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 Sub-netting

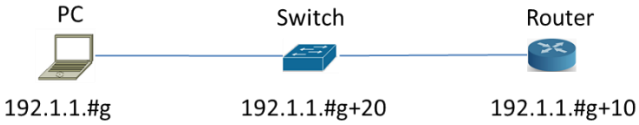
Duration

- 3 weeks

In the end of each class, please send a document to your Professor (e-mail) with the answers to the requests in bold

1.Initial Experiments

1. Build and configure the following network with the equipment 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.**

Concluo que a periodicidade do pacote ICMP é cerca de 1 segundo.
Só é possível pois é associado um Sequence Number à reply de cada request, tornando a round-trip-time curta.

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:** 7c:c2:c6:18:13:03
- **Router Ethernet address:**00:21:d8:65:0e:70
- **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:** 0x0000
- **Hexadecimal code (*Type* field of ICMP header) that identifies the two ICMP packet types (*Echo Request* and *Echo Reply*):** 0x08 (Request) e 0 (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: Source: TPLink_18:13:03

Destination Address: Destination: Broadcast

ARP packet

Origin MAC address: 7c:c2:c6:18:13:03

Origin IP address: 192.1.1.7

Destination MAC address: 00:00:00:00:00:00

Destination IP address: 192.1.1.17

ARP Response **Ethernet header**

Origin address: Source: Cisco_e5:12:20

Destination Address: Destination: TPLink_18:13:03

ARP packet

Origin MAC address: 00:1d:70:e5:12:20

Origin IP address: 192.1.1.17

Destination MAC address: 7c:c2:c6:18:13:03

Destination IP address: 192.1.1.7

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 and explain how it works.**

When the ping command is executed, the ARP resquest only sends 42 bytes, but replys with 60 bytes, which 18 bytes is padding. The ICMP request sends 14 bytes from Ethernet Header, 20 bytes from the IP header, 8 bytes of ICMP Header and 5 bytes of data (47 Bytes total).

At the reply we get 60 bytes, 47 bytes + 13 of padding.

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.

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:
Windows:
ping Router -l 2000
ping Router -l 3100
Linux:
ping Router -s 2000
ping Router -s 3100
Repeat the ping commands from the Router do the PC using 2000 and 3100 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 either 2 or 3 fragments;** 1*
- **the content of the IP header fields that enable the recovery of the complete packet at the destination;** 2*
- **the packet size of each fragment.** 3*

1* Depending on how many bytes are sent by the ping, since the Ethernet MTU is 1500 bytes, the packet is divided by 1500. Accounting for the IP header (20 bytes) and ICMP header (8 bytes) in the first fragment, the data size is 1472 bytes. This division can lead to a scenario with either 2 or 3 fragments, depending on the total packet size.

2* The contents in the IP header that enable recovery at the destination are:

ID: Identifies fragments of the same packet.
Offset: Specifies the position of each fragment in the original packet.
Flag: Indicates if more fragments are coming.

At the destination, fragments with the same Identification are reassembled in order according to the Offset. The process completes when the fragment with flag 0x00 is received.

3* Considerando um pacote ICMP Echo Request com 3100 bytes de dados a ser enviado pela rede:

Dados Totais: 3100 bytes
Cabeçalho ICMP: 8 bytes
Cabeçalho IP: 20 bytes
Tamanho total do pacote: 3128 bytes
Cálculo dos Fragmentos:

Dados por Fragmento:

1º fragmento: 1500 bytes (MTU) – 20 bytes (IP header) – 8 bytes (ICMP header) = 1472 bytes
2º fragmento: 1500 bytes (MTU)–20 bytes (IP header)=1480 bytes

Número de Fragmentos:

Primeiro Fragmento: 1472 bytes de dados + 20 bytes de cabeçalho IP + 8 bytes (ICMP header) = 1500 bytes
Segundo Fragmento: 1480 bytes de dados + 20 bytes de cabeçalho IP = 1500 bytes
Terceiro Fragmento: 148 bytes de dados + 20 bytes de cabeçalho IP = 168 bytes
Total: 3 fragmentos (1500 + 1500 + 168 = 3148 bytes)

Para Ping de 2000 bytes:

Dados a enviar: 2000 bytes
Cabeçalho ICMP: 8 bytes
Cabeçalho IP: 20 bytes
Tamanho total do pacote: 2028 bytes
Cálculo dos Fragmentos:

Dados por Fragmento:

