REDES DE COMUNICAÇÕES 1

Objectives

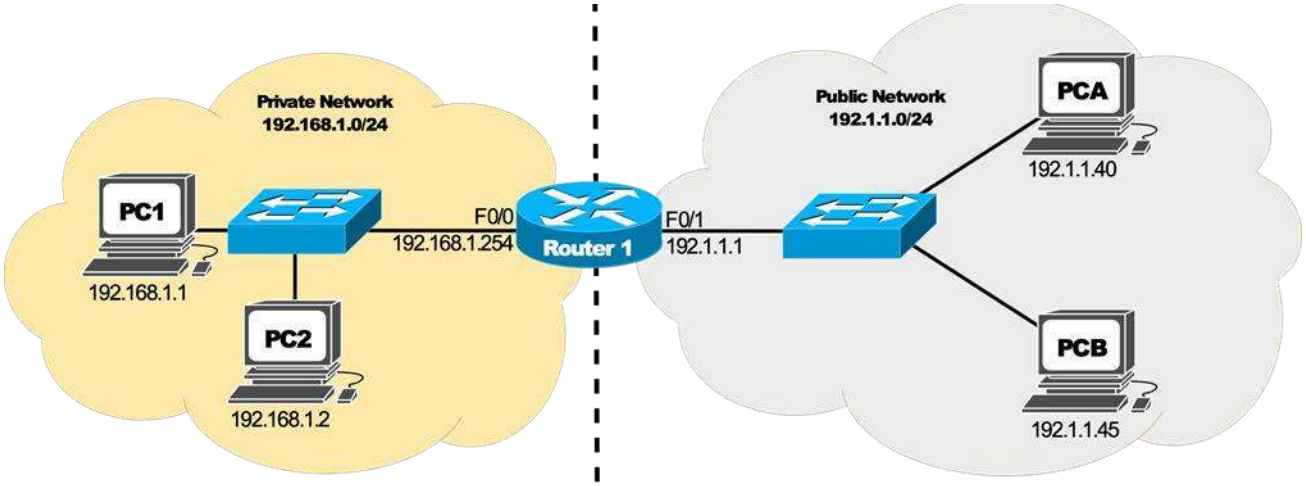
- Study of the NAT/PAT mechanisms.
- Study of DHCP.
- Study of IPv6.

Duration

- 2 weeks

Dynamic NAT

1. Assemble and configure (using the GNS3 and VPCS hosts) the network depicted in the following figure which represents a small company network. The company decided to configure IP private addressing using the network 192.168.1.0/24 and NAT mechanism (without PAT) to manage all Internet accesses. IP addresses and the respective gateway addresses must be manually configured. The company has only 2 public addresses (192.1.1.1/24 and 192.1.1.21/24).
Consider Router 1 with the model 7200.
Configure the network and its addresses before starting the NAT configuration.



Dynamic NAT Configuration

In order to define a pool of global addresses to be allocated by the dynamic NAT process, issue the following command on Router 1:

```
Router1(config)# ip nat pool MYNATPOOL 192.1.1.21 192.1.1.21 netmask 255.255.255.0
```

that defines a pool with a single public address (192.1.1.21).

The name MYNATPOOL is the name of the address pool. The first 192.1.1.21 in the command is the first IP address in the pool, and the second 192.1.1.21 is the last IP address in the pool (this command creates a pool that contains only a single address).
Next, configure a standard access list to define which internal source addresses can be translated. Since any users on the private network are being translated, use the following command:

```
Router1(config)# access-list 2 permit 192.168.1.0 0.0.0.255
```

To establish the dynamic source translation, link the access list to the name of the NAT pool, as shown in the following:

```
Router1(config)# ip nat inside source list 2 pool MYNATPOOL
```

Finally, specify an interface on the Router to be used by inside network hosts requiring address translation:

```
Router1(config)# interface f0/0
                #change the interface name to the one used in your router
Router1(config-if)# ip nat inside
```

Also, specify an interface to be used as the outside NAT interface as follows:

```
Router1(config)#interface f0/1
                #change the interface name to the one used in your router
Router1(config-if)#ip nat outside
```

2. Start a packet capture on the public network and another on the private network. At PC1 execute a ping to 192.1.1.45, and on PC2 execute a ping to 192.1.1.45. Verify (on the router) the active NAT

translations and NAT activity statistics, use the commands

```
Router1# show ip nat translations
Router1# show ip nat statistics
```

>> Which packets had the source IP addresses translated? Explain the obtained results.

3. Execute on the router the command to clear the NAT translation table:

```
Router1# clear ip nat translation *
```

and execute again at PC2 a ping to 192.1.1.40.
>> Explain the observed results.

4. Change NAT timeout to 60 seconds and clear the NAT translations table:

```
Router1(config)# ip nat translation timeout 60
Router1# clear ip nat translation *
```

At PC1 execute a ping to 192.1.1.40, and immediately after, at PC2 execute repeatedly a ping to 192.1.1.40. How much time does it take to obtain connectivity between PC2 and host 192.1.1.40?
>> Explain the observed results.

