universidade de aveiro
theoria poiesis praxis

# Fundamentos de Redes

## 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
- Introduction to GNS3
- Familiarization with equipment configuration
- Ethernet technology (Switching)
- Introduction to IP Routing

# Part A

## Physical Interfaces and Ethernet Addresses

1. Open a console/terminal on your on PC and type the following command:

```
(Windows)
    ipconfig /all
(Linux)
    ip link
```

>> Identify your network interfaces.

>> Identify the MAC address of each interface.

2. Based on the MAC address of your interfaces determine its manufacturer using the following websites:

[https://macvendorlookup.com/](https://macvendorlookup.com/)

[https://maclookup.app/](https://maclookup.app/)

## IPv4 Protocol

3. Connect your PC to any IP network and run the following commands in a console/terminal:

```
(Windows)
    ipconfig
    ipconfig /all
(Linux)
    ip addr
```

>> Identify the IPv4 address and network mask of each interface.

>> Identify the IPv6 address(es) and network mask(s) of each interface.

4. Run the following commands in a console/terminal:

```
(Windows)
    route print
(Linux)
    ip route
    ip -6 route
```

>> Identify the IPv4 gateway(s) of your PC.

>> Identify the IPv6 gateway(s) of your PC (you may not have any).

## IPv4 Address Resolution Protocol (ARP)

5. Visualize the ARP table in your PC by running the following commands in a console/terminal:

```
(Windows)
    arp -a
(Linux)
    ip neigh
```

>> Identify the entries in your ARP table.

# Internet Control Message Protocol (ICMP)

6. Test the connectivity between your PC and your gateway and servers on the Internet. Run the following commands (ywo or three times) in a console/terminal:

```
(Windows and Linux)
    ping <ip address gateway>
    ping ua.pt
    ping ulisboa.pt
    ping abola.pt
    ping facebook.com
    ping baidu.com
```

Note: In Linux you must stop the test with CTRL+C.

>> Identify the average, minimum and maximum RTT time to the different destinations.

>> Explain the possible variation of the measured RTT values between different tests.

7. Identify the routing path from your PC to servers on the Internet. Run the following commands in a console/terminal:

```
(Windows)
    tracert ua.pt
    tracert ulisboa.pt
    tracert abola.pt
    tracert facebook.com
    tracert baidu.com
(Linux)
    traceroute -I ua.pt
    traceroute -I ulisboa.pt
    traceroute -I abola.pt
    traceroute -I facebook.com
    traceroute -I baidu.com
```

Note: In Linux you may have to install `traceroute`. For security reasons, from inside the UA network only ICMP base trace paths are allowed. From home, you may use `traceroute` without the `-I` option or use the default `tracepath` application.

>> Identify the nodes from your PC to the destination.

>> Explain the RTT values to the different nodes, and differences between nodes.

# Traffic Monitoring (with Wireshark)

Install Wireshark on your PC.

In Linux, add your user name to the wireshark group (`usermod -a -G wireshark USERNAME`) and restart. In Windows if your network adapters are nor listed, start Wireshark as administrator.

8. Start a capture on the interface that provides Internet to your PC. Open your browser and access your favorite sites. Stop the capture and analyze the packets.

>> Try to identify the packets exchanged between your PC and the servers.

>> Try to identify used protocols and how are they encapsulated at different layers.

>> Identify the IP addresses of the hosts that exchanged packets.

You may save the capture with File→Save and reopening a previous capture with File→Open.

9. Start a capture on the interface that provides Internet to your PC. Perform a set of connectivity tests. Stop the capture and analyze the packets.

>> Try to identify the packets exchanged.

>> Try to identify used protocols and how are they encapsulated at different layers.

>> Identify the IP addresses of the hosts that exchanged packets.

10. Start a capture on the interface that provides Internet to your PC. Go to YouYube and play a video. Stop the capture and analyze the packets. If possible, repeat the capture using a different browser (Firefox or Chrome/Chromium).