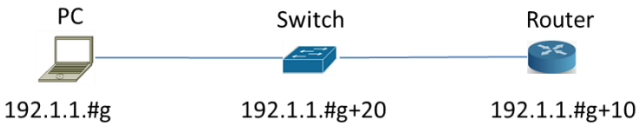
1º fragmento: 1500 bytes (MTU)–20 bytes (IP header)–8 bytes (ICMP header)=1472 bytes
2º fragmento: 1500 bytes (MTU)–20 bytes (IP header)=1480 bytes

Número de Fragmentos:

Primeiro Fragmento: 1472 bytes de dados + 20 bytes de cabeçalho IP + 8 bytes (ICMP header) = 1500 bytes
Segundo Fragmento: 528 bytes de dados + 20 bytes de cabeçalho IP = 548 bytes
Total: 2 fragmentos (1500 + 548 = 2048 bytes)

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 (ipconfig, ifconfig) and on the Router (show interfaces) 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).**

No experimento 4, a MAC Address do PC não desaparece, pois enquanto o Management Console do Switch estiver aberto o PC e o Switch estão sempre a comunicar, isso implica que o contador do Aging Time seja resetado sempre que recebe um pacote do PC.

Já no experimento 5, ao fecharmos o Management Web do Switch, o Switch e o PC param de comunicar, após alguns segundos, quando passar o Aging Time, o Switch vai apagar o MAC Address do PC da sua tabela de encaminhamento, pois os dois já não estão a comunicar.

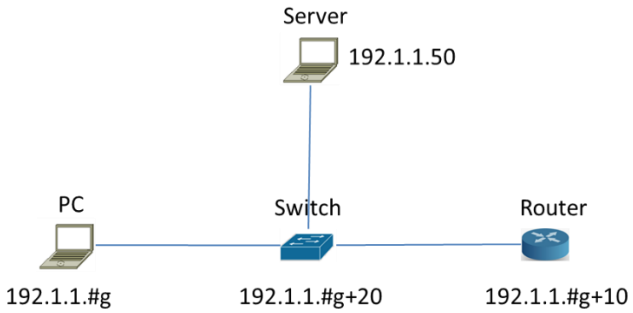
6. Remember from the theoretical classes that, when a Switch receives a packet on an incoming port, it searches for an entry with the packet destination MAC address on its *MAC Address Table*. Then, the behaviour of the Switch is one of two possibilities:

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.**

É enviado um ping do Router para o Server, este ping chega ao Switch, se a tabela de encaminhamento estiver vazia, o switch não vai saber por qual interface tem que chegar ao Server.

Para isso, o Switch faz um flooding, ou seja, envia por todas as interfaces um request, no qual haverá um reply a indicar qual é o Server.

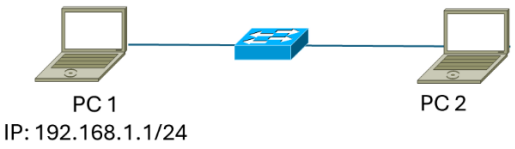
A única informação que o PC (e outros dispositivos ligados) recebe(m) é o request do flooding.

O Switch após receber o reply, o mesmo regista na tabela a interface e o MAC Address de onde veio a resposta (MAC Address do Server), bem como a interface e MAC Address do Router.

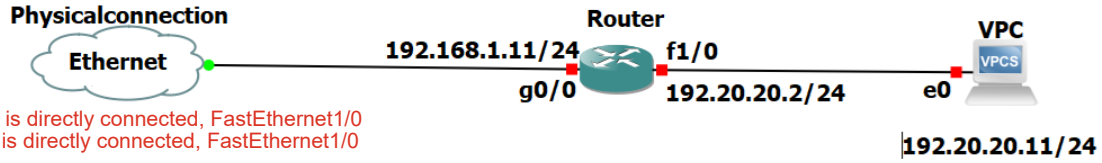
Durante o Aging Time, o Switch saberá o Endereço MAC do Server (e do Router) e todos os pacotes enviados do Router para o Server serão redirecionados (fowarding) quando passam pelo Switch e vice-versa para os replys do Server para o Router.

3. Experiments with routers

For the following exercise, you need to build the network shown below, using one switch from the classroom and two PCs.



