universidade de aveiro
theoria poiesis praxis

Fundamentos de Redes

Introduction to GNS3

GNS3 Basic Configurations

1. Choose your operating system (Linux/Windows), download/install GNS3 (version>2.x) 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).

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 don't need remote servers running on an external virtual machine (VM).

4. 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), 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.

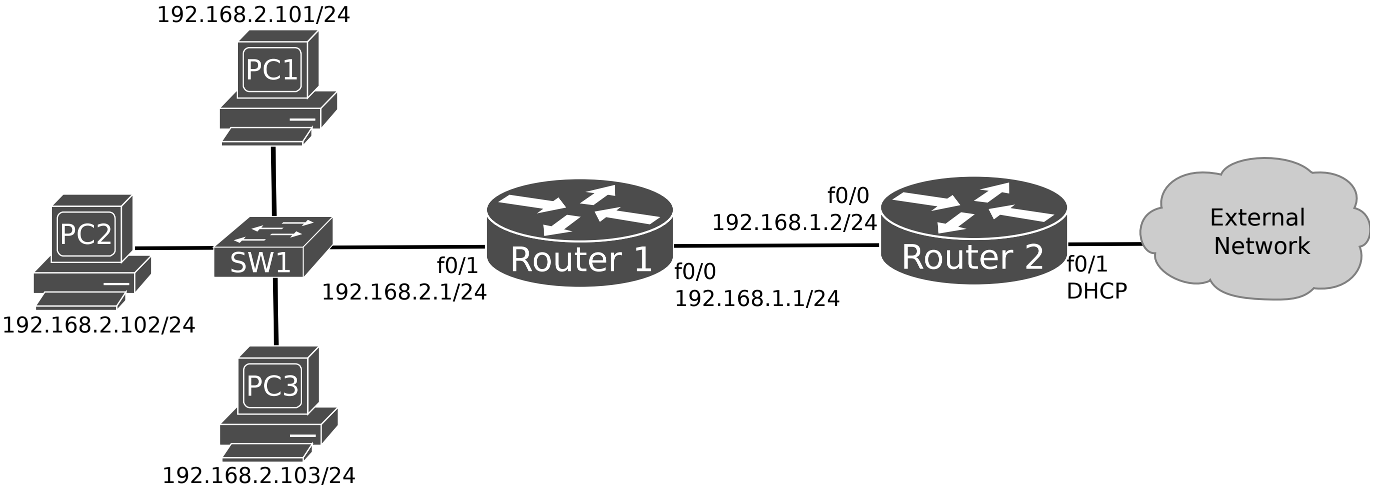
5. The definition of the "Idle-PC" value will allow the host machine to assign the correct amount of resources to the virtual devices. You must repeat this procedures every time your PC CPU reaches values higher than 90%. Check the CPU utilization with the "Task Manager" in Windows, top command in Linux 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.

GNS projects

6. Create a new Blank Project (File menu or CTRL+N) and give it a name.



Routers

4. Add one router (Router1) to your project. 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 an interface:

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

Ethernet Switch and Virtual PCs (vpcs)

7. Add an “Ethernet Switch” (SW1) and a “VPCS device” (PC1), and perform the link connections between PC1, SW1, and Router1.
By default the “Ethernet Switch” has 8 access ports (numbered from 0 to 7) assigned to VLAN 1. The port VLAN and type of port (access or trunk/802.1q/dot1q) can be changed by left clicking the device and choosing configure.
For now, keep all ports as access and VLAN 1.
Note: For trunk ports the VLAN must be the native/default VLAN (use VLAN 1).

8. Configure vpcs PC1 IPv4 address:

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

Use ? to check all available commands. Use the `save` and `load` commands to save/load configurations.

9. 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
```

10. Add a static route to the PCs network in Router 2:

```
Router2(config)# ip route 192.168.2.0 255.255.255.9 192.168.1.1
```

Test connectivity between devices.

Interconnection with virtual machines

11. 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.

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

```
ifconfig eth0 up
ifconfig eth0 192.168.2.102/24
route add default gw 192.168.2.1
```

Test connectivity to the other network elements.

Interconnection with the host machine

13. (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 “Host” and perform network connection using the respective VirtualBox interface. Try to ping Router1 and Router2 from the host machine, and vice-versa.

13. (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
```

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

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

13. (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.

Interconnection with external networks (optional)

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
```