universidade de aveiro
theoria poiesis praxis

Arquitetura de Redes

Introduction to GNS3

Objectives

- Study and test of the functionalities of the GNS3 emulator/simulator

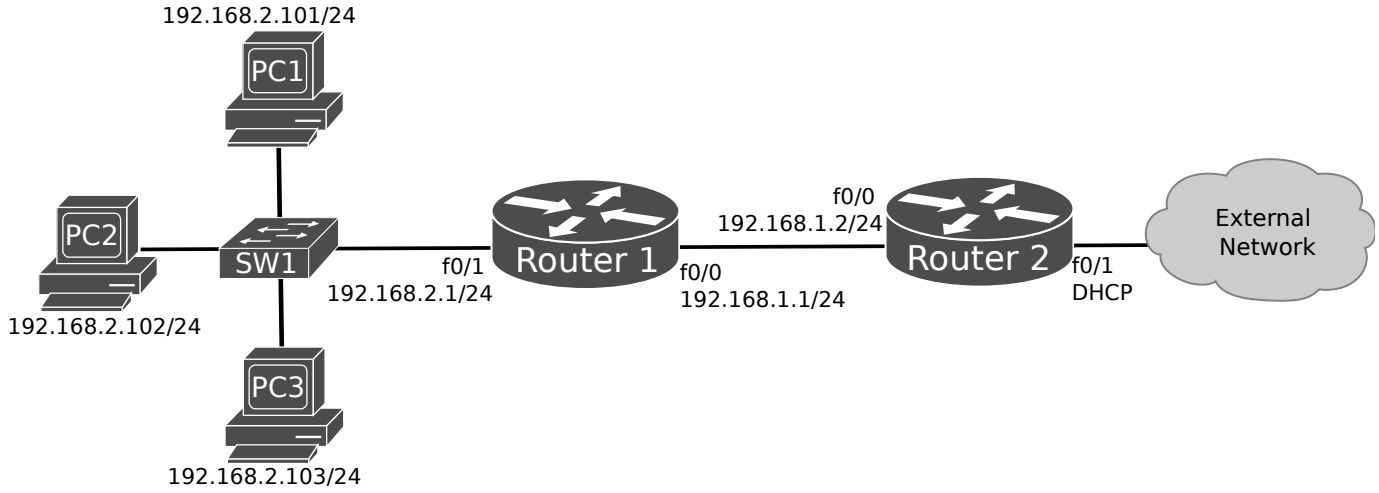
GNS3 Basic Configurations

Choose your operating system (Linux/MacOS/Windows), download/install GNS3 (version>1.3) and related software (Wireshark, VirtualBox) and start GNS3.

1. Go to (Edit-Preferences) and: (i) verify/setup all storing and programs paths, avoiding paths with spaces and non ASCII caracters, (ii) at (Preferences-General-Topology View) choose the option "Always use manual mode when adding links", (iii) at (Preferences-Dynamips-IOS router templates" and create a new router template (recommended: 7200 with IOS 15.1(4) and adapters C7200-IO-2FE+PA-2FE-TX).

GNS projects

2. Create a new Blank Project (File menu or CTRL+N) and give it a name. Add one router (Router1) to your project.



3. Start Router1, open the console of the Router, wait for the command prompt, and search/define the IdlePC value until the load of the PC processor becomes lower (choose the values marked with *). If no value reduces the processor load, search new IdlePC values.
Note: the IdlePC value is applied to all equipments of the same model/firmware. The search/definition of the IdlePC must be made only once per model/firmware.
Note 2: if new slot cards are added to a router, the IdlePC may have to be redefined.

Router configurations

4. Preform some configurations in Router1 (e.g., IP address/mask, activation of interfaces):

```
Router1# configure terminal
Router1(config)# interface FastEthernet 0/0
Router1(config-if)# ip add 192.168.1.1 255.255.255.0
Router1(config-if)# no shutdown
```

Save the configurations in the router(s):

```
Router# write
```

And save the project or right-click the router and save configuration. Analyze the configuration (*.cfg) file created in your project folder.
Note: The *.gns3 and *.cfg files are the only ones required (maintaining the folder hierarchy) to transfer the GNS3 project to another computer or store the project in a repository (e.g., git).

5. According to the above network diagram; add a second router (Router2), connect it to Router1, configure it and test connectivity between the routers.