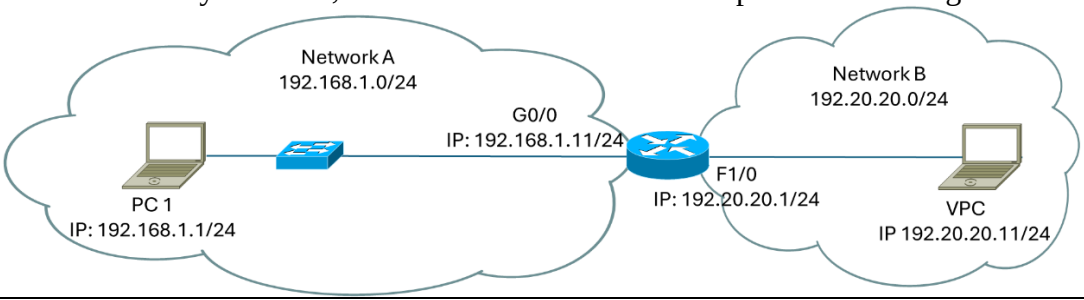
"PC 1" (Windows, Linux or MAC) needs to have at least Wireshark installed and should have firewall disabled during the exercise. The network interface connected to the switch must be configured as shown.
"PC 2" must have GNS3 installed and, using GNS3, the following network must be created (use the "cloud" from GNS to allow GNS to use the Ethernet adapter of your PC 2 to connect to the external switch):



C 192.20.20.0/24 is directly connected, FastEthernet1/0
L 192.20.20.7/32 is directly connected, FastEthernet1/0

C 192.168.1.0/24 is directly connected, GigabitEthernet0/0
L 192.168.1.17/32 is directly connected, GigabitEthernet0/0

Once everything is configured and connected, your network should be two IP networks, interconnected by a Router, with the class C IP addresses specified in the figure below.



1. Without configuring *Default Gateway on the PC and VPC,*
Register and justify the routing table of the Router.

2. Start a capture with *wireshark*. Execute the *ping* command from the PC to the VPC. Repeat the experiment but now executing the *ping* command from the VPC (through its console in GNS3) to the PC.
Register and justify both the *ping* command results and the captured packets.

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

4. 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 VPC.
Register and justify the *ping* command result. Register also the following

2. Without a default gateway configured on both the PC and VPC, the ping commands will fail. This is because neither device knows how to route packets to destinations outside their local subnet

3.
192.168.1.1 icmp_seq=1 timeout
192.168.1.1 icmp_seq=2 timeout
192.168.1.1 icmp_seq=3 timeout
192.168.1.1 icmp_seq=4 timeout
192.168.1.1 icmp_seq=5 timeout

The ping from the VPC to the PC will fail. This is because while the VPC can send ICMP Echo Requests to the PC through the router (via the configured default gateway), the PC will not know how to route the ICMP Echo Replies back to the VPC, as it lacks a route for destinations outside its subnet

The result is that ICMP Echo Request packets are sent, but ICMP Echo Reply packets do not return to the VPC. The PC only knows its own subnet, and without a default route or a specific route to the 192.20.20.0/24 subnet, it discarded responses, resulting in "timeouts" on ping attempts.

4.
VPC: 192.20.20.11/24
PC: 192.168.1.1/24

PC> ping 192.20.20.11
84 bytes from 192.20.20.11 icmp_seq=1 ttl=63 time=30.687 ms
84 bytes from 192.20.20.11 icmp_seq=2 ttl=63 time=30.528 ms
84 bytes from 192.20.20.11 icmp_seq=3 ttl=63 time=30.243 ms
84 bytes from 192.20.20.11 icmp_seq=4 ttl=63 time=31.154 ms
84 bytes from 192.20.20.11 icmp_seq=5 ttl=63 time=30.233 ms

Now that both PCs have a Default Gateway properly configured, they can now communicate with each other.

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:** 00:50:79:66:68:01 **Destination MAC Address:** ca:01:2a:00:00:08
IP packet header	**Source IP Address:** 192.168.1.1 **Destination IP Address:** 192.20.20.11

ICMP Echo Reply	
Ethernet packet header	**Source MAC Address:** ca:01:2a:00:00:08 **Destination MAC Address:** 00:50:79:66:68:01
IP packet header	**Source IP Address:** 192.20.20.11 **Destination IP Address:** 192.168.1.1

5. 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 VPC on the previous experiment (if needed, check the addresses on the equipment):**

ICMP Echo Request	
Ethernet packet header	**Source MAC Address:** ca:01:2a:00:00:1c **Destination MAC Address:** 00:50:79:66:68:00
IP packet header	**Source IP Address:** 192.20.20.1 **Destination IP Address:** 192.20.20.11

ICMP Echo Reply	
Ethernet packet header	**Source MAC Address:** 00:50:79:66:68:00 **Destination MAC Address:** ca:01:2a:00:00:1c
IP packet header	**Source IP Address:** 192.20.20.11 **Destination IP Address:** 192.20.20.1

