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

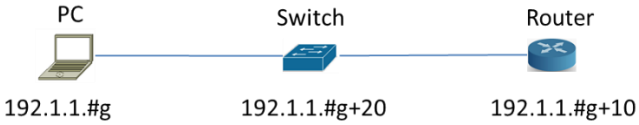
Duration

- 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.**

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: 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: ff:ff:ff:ff:ff:ff

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 and 13 bytes from ICMP Header (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 much bytes are sent by the ping, since the Ethernet MTU is 1500 bytes, the packet is divided by 1500. Accounting for the IP header and ICMP header (only for the first fragments) the data is 1480 bytes, and this leads to the 2 or 3 fragment scenario, depending on the total packet size

2* The content 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 recovered in order using the Offset order, and the process completes when the fragment with flag = 0x01 is received.

3* The packet size of each fragment is determined by the MTU and the size of the original packet.

Ethernet MTU: 1500 bytes
IP header: 20 bytes
Data per fragment: 1480 bytes (1500 bytes - 20 bytes for the IP header).

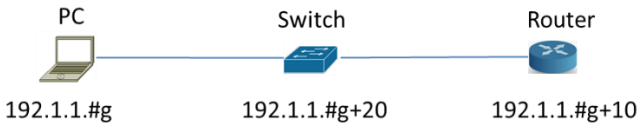
Example Fragment Sizes:
For ping Router -l 2000

2000 bytes ICMP payload
8 bytes ICMP header
2008 bytes total

First fragment: 1500 bytes (1480 bytes data + 20 bytes IP header).
Second fragment: 548 bytes (528 bytes data + 20 bytes IP header).

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).**

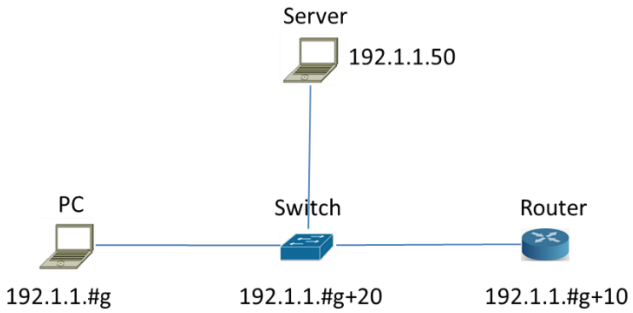
6. Remember from the theoretical classes that, when a Switch receives a packet on an incoming port, it searches for an entry with the packet destination MAC address on its *MAC Address Table*. Then, the behaviour of the Switch is one of two possibilities:

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.

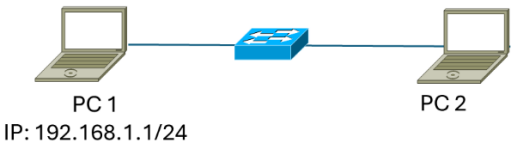
7. Add to your network a connection to a Server (your PC in the lab or of your colleague) connected to the switch. Test the connectivity by executing a *ping* command from the Router to the Server.



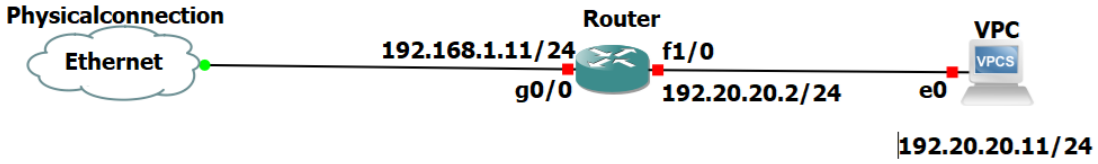
8. With *WireShark*, start a capture with a display filter for ICMP packets. Execute once again the *ping* command from the *router* to the Server. **Register the captured packets.** Note that the *ping* command has generated the exchange of 5 ICMP *Echo Request* and 5 ICMP *Echo Reply* packets between 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