>> Try to identify the packets exchanged.

>> Try to identify used protocols and how are they encapsulated at different layers.

>> Identify the IP addresses of the hosts that exchanged packets.

11. Open the previous captures and apply the following visualization filters:

- IPv4 packets sent or received by your PC: `ip.addr==<ipv4_address>`
- IPv4 packets sent by your PC: `ip.src==<ipv4_address>`
- IPv4 packets received by your PC: `ip.dst==<ipv4_address>`
- Only ICMP packets: `icmp`
- ICMP **and** ARP packets: `icmp or arp`
- Only TCP packets: `tcp`
- Only HTTPS (TCP port 443): `tcp.port==443`

Note: Replace `<ipv4_address>` by your PC IPv4 address.

Note2: Wireshark has capture and visualization filters. At this stage do do not use capture filters.

12. Open one of the previous captures and explore the traffic analysis features from Wireshark. From Wireshark menu:

- Statistics→ Endpoints
- Statistics→ Conversations
- Statistics→ I/O Graphs

(optional) 13. Start a capture on the interface that provides Internet to your PC. Repeat the trace path identification tests. Stop the capture and analyze the packets.

>> Observe how trace path works (RRT values and error messages).

# Part B

## Introduction to GNS3

1. Choose your operating system (Linux/Windows), download/install GNS3 (version>2.2.0) and related software (Wireshark, VirtualBox and VPCS).

(Windows and MacOS) Download package from website https://gns3.com.

(Linux) Install from repositories; AUR for Arch/Manjaro distributions and PPA https://launchpad.net/~gns3/+archive/ubuntu/ppa for Debian/Ubuntu based distributions. Install packages gns3-server, gns3-gui, wireshark-qt, virtualbox, and VPCS. Add your user name to the wireshark group (`usermod -a -G wireshark USERNAME`) and restart.

2. At (Preferences-General), verify/setup all storing and program paths, avoiding paths with spaces and non ASCII characters.

3. At (Preferences-Server) enable **local server**, define **127.0.0.1** as host binding address.

Note: You do not need an external virtual machine (VM) to run emulation/simulation software. At (Preferences-GNS3 VM) disable the option “Enable the GNS3 VM”.

4. Download the following routers’ firmware: (i) Router 7200 Firmware 15.1(4) IOS Image, and (ii) Router 3725 Firmware 12.4(21) IOS Image.

5. At (Preferences-Dynamips-IOS Routers”) create three new router templates (“New” button on the bottom left):

- **Router 7200** - recommended IOS image: 7200 with IOS 15.1(4) and network adapters C7200-IO-2FE and PA-2FE-TX (4 FastEthernet → F0/0,F0/1+F1/0,F1/1), all other values can be the default ones;
- **Router 3725** - recommended IOS image: 3725 with IOS 12.4(21) and adaptersGT96100-FE and NM-1FE-TX (2 FastEthernet), all other values can be the default ones;
- **Switch L3** – will be a router 3725 with IOS image 12.4(21) and adapters GT96100-FE and NM-16ESW (1 FastEthernet + 16 port switch module). Choose option "This is an EtherSwitch router" when defining the device platform, all other values can be the default ones.

6. The definition of the “Idle-PC” value will allow the host machine to assign the correct amount of resources to the virtual devices. <u>You must repeat this procedures every time your PC CPU reaches values higher than 90%.</u> Check the CPU utilization with the “Task Manager” in Windows, top command in Linux and “monitor” in MacOS.

To define the “Idle-PC” value:

- Click "Idle-PC finder" during template setup, OR
- Add router to project, start it (should be the only one ON), open console (wait for prompt), left click the device and choose option "Auto Idle-PC", OR
- Add router to project, start it (should be the only one ON), open console (wait for prompt), left click the device and choose option "Idle-PC", choose one value (prefer the ones marked with *) and verify the CPU utilization. If any "dynamips" process is using more than 5%-10% CPU choose another value.