6. **Register and justify the ARP tables of the Router.**

7. Start a new capture with a display filter for ICMP and ARP packets and execute the *ping* command from the *VPC* to the IP address 192.1.1.10 (an inexistent IP address of your network). **Register the captured packets and explain the obtained results.**

8. Start a capture with *wireshark*. Execute the *ping* command from the VPC to the IP address 194.100.1.1. Register the captured packets. Justify the observed packets taking

6. R1#show ip arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	192.20.20.1	-	ca01.2a00.001c	ARPA	FastEthernet1/0
Internet	192.20.20.11	5	0050.7966.6800	ARPA	FastEthernet1/0
Internet	192.168.1.1	21	0050.7966.6801	ARPA	GigabitEthernet0/0
Internet	192.168.1.11	-	ca01.2a00.0008	ARPA	GigabitEthernet0/0

The IPs (192.20.20.1 & 192.168.1.11) without age are the interfaces of the router itself. Where you can see the IP and MAC Address associated with each interface.

192.20.20.11 is the IP of PC1, the age means that the router performed an ARP request 5 minutes ago for this IP address, or it was recently accessed on the network. We can also see your MAC address and which interface you are connected to.

192.168.1.1 is the IP of PC1, the age means that the router performed an ARP request 21 minutes ago for this IP address, or it was recently accessed on the network. We can also see your MAC address and which interface you are connected to.

7.

4	18.807293	Private_66:68:00	Broadcast	ARP	64 Who has 192.20.20.1? Tell 192.20.20.11
5	18.822404	ca:01:2a:00:00:1c	Private_66:68:00	ARP	60 192.20.20.1 is at ca:01:2a:00:00:1c
6	18.837324	192.20.20.11	192.1.1.10	ICMP	98 Echo (ping) request id=0xbbd3, seq=1/256, ttl=64 (no response found!)
7	18.852483	192.20.20.1	192.20.20.11	ICMP	70 Destination unreachable (Host unreachable)
8	19.882309	192.20.20.11	192.1.1.10	ICMP	98 Echo (ping) request id=0xbcd3, seq=2/512, ttl=64 (no response found!)
9	19.897511	192.20.20.1	192.20.20.11	ICMP	70 Destination unreachable (Host unreachable)
11	20.913669	192.20.20.11	192.1.1.10	ICMP	98 Echo (ping) request id=0xbdd3, seq=3/768, ttl=64 (no response found!)
12	20.928627	192.20.20.1	192.20.20.11	ICMP	70 Destination unreachable (Host unreachable)
13	21.958893	192.20.20.11	192.1.1.10	ICMP	98 Echo (ping) request id=0xbed3, seq=4/1024, ttl=64 (no response found!)
14	21.974448	192.20.20.1	192.20.20.11	ICMP	70 Destination unreachable (Host unreachable)
15	22.989872	192.20.20.11	192.1.1.10	ICMP	98 Echo (ping) request id=0xbfd3, seq=5/1280, ttl=64 (no response found!)
16	23.005417	192.20.20.1	192.20.20.11	ICMP	70 Destination unreachable (Host unreachable)

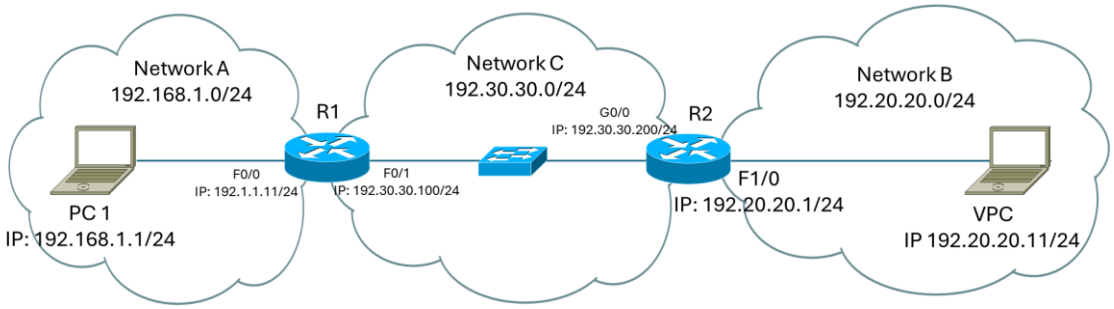
This ARP exchange allows the PC to learn the MAC address of the router so that it can send packets to it for forwarding. Since the PC wants to ping an IP outside its local network (192.1.1.10), it first needs the router's MAC address to act as the next hop.
The router, acting as the gateway, forwards the PC's ICMP Echo Requests to 192.1.1.10, but since this IP address does not exist in the network, it sends back an ICMP Destination Unreachable message to the PC, informing it that the destination is unreachable.