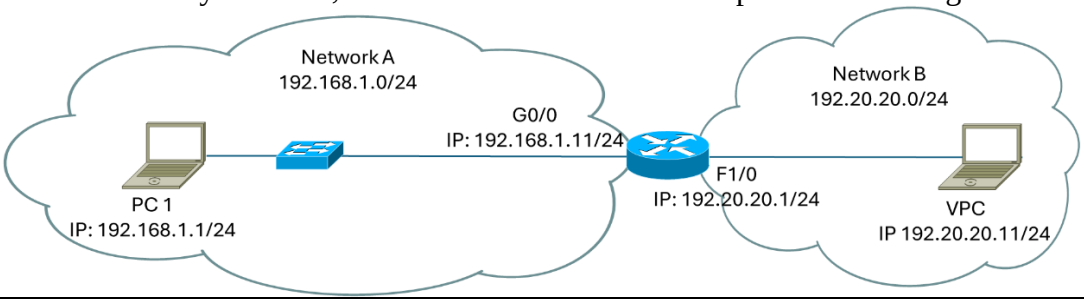
For the following exercise, you need to build the network shown below, using one switch from the classroom and two PCs.



"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):



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

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:** **Destination MAC Address:**
IP packet header	**Source IP Address:** **Destination IP Address:**

ICMP Echo Reply	
Ethernet packet header	**Source MAC Address:** **Destination MAC Address:**
IP packet header	**Source IP Address:** **Destination IP Address:**

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:** **Destination MAC Address:**
IP packet header	**Source IP Address:** **Destination IP Address:**

ICMP Echo Reply	
Ethernet packet header	**Source MAC Address:** **Destination MAC Address:**
IP packet header	**Source IP Address:** **Destination IP Address:**

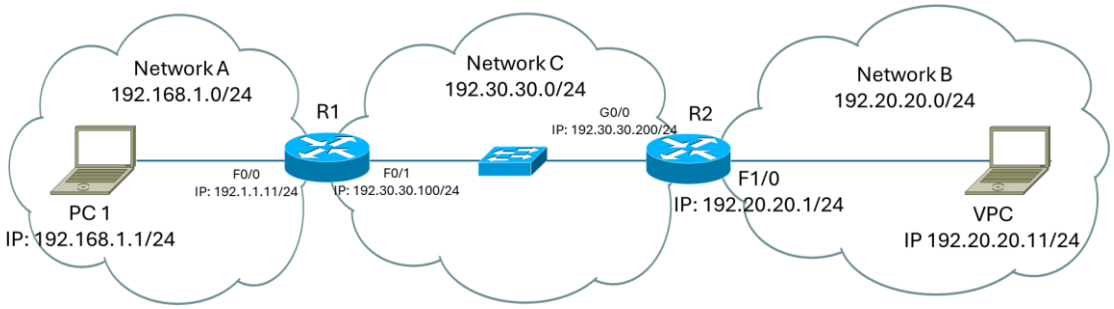
6. **Register and justify the ARP tables of the Router.**

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

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*.**

4. IP Sub-netting

1. What is the broadcast address of the networks:
   - 200.3.27.128/25
   - 200.3.27.0 and 200.3.27.128 mask 255.255.248.0?
2. What is the first terminal of the networks that contain the terminals with the address:
   - 175.0.92.191/23
   - 175.0.92.190/26
   - 175.0.92.18/28?
3. What is the last terminal in the networks:
   - 175.0.32.0 mask 255.255.248.0
   - 175.0.0.0 mask 255.255.224.0
   - 175.0.16.0 mask 255.255.248.0
4. What are the networks of the terminal:
   - 175.0.22.79/25
   - 175.0.117.215/23
   - 175.0.117.215/27?
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
   - 175.0.114.0 255.255.255.240?
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.

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`

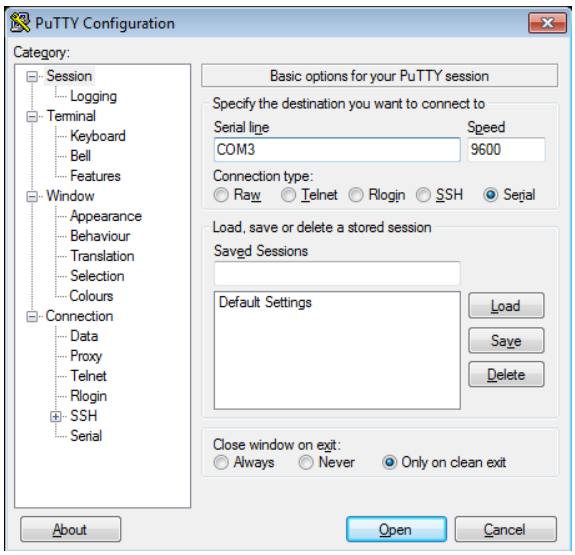
To configure the Gateway of the PC, execute the following command:

```
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
```