**This must be done for each router template, NOT each router! Each template will have a different “Idle-PC” value. All routers from the same template will share the same value.**

Note 1: All devices from the same template must be equal in terms of virtual hardware.

Note 2: After changing any device hardware characteristic or adding/removing network modules, the “Idle-PC” value must be changed in the template. If necessary, create a new template with different characteristics/modules.

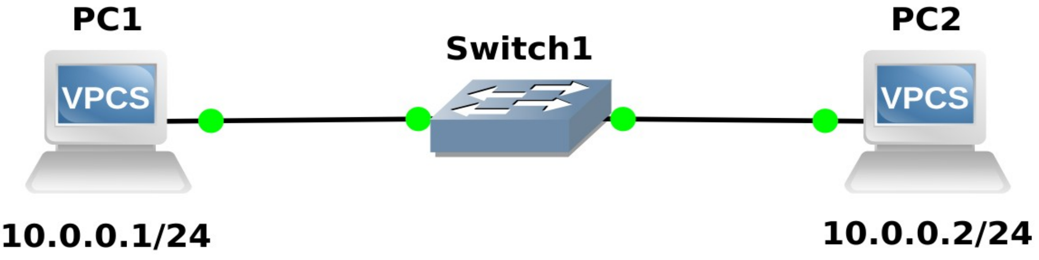
Note: At this phase your GNS3 installation should have (at least):

- 2 Routers: a Cisco c7200 and a Cisco c3725;
- An “EtherSwitch” (Layer 3 switch) based on a router c3725 with a 16 port switch module;
- An “Ethernet Switch” (does not run Spanning Tree Protocol, consumes less memory and CPU);
- Simple PC terminal with VPCS.

## Introduction to Ethernet/Switching

7. Create a new Blank Project (File menu or CTRL+N) and give it a name. Setup the following network: (i) add two PCs (VPCS) and one “Ethernet Switch”, (ii) connect each PC to the switch, you may use any port of the switch, and (iii) start the project in Control-Start/resume all nodes.

Note: The “Ethernet Switch” can be configured by right clinking it and choosing C*onfigure. By* default it has 8 access ports (numbered from 0 to 7) assigned to VLAN 1. Each port VLAN and type of port (access or trunk/802.1q/dot1q) can be changed. For now, keep all ports as access and VLAN 1.



8. Configure PC1 and PC2 IPv4 addresses and masks. Right click each PC and choose console. Add the following commands in each PC:

```
 PC1> ip 10.0.0.1/24
–
 PC2> ip 10.0.0.2/24
```

Use `show` and `show ip` commands to verify addresses and configuration. Use the `save` and `load` commands to save/load configurations. Use ? to check all available (sub-)commands.

Note: /24 defines IPv4 address mask as 255.255.255.0.

9. From PC1 console ping PC2 and vice-versa. Verify full connectivity between PC1 and PC2.

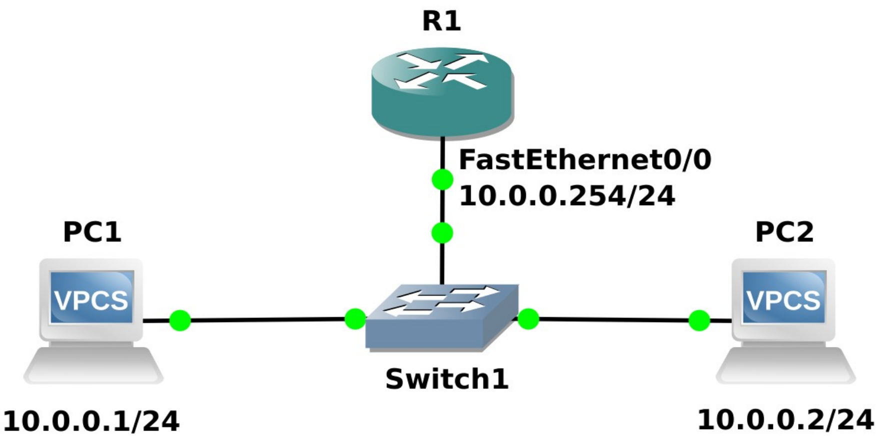
10. Start a packet capture on the link between PC1 and the Switch (right click the link and choose Start Capture). Redo pings between the PC.