---

8. Switches handle unknown destinations through flooding, which is a broadcast approach within the local network, allowing t hem to learn MAC addresses dynamically.
Routers handle unknown destinations by checking their routing tables and, if no route is found, return an ICMP error message (no broadcasting occurs)

in mind that the Router has no entry for this IP address. **What do you conclude about the difference between the Switch forwarding process (previous switching experiment) and the Router forwarding processes when the destination address is not known?**

9. Connect a real router to the first one, as shown in the next figure and reconfigure the interface of the router in GNS3, according to the figure.



10. **Register the routing table of R1 and compare it with the one of the experiment with only 1 router.** 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.

11. Start a capture with *wireshark*. Execute the *ping* command from the PC to the IP address 192.30.30.150 (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 R1 (in the network 192.30.30.0) taking into consideration the results of experiment 8.**

12. Start a capture with a display filter for ICMP packets. Execute the *ping* command from the PC to the IP address 192.20.20.1 (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 192.30.30.0) taking into consideration the results of experiment 11.**

13. Configure a static route in R1 and another in R2 to allow full connectivity on the entire network. **Register the routing table of the routers. Observe that, now, the routing protocol enabled the routers to add information on their routing table concerning the new network.** Then, execute the *ping* command from the PC to the IP address 192.20.20.11 to verify the connectivity between your PC and the new network.

☞ 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):

```
R1 (Define the path to network 192.20.20.0 through R2)
```

```
Router#configure terminal
Router(config)#ip route 192.20.20.0 255.255.255.0 192.30.30.200
```

R2 (**Define the path to network 192.1.1.0 through R1**)

```
Router#configure terminal
Router(config)#ip route 192.1.1.0 255.255.255.0 192.30.30.100
```

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

```
ping –i 1 192.20.20.11
ping –i 2 192.20.20.11
ping –i 3 192.20.20.11

Note: If you PC1 was also inside GNS3, the command would be:
ping 192.20.20.11 -T 1
ping 192. 20.20.11 -T 2
ping 192. 20.20.11 -T 3
```

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.**

15. At your PC, start a capture with a display filter for ICMP packets and execute the command *tracert –d 192.20.20.11 (GNS-3:* trace 192.20.20.11). 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.

16. **Verify and justify the differences obtained when executing in your PC the command *tracert –d* for the IP addresses *192.20.20.11* and *192.20.20.1*.**

10.
R1#show ip route

Gateway of last resort is not set

C 192.30.30.0/24 is directly connected, FastEthernet0/1
C 192.168.1.0/24 is directly connected, FastEthernet0/0

11.

27	180.147050	Private_66:68:00	Broadcast	ARP	64 Who has 192.168.1.11? Tell 192.168.1.1
28	180.162186	c2:01:33:34:00:00	Private_66:68:00	ARP	60 192.168.1.11 is at c2:01:33:34:00:00
29	180.177213	192.168.1.1	192.30.30.150	ICMP	98 Echo (ping) request id=0x25e6, seq=1/256, ttl=64 (no response found!)
31	182.194130	192.168.1.1	192.30.30.150	ICMP	98 Echo (ping) request id=0x27e6, seq=2/512, ttl=64 (no response found!)
32	184.199295	192.168.1.1	192.30.30.150	ICMP	98 Echo (ping) request id=0x29e6, seq=3/768, ttl=64 (no response found!)
33	186.225440	192.168.1.1	192.30.30.150	ICMP	98 Echo (ping) request id=0x2be6, seq=4/1024, ttl=64 (no response found!)
34	188.246052	192.168.1.1	192.30.30.150	ICMP	98 Echo (ping) request id=0x2de6, seq=5/1280, ttl=64 (no response found!)

In this capture, the ARP exchange allows the device to learn the MAC address of the router (192.168.1.1). The device then sends ICMP Echo Requests to 192.30.30.150, which is outside its local network. Since there are no responses to the ICMP Echo Requests, the destination IP 192.30.30.150 does not exist or is unreachable in the network 192.30.30.0.

R1 receives these requests and forwards them into the 192.30.30.0 network, looking for the nonexistent 192.30.30.150 (flooding).

12.