Restore NAT timeout value to 86400 seconds (24 hours):

```
Router1(config)# ip nat translation timeout 86400
```

Dynamic NAT/PAT

5. The most powerful feature of NAT is address overloading, or port address translation (PAT). Overloading allows multiple inside addresses to map to a single global address. With PAT, the NAT router keeps track of the different conversations by mapping TCP and UDP port numbers.
After defining the pool of global addresses to be allocated by the dynamic NAT process and configuring the standard access list that defines which internal source addresses can be translated, configure address overloading on Router with the following command:

```
Router1(config)#ip nat inside source list 2 pool MYNATPOOL overload
```

Note: You may have to reset the active NAT translations: `clear ip nat translation *`
Repeat experience 2.
>>Which are the advantages of using NAT and PAT mechanisms?

6. From PC1 (and PC2) try to establish UDP and TCP connections (ports 80 and 22) to the host 192.1.1.40:

```
PC> ping 192.1.1.40 -2 -p 80          ! UDP port 80
PC> ping 192.1.1.40 -2 -p 22          ! UDP port 22
PC> ping 192.1.1.40 -3 -p 80          ! TCP port 80
PC> ping 192.1.1.40 -3 -p 22          ! TCP port 22
```

Note: The option -p must have a space after (before the port number).
>> Verify (on the router) the active NAT translations and NAT activity statistics. Explain the obtained results.

Static NAT/PAT Translations

7. Try to ping the private network machines from PCA.

8. Suppose that now you have another public IP address available (192.1.1.201), configure the router in order to allow the PCA to access PC1.
A static translation between the inside local address of a host and one of the inside global addresses can be created using the following commands:

```
Router(config)#ip nat inside source static 192.168.1.1 192.1.1.201
```

From PCA, ping PC1's static public address (192.1.1.201)

```
PCA> ping 192.1.1.201
```

>> Analyze the captured packets on the private network and explain the obtained results.
>> Discuss a scenario where static NAT/PAT is required.

DHCP

9. Configure Router 1 as DHCP server for the private network. Assume that you want to dynamically assign addresses from the range 192.168.1.100 to 192.168.1.200.

```
Router1(config)# service dhcp
Router1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.99
Router1(config)# ip dhcp excluded-address 192.168.1.201 192.168.1.254
Router1(config)# ip dhcp pool 1
Router1(dhcp-config)#network 192.168.1.0 255.255.255.0
Router1(dhcp-config)#default-router 192.168.1.254
```

Use the following commands to verify the configuration and status of the DHCP server:

```
show ip dhcp pool
show ip dhcp server statistics
show ip dhcp binding
```

10. Start a capture on Router 1's F0/0 interface. Configure PC1 to acquire the IPv4 address dynamically:

```
PC1> ip dhcp
```

Configure PC1 to renew the IPv4 address dynamically:

```
PC1> ip dhcp -r
```

Configure PC1 to release the IPv4 address dynamically:

```
PC1> ip dhcp -x
```

Configure PC1 to acquire again the IPv4 address dynamically:

```
PC1> ip dhcp
```

>> In each step, analyze the exchanged DHCP packets and the contents of the DHCP Bindings at Router 1.

IPv6 Basic Mechanisms

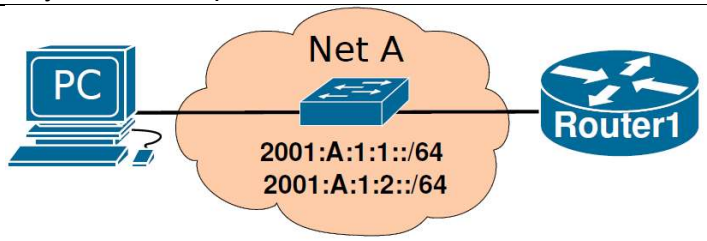
1. Considering the following network, start by connecting the PC (a VirtualBox VM Linux) to the switch without any other connections (check Annex A)
To avoid incompatibilities, disable the Linux network manager (if active):

```
sudo service network-manager stop
```

Note: The commands `sudo service network-manager start` can be used to restart the application/service.
Start a capture in the link between the PC and the Switch. Turn off and on the PC's Ethernet interface:

```
sudo ifconfig eth0 down
sudo ifconfig eth0 up
```

Stop the capture and analyze the IPv6 packets.



2. Connect Router1 to the switch and start a capture in the link between the PC and the Switch. Power on Router1 and configure its interface to network A.

```
Router1(config)# ipv6 unicast-routing
Router1(config)# interface <if-name>
Router1(config-if)# ipv6 enable
Router1(config-if)# no shutdown
```

Verify router's interfaces names and configuration:

```
Router1# show ipv6 interface
Router1# show ipv6 interface brief
```

Restart PC's Ethernet interface and verify its interface information:

```
sudo ifconfig eth0 down              (sudo ifconfig enp4s0 down)
sudo ifconfig eth0 up                (sudo ifconfig enp4s0 up)
ifconfig eth0                        (ifconfig enp4s0)
```

Stop the capture and analyze the IPv6 packets and equipment's information. Use the commands:

```
show ipv6 interface brief
show ipv6 route
```

to verify interfaces' IPv6 addressing and verify router's IPv6 routing table.