>> Analyze the ARP and ICMP exchanged packets.

>> Analyze the PC’s ARP table with command `show arp`.

To save resources, stop the packet capture (right click the link and choose Stop Capture).

Note: The packet capture it is visualized with Wireshark, however is not made with Wireshark. A background process is used to store the packets in a file, Wireshark is only used to show the contents of that file. To open a background capture file right click the link and choose Stop Wireshark.



11. Add a **c7200 Router** to the project (R1) and connect port FastEthernet0/0 to any free port on Switch 1. Start the router by right clinking it and choosing Start. Wait a few seconds, open the Router's console by right clinking it and choosing Console, and press enter to obtain prompt (R1#).

12. Configure FastEthernet 0/0 interface of Router1. To enter configuration mode:

```
R1# configure terminal
R1(config)#
```

To enter the configuration of interface FastEthernet 0/0, configure IPv4 address, enable the interface and exit the configuration mode:

```
R1(config)# interface FastEthernet 0/0
R1(config-if)# ip address 10.0.0.254 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

Use the following command to verify the assign IPv4 address and interfaces status:

```
R1# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        10.0.0.254      YES manual up                    up
FastEthernet0/1        unassigned      YES unset  administratively down down
```

Note: Cisco's OS commands can be abbreviated if no conflict exists:

```
interface FastEthernet 0/0 → int f0/0
configure terminal → conf t
no shutdown → no shut
show ip interface brief → sh ip int br
```

>> Test the connectivity between both PCs and the Router.

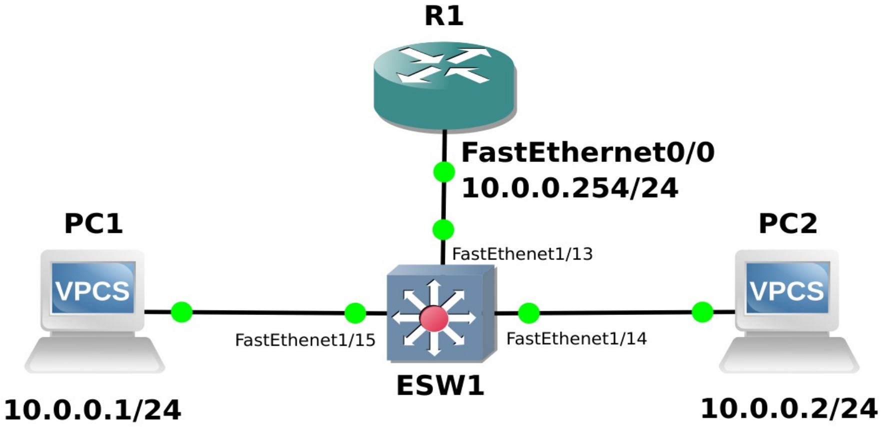
13. Start a packet capture on the link between PC1 and the Switch (right click the link and choose Start Capture). Perform ping commands from PC1 to Router using 56 bytes of data (default), 100 bytes of data, and 5 bytes of data:

```
PC1> ping 10.0.0.254
PC1> ping 10.0.0.254 -l 100
PC1> ping 10.0.0.254 -l 5
```

>> Analyze, for each packet, (i) the overall frame size, (ii) the total length of the IPv4 packet, and (iii) the ICMP data size.

>> Explain the additional information transmitted at Ethernet/Frame and Internet/IPv4 levels.

>> Analyze the ICMP response packets sent by the router (with 5 bytes of data) and explain the added zeros at the end.



14. Remove the “Ethernet Switch” and add an “EtherSwitch”. Place the cursor over the “EtherSwitch” and identify (on the pop-up) the 16 ports numbered from FastEthernet x/0 to FastEthernet x/15, those are the ones of the switch module that must be used (usually is F1/0 to F1/15). Connect PC1 to port FastEthernet x/13, PC2 to port FastEthernet x/14, and the Router to port FastEthernet x/15.

Start the equipment by right clinking it and choosing Start. Wait a few seconds, open the console of the switch by right clinking it and choosing Console, and press enter to obtain prompt (ESW1#).

>> Use the command `show ip interface brief` to verify the status of the connected ports. The status and protocol should be up and up. If not, stop and start the equipment.

15. Register PC1 and PC2 MAC addresses and Router’s F0/0 interface MAC address:

```
PC1> show
–
PC2> show
–
R1# show interfaces F0/0
```

16. From PC1 and PC2 ping the Router. Visualize the Switch’s (ESW1) forwarding table:

```
ESW1# show mac-address-table
```

>> Explain the contents of the Switch’s forwarding table.

Note: A switch forwarding table contains the learned MAC addresses and the port where the switch received a packet from that MAC address (and through which any packet to that MAC address should be sent).

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.

17. Register the Switch’s forwarding table aging time:

```
ESW1# show mac-address-table aging-time
```

Change it to 10 seconds

```
ESW1# configure terminal
ESW1(config)# mac-address-table aging-time 10
```

18. Start a capture on the link between the Switch and PC2.

From PC1 ping the Router. Check the Switch’s forwarding table

Wait 20 seconds, and perform another ping from PC1 to the router. Check again the Switch’s forwarding table

Stop the capture.

>> Explain the captured packets and the contents of the forwarding table

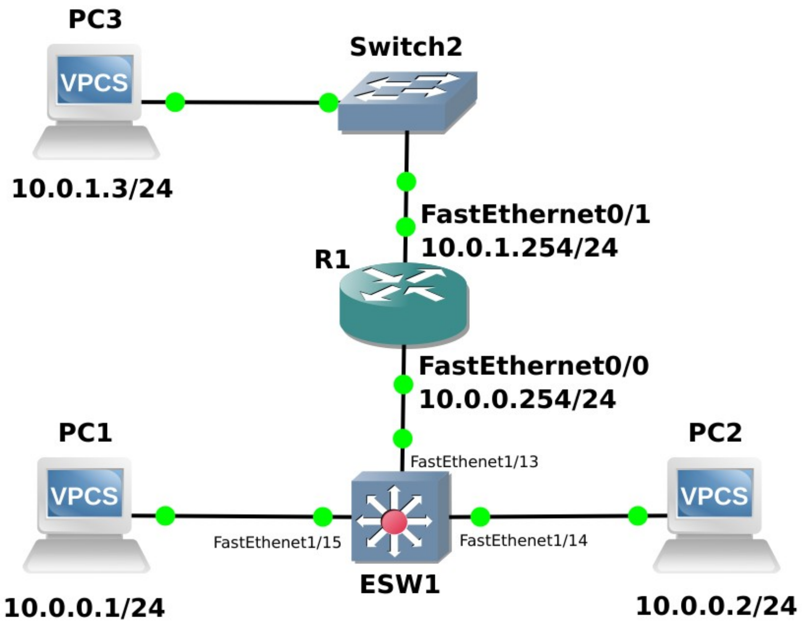
>> Explain why the Router’s interface MAC address never disappears from the forwarding table.

## Introduction to IPv4/Routing

19. Start a capture on the link between the Switch and PC1. Perform ping commands from PC1 to Router using 2000 and 3000 bytes of data:

```
PC1> ping 10.0.0.254 -l 2000
PC1> ping 10.0.0.254 -l 3000
```

>> Analyze the captured packets and explain the fragmentation process. In particular, explain: (I) why each packet is fragmented in 3 fragments, (ii) the content of the IP header fields that enable the recovery of the complete packet at the destination, and (iii) the packet size of each fragment.



20. Add another PC (PC3) and a new switch (“Ethernet Switch”). Configure the IPv4 address of PC3 and interface F0/1 of the router.