No.	Time	Source	Destination	Protocol	Length	Info
6	14.625224	192.168.1.1	192.20.20.1	ICMP	98	Echo (ping) request id=0xa0ec, seq=1/256, ttl=64 (no response found!)
7	14.640640	192.168.1.11	192.168.1.1	ICMP	70	Destination unreachable (Host unreachable)
8	15.656382	192.168.1.1	192.20.20.1	ICMP	98	Echo (ping) request id=0xa1ec, seq=2/512, ttl=64 (no response found!)
9	15.671408	192.168.1.11	192.168.1.1	ICMP	70	Destination unreachable (Host unreachable)
10	16.693010	192.168.1.1	192.20.20.1	ICMP	98	Echo (ping) request id=0xa2ec, seq=3/768, ttl=64 (no response found!)
11	16.708561	192.168.1.11	192.168.1.1	ICMP	70	Destination unreachable (Host unreachable)
12	17.732733	192.168.1.1	192.20.20.1	ICMP	98	Echo (ping) request id=0xa3ec, seq=4/1024, ttl=64 (no response found!)
13	17.747743	192.168.1.11	192.168.1.1	ICMP	70	Destination unreachable (Host unreachable)
14	18.766398	192.168.1.1	192.20.20.1	ICMP	98	Echo (ping) request id=0xa4ec, seq=5/1280, ttl=64 (no response found!)
15	18.781418	192.168.1.11	192.168.1.1	ICMP	70	Destination unreachable (Host unreachable)

The PC sends an ICMP Echo Request to 192.20.20.1, attempting to reach a host in a network unknown to the router.
These packets are forwarded to the router (R1), which then tries to find the path to 192.20.20.1.
R1 doesn't have a route to 192.20.20.1 and hence sends back ICMP Destination Unreachable messages to 192.168.1.1.
Nothing should appears on 192.30.30.0 because the packets don't leave the router, because there's no route and sends back an error message

---

13.
R1#show ip route
Gateway of last resort is not set

C 192.30.30.0/24 is directly connected, FastEthernet0/1
S 192.20.20.0/24 [1/0] via 192.30.30.200
C 192.168.1.0/24 is directly connected, FastEthernet0/0

R2#show ip route
Gateway of last resort is not set

192.20.20.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.20.20.0/24 is directly connected, FastEthernet1/0
L 192.20.20.1/32 is directly connected, FastEthernet1/0
192.30.30.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.30.30.0/24 is directly connected, GigabitEthernet0/0
L 192.30.30.200/32 is directly connected, GigabitEthernet0/0
S 192.168.1.0/24 [1/0] via 192.30.30.100

These static routes were added so that each router is aware of networks that are not directly connected to it:

On R1: Static route S 192.20.20.0/24 [1/0] via 192.30.30.200 allows R1 to route packets destined for network 192.20.20.0/24 (where the VPC is located) through R2, using next hop 192.30.30.200.

On R2: Static route S 192.168.1.0/24 [1/0] via 192.30.30.100 allows R2 to route packets destined for network 192.168.1.0/24 (where the PC is located) through R1, using next hop 192.30.30.100.

These routes allowed the networks 192.168.1.0/24 and 1 92.20.20.0/24 to communicate via the intermediate link 192.30.30.0/24, enabling ping between the PC and the VPC.

14.
Case 1: Ping with TTL=1
The ICMP packets start with a TTL of 1. When these packets reach the first router (R1), the TTL is decremented by 1, causing it to reach 0. The router then discards the packet and sends back an ICMP Time Exceeded message to the source (PC1).

Case 2: Ping with TTL=2
The ICMP packets start with a TTL of 2. They reach R1, which decrements the TTL to 1 and forwards the packets to R2. Upon reaching R2, the TTL is decremented again to 0, causing R2 to discard the packets and send back ICMP Time Exceeded messages.

Case 3: Ping with TTL=3
The ICMP packets start with a TTL of 3 to reach the destination (192.20.20.11) without being discarded.
Each hop decrements the TTL by 1, but the packets reach the destination before TTL expires.