3. Re-start a capture in the link between the PC and the Switch. Configure Router's interface with a manually defined IPv6 global address from network 2001:A:1:1::/64.

```
Router1(config)# interface <if-name>
Router1(config-if)# ipv6 address 2001:A:1:1::100/64
Router1(config-if)# no shutdown
```

Verify PC's Ethernet interface information. Stop the capture and analyze the IPv6 packets. Verify the Router's interfaces IPv6 addresses and the router's IPv6 routing table.
>> Explain the process by which the PC obtained the IPv6 addresses.

4 Re-start a capture in the link between the PC and the Switch. Configure Router's interface with a EUI-64 based IPv6 global address from network 2001:A:1:2::/64.

```
Router1(config)# interface <if-name>
Router1(config-if)# ipv6 address 2001:A:1:2::/64 eui-64
Router1(config-if)# no shutdown
```

Verify PC's Ethernet interface information. Stop the capture and analyze the IPv6 packets. Verify the Router's interfaces IPv6 addressing and the router's IPv6 routing table.
>> Explain the process by which the Router completed the last 64 bits of its IPv6 addresses.
>> Discuss a possible disadvantage of using the standard EUI-64 at routers' interfaces.
>> Does the process, by which the PC obtained the IPv6 addresses, change by using the EUI-64 standard at the Router?

5. Re-start a capture in the link between the PC and the Switch. At the PC, using the command ***ping6*** perform a ping to:

a) Router's Link-Local address (you need to define the output interface with option "`-I eth0`" or "`-I enp4s0`").

b) Router's Global address from network 2001:A:1:1::/64.

c) Router's Global address from network 2001:A:1:2::/64.

Stop the capture and analyze the IPv6/ICMPv6 packets.

>> Explain the physical addresses resolution process in IPv6.

Annex A

Interconnection with virtual machines (VirtualBox)

Go to (Edit-Preferences-VirtualBox-VirtualBox VMs" and create a new VM template based on an existing VirtualBox machine. Use a Debian LXDE VirtualBox appliance available to download in the e-learning (login/password: labcom/labcom).
Note1: To use the VM in GNS3, the VM should be powered off and the network adapter should be "not attached".
Note2: To connect the VM to the Internet, start the VM from VitualBox GUI with the network adapter attached to "NAT".
Note3: To use multiple VM instances, you may clone the original machine.

To add a PC as an end device based on the created VM template, configure its IPv4 address and gateway, as root:

```
ip link set up dev enp1s0
ip addr add 192.168.2.102/24 dev enp1s0
ip route add default via 192.168.2.1
```

Test connectivity to the other GNS3 network elements.
Note: your virtual Ethernet port may have another name. List devices with `ip addr` to identify it.

Interconnection with virtual machines (QEMU)

Go to (Edit-Preferences-QEMU-QEMU VMs" and create a new VM template based on an existing virtual disk image (*.img). Use a Debian LXDE QEMU virtual disk (LabComServer2.qcow2) available to download in the e-learning (login/password: labcom/labcom). Choose console type "none".
Note1: To use the VM in GNS3, the VM should be powered off.
Note2: To connect the VM to the Internet, start the VM from the command line (or *virt-manager*) using the command "`qemu-system-x86_64 -m 1024 -enable-kvm LabComServer2.qcow2`".
Note3: To use multiple VM instances, you may copy the original VM disk file "LabComServer2.img" and start another VM.
Note4: In Windows, QEMU requires HAXM, see how to install here. Also, replace option "-enable-kvm" with option "-accel hax" when running from the command line.

To add a PC as an end device based on the created VM template, configure its IPv4 address and gateway, as root:

```
ip link set up dev enp1s0
ip addr add 192.168.2.103/24 dev enp1s0
ip route add default via 192.168.2.1
```

Test connectivity to the other GNS3 network elements.
Note: your virtual Ethernet port may have another name. List devices with `ip addr` to identify it.

Connect an Ubuntu VM to GNS3 in a MacBook M1 (using VMware)

- Verify if you do not have issues running GNS3. It should run normally, as it uses Rosetta x86 emulation.
- Install the free version of VMware Fusion Public Tech Preview. Download from https://customerconnect.vmware.com/downloads/get-download?downloadGroup=FUS-PUBTP-2021H1
- Download Ubuntu 20.04 Arm from https://cdimage.ubuntu.com/focal/daily-live/current/focal-desktop-arm64.iso
- Create Ubuntu Arm VM in VMware using the downloaded Ubuntu image

- Close VMware. On your Mac, go to /Applications folder and rename the app “VMware Fusion Tech Preview” to “VMware Fusion”
- Open GNS3, go to the VMware tab in Preferences, and import the new Ubuntu image.

To add a PC as an end device based on the created VM template, configure its IPv4 address and gateway, as root:

```
ip link set up dev enp1s0
ip addr add 192.168.2.103/24 dev enp1s0
ip route add default via 192.168.2.1
```

Test connectivity to the other GNS3 network elements.
Note: your virtual Ethernet port may have another name. List devices with `ip addr` to identify it.