```
PC3> ip 10.0.1.3/24
–
R1# configure terminal
R1(config)# interface FastEthernet 0/1
R1(config-if)# ip address 10.0.1.254 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# end
```

Note: IPv4 networks 10.0.0.0/24 (bottom) and 10.0.1.0/24 (top) are different networks because the first 24 bytes (number defined by the mask) are different (10.0.0._ ≠ 10.0.1._).

21. Register PC3 MAC address and Router’s F0/1 interface MAC address. Start packet capture between PC1 and ESW1. Ping from PC1 to PC3.

22. Test the connectivity between all PCs and between the PCs and the Router' terminals.

>> Explain the results and propose a solution to achieve full connectivity.

23. Configure the PCs' IPv4 gateways:

```
PC1> ip 10.0.0.1/24 10.0.0.254
–
PC2> ip 10.0.0.2/24 10.0.0.254
–
PC3> ip 10.0.1.3/24 10.0.1.254
```

>> Explain the results and why now there are full connectivity.

24. Register the contents of the ARP tables of the PCs and Router. Command `show arp` in both systems.

>> Explain the contents of the ARP tables.

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 Local Area Network (LAN). Therefore, the MAC addresses of the Ethernet header are specified with the MAC addresses of the communicating hosts on each LAN.

25. Start a new capture on the link between PC1 and the ESW1. Ping from PC1 to PC3, and record the MAC and IPv4 address (source and destination).

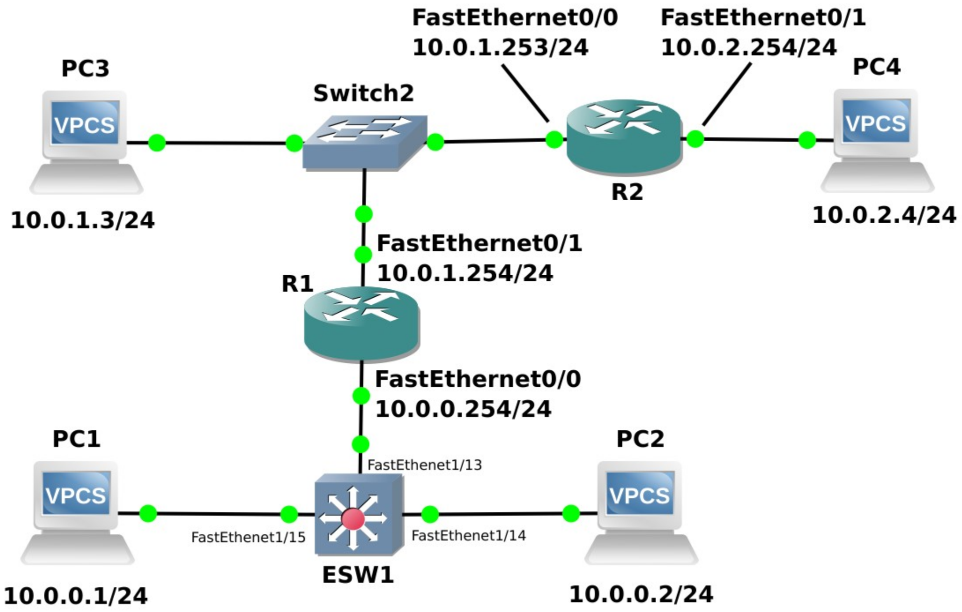
| **ICMP Echo Request** | **PC1←→ESW1** |
|---|---|
| Ethernet packet header | Source MAC Address: |
| | Destination MAC Address: |
| IPv4 packet header | Source IP Address: |
| | Destination IP Address: |
| **ICMP Echo Reply** | |
| Ethernet packet header | Source MAC Address: |
| | Destination MAC Address: |
| IPv4 packet header | Source IP Address: |
| | Destination IP Address: |

26. Without any capture, predict the content of the Ethernet and IPv4 headers on the link between ESW1 and the Router (same LAN/IPv4 network) and on the link between Switch2 and PC3 (different LAN/IPv4 network)

| **ICMP Echo Request** | **ESW1 ←→ Router** |
|---|---|
| Ethernet packet header | Source MAC Address: |
| | Destination MAC Address: |
| IPv4 packet header | Source IP Address: |
| | Destination IP Address: |
| **ICMP Echo Reply** | |
| Ethernet packet header | Source MAC Address: |
| | Destination MAC Address: |
| IPv4 packet header | Source IP Address: |
| | Destination IP Address: |

| **ICMP Echo Request** | **Switch2 ← → PC3** |
|---|---|
| Ethernet packet header | Source MAC Address: |
| | Destination MAC Address: |
| IPv4 packet header | Source IP Address: |
| | Destination IP Address: |
| **ICMP Echo Reply** | |
| Ethernet packet header | Source MAC Address: |
| | Destination MAC Address: |
| IPv4 packet header | Source IP Address: |
| | Destination IP Address: |

27. Confirm your predictions performing packet captures on the link between ESW1 and the Router and on the link between Switch2 and PC3. Repeat the ping from PC1 to PC3.



28. Add another Router (R2) and another PC (PC4). Configure the IPv4 address and gateway of PC4 and interfaces F0/0 and F0/1 of router R2.

29. Analyze the IPv4 routing table of both routers (R1 and R2):