15.
tracert is a network diagnostic tool used to track the path that packets take from your computer to a destination IP. It helps identify each hop along the route and measures the round trip time (RTT) to each hop.

i) The PC sends ICMP Echo Requests and receives ICMP Time Exceeded messages from each router along the path. For example, a TTL=1 packet receive a response from 192.168.1.11, indicating the first router in the path.

ii) The PC sends three ICMP Echo Request packets for each TTL value to get a better RTT estimate:
1 192.168.1.11 15.169 ms 15.403 ms 15.737 ms
2 192.30.30.200 45.006 ms 46.021 ms 45.053 ms
3 * * *
4 192.20.20.11 44.809 ms (ICMP type:3, code:3, Destination port unreachable)
The response show the RTT for each hop

iii) The process stops when the destination (192.20.20.11) responds with an ICMP Echo Reply or an ICMP Destination Unreachable message:

4 192.20.20.11 44.809 ms (ICMP type:3, code:3, Destination port unreachable)

Hops and Response Times:
trace 192.20.20.11, there are responses from the first hop (192.168.1.11) and the second hop (192.30.30.200) and last message indicating "Destination port unreachable."

trace 192.20.20.1, responses are received from the first hop (192.168.1.11), and then an "Destination port unreachable" message from the second hop (192.30.30.200).

Difference in Hops:
192.20.20.11 trace reaches further, showing the attempt to reach the destination address through multiple hops but eventually hitting a "Destination port unreachable" response at hop 4.

192.20.20.1 trace gets an immediate "Destination port unreachable" message at the second hop, indicating that the router directly connected to 192.20.20.1 knows it cannot route to the address.

4. IP Sub-netting

1. What is the broadcast address of the networks:
   - 200.3.27.128/25 Endereço de broadcast: 200.3.27.255
   - 200.3.27.0 and 200.3.27.128 mask 255.255.248.0? Endereço de broadcast para ambas as redes: 200.3.31.255.
2. What is the first terminal of the networks that contain the terminals with the address:
   - 175.0.92.191/23 Primeiro terminal: 175.0.92.1
   - 175.0.92.190/26 Primeiro terminal: 175.0.92.129
   - 175.0.92.18/28? Primeiro terminal: 175.0.92.17
3. What is the last terminal in the networks:
   - 175.0.32.0 mask 255.255.248.0 Último terminal: 175.0.39.254
   - 175.0.0.0 mask 255.255.224.0 Último terminal: 175.0.31.254
   - 175.0.16.0 mask 255.255.248.0 Último terminal: 175.0.23.254
4. What are the networks of the terminal:
   - 175.0.22.79/25 Network: 175.0.22.0/25
   - 175.0.117.215/23 Network: 175.0.116.0/23
   - 175.0.117.215/27? Network: 175.0.117.192/27
5. How many networks and with how many terminals can be obtained with the networks divided as in the following:
   - 175.0.4.0 mask 255.255.255.252 1 Network and 2 terminals per network.
   - 175.0.114.0 255.255.255.240? 1 Network and 14 terminals per network.
6. Consider that you have the following IPv4 class C addresses 200.123.189.0/24, and you have to use it through different sub-networks:
   - 55 PCs at the Networks1 Lab
   - 48 PCs at the Networks2 Lab
   - 45 servers at the Internal Datacenter
   - 5 PCs in the Professors Lab
   - 9 PCs in the Administration room.

   Define an addressing scheme for the different sub-networks using the overall available class C address.

Required subnets

Network1 Lab (55 PCs) - We need 64 addresses (/26).
Network2 Lab (48 PCs) - We need 64 addresses (/26).
Internal Datacenter (45 servers) - We need 64 addresses (/26).
Professors Lab (5 PCs) - We need 8 addresses (/29).
Administration Room (9 PCs) - We need 16 addresses (/28).

Addressing scheme

200.123.189.0/26: Network1 Lab (64 addresses, from 200.123.189.0 to 200.123.189.63)
200.123.189.64/26: Network2 Lab (64 addresses, from 200.123.189.64 to 200.123.189.127)
200.123.189.128/26: Internal Datacenter (64 addresses, from 200.123.189.128 to 200.123.189.191)
200.123.189.192/29: Professors Lab (8 addresses, from 200.123.189.192 to 200.123.189.199)
200.123.189.200/28: Administration Room (16 addresses, from 200.123.189.200 to 200.123.189.215)

1)

a) 200.3.27.128/25

Mask /25 means 255.255.255.128
The range is 128 addresses (since 2^7 = 128).
The network starts at 200.3.27.128 and goes to 200.3.27.255.

Broadcast address: 200.3.27.255.

b) 200.3.27.0 and 200.3.27.128 with mask 255.255.248.0
The mask 255.255.248.0 corresponds to /21, which covers 2048 addresses.
The 200.3.24.0/21 network covers from 200.3.24.0 to 200.3.31.255.
Broadcast address for both networks: 200.3.31.255.