Capturing traffic

6. Start a capture on the link between the routers, open the capture (if necessary, start wireshark from the "Topology Summary" dock), generate some traffic in that link by performing a ping from a router to another. Analyze the captured packets and verify that the source/destination MAC addresses match the respective MAC address of the interfaces of the routers.
To obtain the MAC address of a outer interface:

```
Router# show interfaces FastEthernet 0/0
```

Note: In Linux, your user must belong to group *wireshark* (a re-login is required):

```
sudo usermod -a -G wireshark username
```

Virtual PCs (vpcs)

7. Add an "Ethernet Switch" (SW1) and a "VPCS device" (PC1), and perform the link connection to Router1. Configure FastEthernet 0/1 interface of Router1:

```
Router1(config)# interface FastEthernet 0/1
Router1(config-if)# ip address 192.168.2.1 255.255.255.0
Router1(config-if)# no shutdown
```

PC1:

```
PC1> ip 192.168.2.101/24 192.168.2.1
```

and Router2 static routing:

```
Router2(config)# ip route 192.168.2.0 255.255.255.0 192.168.1.1
```

Test connectivity between PC1 and Routers 1 and 2.

Interconnection with virtual machines

8. Go to (Edit-Preferences-VirtualBox-VirtualBox templates" and create a new VM template based on an existing VirtualBox machine. Use an Debian LXDE VirtualBox appliance available to download here (login/password: labcom/labcom) or choose another appliance from http://www.osboxes.org/virtualbox-images/ .
Note1: The VM should be powered off and the network adapter should be "not attached".
Note2: To use multiple VM instances, you may clone the original machine.

9. Add PC2 as an end device based on the created VM template. Configure its IPv4 address and gateway, as root do:

```
ip link set up dev eth0
ip addr add 192.168.2.102/24 dev eth0
ip route add default via 192.168.2.1
```

Test connectivity to the other GNS3 network elements.

Interconnection with the host machine

10. (Windows) PC3 should be your host machine. This connection requires a virtual interface. The simplest solution is to use a VirtualBox Host-only network/interface. In VirtualBox interface go to (File-Preferences-Network-Host-only networks) and create a new network (with disabled DHCP server). This step will create a new virtual network interface on your host machine, configure it with PC3 IPv4 address. Define a route to the GNS3 network, using a Command Prompt as Administrator:

```
route ADD 192.168.1.0 MASK 255.255.255.0 192.168.2.1
```

In GNS3, add your host machine connection as an end device "Cloud" and perform network connection using the respective VirtualBox interface. Try to ping Router1 and Router2 from the host machine, and vice-versa.
Note: if a newly created interface does not appear on the "Cloud" list: reboot Windows.

10. (Linux) PC3 should be your host machine.. Verify if the machine has a **tap interface** if not create one:
With the command *tunctl* (part of the *uml-utilities* package): `sudo tunctl -u your_user_name`
Or, with the command *ip*: `sudo ip tuntap add dev tap0 mode tap`
Add a "Cloud" end device into your project. Right-click the Cloud and choose Configure. To connect to the host machine, click on cloud's name under Clouds, choose the **NIO TAP** tab and add a tap interface (usually tap0 or tap1). Configure your host machine tap interface (IP address and mask) and define a route to the GNS3 network:

```
sudo ip link set up dev tap0
sudo ip addr add 192.168.2.103/24 dev tap0
sudo ip route add 192.168.1.0/24 via 192.168.2.1
```

Try to ping Router1 and Router2 from the host machine, and vice-versa.

10. (OS X) PC3 should be your host machine. Install the software available at tuntaposx.sourceforge.net (You may have to reboot). When the installation is complete, you can check if the tap interfaces are available by doing:

```
ls /dev/tap*
```

If they aren't available, either the installation process failed or you need to reboot your laptop before proceeding.Execute the following commands to allow GNS3 to manipulate the tap interfaces:

```
sudo chown $(id -un):$(id -gn) /dev/tap*
sudo chmod 0755 /dev/tap*
```

Add a "Cloud" end device into your project. Right-click the Cloud and choose Configure. To connect to the host machine, click on cloud's name under Clouds, choose the NIO TAP tab and add a tap interface (usually /dev/tap0 or /dev/tap1). Configure your host machine tap interface (IP address and mask) and define a route to the GNS3 network:

```
sudo ifconfig tap0 192.168.2.103/24
sudo route add -net 192.168.1.0/24 192.168.2.1
```

Try to ping Router1 and Router2 from the host machine, and vice-versa.

Note: this method can also be used to interconnect VMs with GNS3. Just define the VMs network adapter as Host-only → vboxnet interface, or Bridged → tap interface.

(Extra) Interconnection with external networks

11. To connect a GNS3 network to other external networks, add a "Cloud" end device into your project. On cloud's configuration interface, choose the **Ethernet Interfaces** tab and add one of the host machine's Ethernet/Wifi interfaces (by default all interfaces should be already added). Connect the cloud to Router 2 using the desired (external) interface.

12. Connect your host machine Ethernet interface to lab's network. Configure Router2 to obtain its FastEthernet 0/1 interface by DHCP:

```
Router2(config)# interface FastEthernet 0/1
Router2(config-if)# ip address dhcp
Router2(config-if)# no shutdown
```

Verify that the router was able to connect and obtain an IPv4 address and gateway from the external network:

```
Router2# show ip interface brief
Router2# show ip route
```