```
R1# Show ip route
```

>> identify which IPv4 networks each router knows.

>> Predict between which PC will exist connectivity.

30. Test the connectivity between all PCs.

31. Start a new capture on the link between PC1 and the ESW1. From PC1 ping PC4 (based on the knowledge of R1, a terminal from a nonexistent IPv4 network).

>> Analyzed the captured packets and explain how PC1 knows that the destination network is nonexistent.

32. Start a new capture on the link between PC1 and the ESW1. From PC1 ping an nonexistent IPv4 address from the same network as PC3 (e.g., 10.0.1.100).

>> Analyzed the captured packets and explain how PC1 knows that the destination IPv4 is nonexistent.

33. Routers R1 and R2 must learn the existence (and path to) of the unknown IPv4 networks. One way of doing it is adding static routes (destination and IPv4 address of the next router in path):

```
R1# configure terminal
R1(config)# ip route 10.0.2.0 255.255.255.0 10.0.1.253
—
R2# configure terminal
R2(config)# ip route 10.0.0.0 255.255.255.0 10.0.1.254
```

Retest the connectivity between all PCs.

34. Re-analyze the IPv4 routing table of both routers (R1 and R2):

```
R1# Show ip route
```

>> Explain how IPv4 packets are routed between different IPv4 networks networks.

35. Start a new capture on the link between PC1 and the ESW1. From PC1 ping PC4 but with lower TTL values (default in VPCS is 64):

```
PC1> ping 10.0.2.4 -T 1
PC1> ping 10.0.2.4 -T 2
PC1> ping 10.0.2.4 -T 3
```

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

36. The *traceroute* command is a tool to discover the routers of the routing path from an origin IPv4 host to a destination IPv4 host. Start a new capture on the link between PC1 and the ESW1. At PC1, execute a trace route to PC4

Using UDP packets (default in VPCS):

```
PC1> trace 10.0.2.4
```

>> Based on the analysis of the captured packets, explain how the traceroute works. In particular: (i) identify how the PC identifies each router in the path, (ii) observe that the PC sends three packets for each growing value of TTL in order to obtain a better estimation of the round trip time, and (iii) determine how the PC stops the process.

37. Repeat the *traceroute* using now ICMP and TCP packets:

Using ICMP packets:

```
PC1> trace 10.0.2.4 -P 1
```

Using TCP packets:

```
PC1> trace 10.0.2.4 -P 6
```

38. Perform a *traceroute* to IPv4 address 10.0.2.254 and 10.0.2.4 (addresses from the same IPv4 network).

>> Explain the difference in the results