2)
a) 175.0.92.191/23
Mask /23 means 255.255.254.0
This network goes from 175.0.92.0 to 175.0.93.255.
First terminal: 175.0.92.1.

b) 175.0.92.190/26
Mask /26 means 255.255.255.192
Each subnet has 64 addresses, so this network covers from 175.0.92.128 to 175.0.92.191.
First terminal: 175.0.92.129.

c) 175.0.92.18/28
Mask /28 means 255.255.255.240
Each subnet has 16 addresses, so this network covers from 175.0.92.16 to 175.0.92.31.
First terminal: 175.0.92.17.

3)
a) 175.0.32.0 with mask 255.255.248.0
Mask 255.255.248.0 is equivalent to /21.
This network covers from 175.0.32.0 to 175.0.39.255.
Last terminal: 175.0.39.254.

b) 175.0.0.0 with mask 255.255.224.0
Mask 255.255.224.0 is equivalent to /19.
This network covers from 175.0.0.0 to 175.0.31.255.
Last terminal: 175.0.31.254.

c) 175.0.16.0 with mask 255.255.248.0
Mask 255.255.248.0 is equivalent to /21.
This network covers from 175.0.16.0 to 175.0.23.255.
Last terminal: 175.0.23.254.

4)
a) 175.0.22.79/25
Mask /25 means 255,255,255,128.
This network goes from 175.0.22.0 to 175.0.22.127.
Network: 175.0.22.0/25.

b) 175.0.117.215/23
Mask /23 means 255.255.254.0.
This network goes from 175.0.116.0 to 175.0.117.255.
Network: 175.0.116.0/23.

c) 175.0.117.215/27
Mask /27 means 255,255,255,224.
This network goes from 175.0.117.192 to 175.0.117.223.
Network: 175.0.117.192/27.

5)
a) 175.0.4.0 with mask 255.255.255.252
Mask 255.255.255.252 is equivalent to /30, which has 4 addresses (2 for hosts).
Number of possible subnets: N/A (it is a single network with mask /30).
Number of terminals per network: 2 (addresses for hosts).

b) 175.0.114.0 with mask 255.255.255.240
Mask 255.255.255.240 is equivalent to /28, which has 16 addresses (14 for hosts).
Number of possible subnets: N/A (it is a single network with mask /28).
Number of terminals per network: 14 (addresses for hosts).

Annex A

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:

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

The interface name may not be eth0 (remember Guide1). Check on your environment the correct interface name, running `ifconfig`

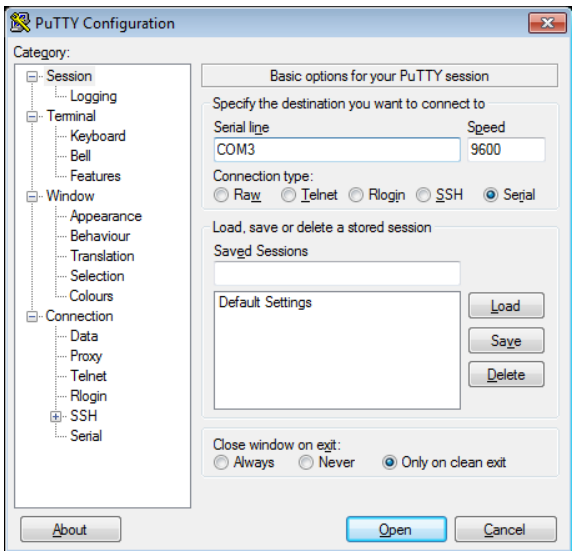
To configure the Gateway of the PC, execute the following command:

```
sudo 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 (Windows). 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 and check which COM is active. In the Putty configuration below, COM3 is used.



For Linux, use **picocom**:

```
sudo apt install picocom
sudo picocom -b 9600 /dev/ttyUSB0
```

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

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
```

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 `FastEthernet 0/0`), execute the following commands (the following example refers to Group 1):

```
router>enable
router#show run
```

observe the router configuration and the name of the interfaces

```
router#configure terminal
router(config)#
router(config)#interface FastEthernet 0/0
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 FastEthernet 0/0`? 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
```

Annex B

6. 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", consumes less memory and CPU, but does not have an IP address;
- Simple PC terminal with VPCS.

Configuration of The PCs:

```
PC1> ip 10.0.0.1/24
```

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.

Configuration of the VPCs:

```
PC1> ip 10.0.0.1/24
```

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.

Configuration of the default route in the VPCs:

```
PC1> ip 10.0.0.1/24 10.0.0.254
```

Run ping in the VPCs with TTL:

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

Run traceroute in the VPCs:

```
PC1> trace 10.0.2.4
```