities:

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.

7. Add to your network a connection to a Server (your PC in the lab or of your colleague) connected to the switch. Test the connectivity by executing a *ping* command from the Router to the Server.



8. 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.

3. Experiments with routers

…

4. IP Subnetting

…

Annex

5. PC, Switch and Router configuration

PC configuration:

Configuration of the PC in *Windows*

To configure the PC IP address, its Gateway, and even the DNS server, go to Settings and Network and Internet to the configuration pane.

Configuration of the PC in *Linux*

To configure the PC IP address, execute the following command:

```
ifconfig eth0 <IPaddress> netmask <IPmask>
```

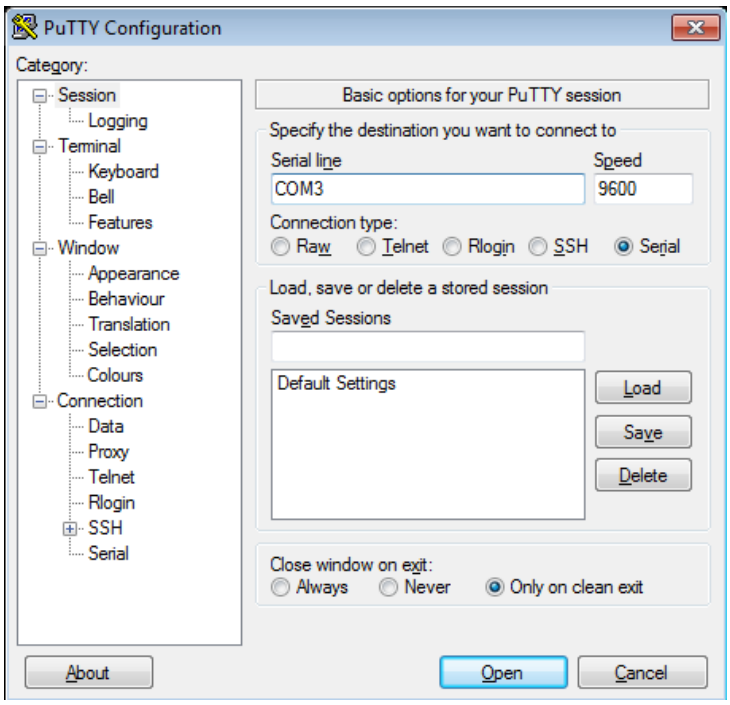
To configure the Gateway of the PC, execute the following command:

```
route add default gw <IPaddress>
```

The command `route -n` shows the routing table of the PC and the configured *Gateway*.

Connect the switch and router:

To configure the switch and router, connect its console port to the PC USB port (with the appropriate cable) and use, for example, 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. 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), or the linux dmesg command (although it is usual tty0) and check which COM is active. In the putty configuration below, COM3 is used.



Then, you can change your cable between the switch and the router without any problems.

Router configuration:

Connect the Router to the PC. 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
```

At the PC:

```
C:\ping 192.1.1.11
```

Switch configuration:

Connect the Switch to the PC. After a while, the Switch prompt will appear:

```
#
```

To configure the IP address of the Switch, execute the following command (the following example refers to Group 1):

```
#config ipif System ipaddress 192.1.1.21/24
#show ipif
```

To show the switching table of the switch:

```
#show fdb
```

To create a default gateway on the switch:

```
#create iproute default 192.1.1.11 1
#show iproute
```

☞ **Execution of command *ping***

```
#ping 192.1.1.1 times 4
```