Universidad Carlos III de Madrid

Departamento de Ingeniería Telemática

Computer Networks

IP Address Assignment Exercise

Computer Science

1. Goal

The main objective of this exercise is to get used to the problem of designing and assigning IP addresses in a given networking scenario, with room for customization.

It is **NECESSARY TO PROVIDE A DESIGN SOLUTION BEFORE THE CLASS WHERE THE EXERCISE IS SOLVED** (using Aula Global). It is important to realize that **NO routing plan will be provided** in this assignment; hence you must create one **based on the design choices you make during this exercise**

This exercise may be done alone or in couples. It is important to read through the entire document before starting the IP addressing process, to take all the requirements into account for your design.

Once the network design is ready, you will proceed to implement it under **CORE program**, to check that everything is correct before starting the next assignment (routers lab).

2. Problem description

You are in charge of interconnecting the equipment of a large company. The company has several branches, all connected to each other. Each branch has a given number of devices (computers, printers, routers, etc.) to be connected through the network. Branches communicate with each other through a backbone network that connects to the router of each branch.

2.1. Global requirements

The enterprise "owns" the **10.0.0.0/16** address range, which is divided between the branches as follows.

- All the branches are connected to a backbone network that must be addressed within the **10.0.0.0/24** range.
- Each branch has **10.0.X.0/24** range, where '**X**' indicates the number of the branch. For example, the branch 1 has 10.0.1.0/24 range, the branch 2, 10.0.2.0/24 range, and so on.
- Every group must ONLY DESIGN ONE BRANCH and will use the **last two digits of the NIA** of any of the group components as the **value for 'X'**.

2.2.Requirements for the design of the branch networks

- Routers can be configured with a maximum of 5 LAN interfaces. It is not necessary to use them all (some of them will remain unused).
- The branch to be designed has three units, namely, the **Office 1, Office 2** and the room that hosts the various **servers** used by the branch. Due to the nature of the various tasks performed in each unit, it has been decided to have one **separate network** for each one.
- The network for **Office 1** must be able to accommodate up to **120 terminal devices** (PCs, printers, etc.).
- The network for **Office 2** must be able to accommodate up to **27 terminal devices** (PCs, printers, etc.).
- The network dedicated to the **servers** must be able to accommodate up to **12 terminal devices**.
- The routers of the three branch networks must be interconnected together with a dedicated **point-to-point link** between them.
- Each one of the routers of the office networks' is connected through a dedicated point-to-point link with the router of the network that hosts the servers.
- The connection of the branch to the central backbone is done through a separate router, which connects via a dedicated line to the router that serves the network hosting the servers. It is not necessary for this practice to configure a default route for this router (but it IS necessary for the rest of the routers).
- The topology design should provide **redundancy** against faults / broken links. Please note that redundancy must also be taken into account when configuring the routing tables (redundant routes are required in the forwarding table).
- It is necessary to also assign addresses (and the corresponding routes) to the networks that are "only" used to interconnect the different routers.
- It is mandatory to perform the addressing from the largest range (i.e. starting with the networks that require a larger number of end systems) to the lowest. That is, SUCCESIVELY assigning addresses starting with the 10.0.X.0 using the corresponding prefixes in the following order: /24, /25, /26,… depending on the design. This way, the unused addresses with remain at the end of the range.

2.3.Design methodology

The steps for the realization of this project are the following:

1. Design the network topology, defining the **number of subnets** needed.
2. Define the **number of addresses** required for each subnet.
3. Assign the **minimal required range** of addresses for each subnet.
4. Define the **routing tables** of all devices. Routers must be configured to take advantage of the existing **physical redundancy**, protecting services against link failures.
5. **Check the design under CORE program**. Design will be implemented in CORE and the correctness of the connections among the elements will be checked by using **ping/traceroute**.

3. CORE program

CORE is a tool that allows configuring and emulating complex networks. It offers a graphical interface to design the network, include the different elements (hosts, routers, etc.) and interconnect them. Each network component is run as a different light virtual machine so when the network is initialized, each element can be accessed as if it were a real equipment (in fact they are independent virtual equipment with their own operating system, etc.).

You can download and install CORE in your personal computer (this is the recommended option) or use the installation available in the Telematic Engineering Department lab.

You can check the installation details in the web page of this course.

Once the virtual machine has been installed and initiated, click in the CORE icon that you may find in the desktop to start the application. Before starting to implement your design take into account the **following considerations**:

1. Each device is independently configurable (IP, address and routing table):

    a) **IP** addresses: right-click on the device → configure. Modify the IPv4 of each interface depending on your design.
    b) **Routing tables**: right-click on the device →services→icon on the right of StaticRoute (Figure 1). Write all the command required to introduce the routes for the device and click on Apply. Commands introduced here will be automatically executed upon simulation start, generating the routing tables for the device.

2. Once **ALL** IP address and routing tables have been assigned, you can proceed to emulate the network.

3. **EMULATION**: after pressing emulation button (play), wait until the network becomes stable (green squares disappear).

4. **CHECK ROUTING TABLES**: during the emulation, check by commands ip route/traceroute/ping that everything is correct. Execute these commands on an individual terminal for each device: **right-click on the element →shell window →bash**. You can also use the CORE tool 'Two Node' integrated in the left navigation panel. However it is highly recommended to use the terminal directly since this is what will be later on available during the lab practice with real routers.

5. **SAVE CONFIGURATION FILE**. It is possible to save all the configured network (topology, routes, etc.) in an IMN file. To do this: **File → Save as imn**
   IMPORTANT: under these terminals it is possible to configure the devices graphically (double-clicking on them) or as if they were real network elements (accessing with a terminal with double click during the emulation and modifying by means of the corresponding commands the IP addresses, routing tables, enable and disable interfaces, execute ping or traceroute). However, all the different changes performed during a emulation (with the terminal) are just temporal and they are lost when stopping it. If you want to make the changes permanent (so that you can store them in the IMN file), you must do them using the graphical tool.

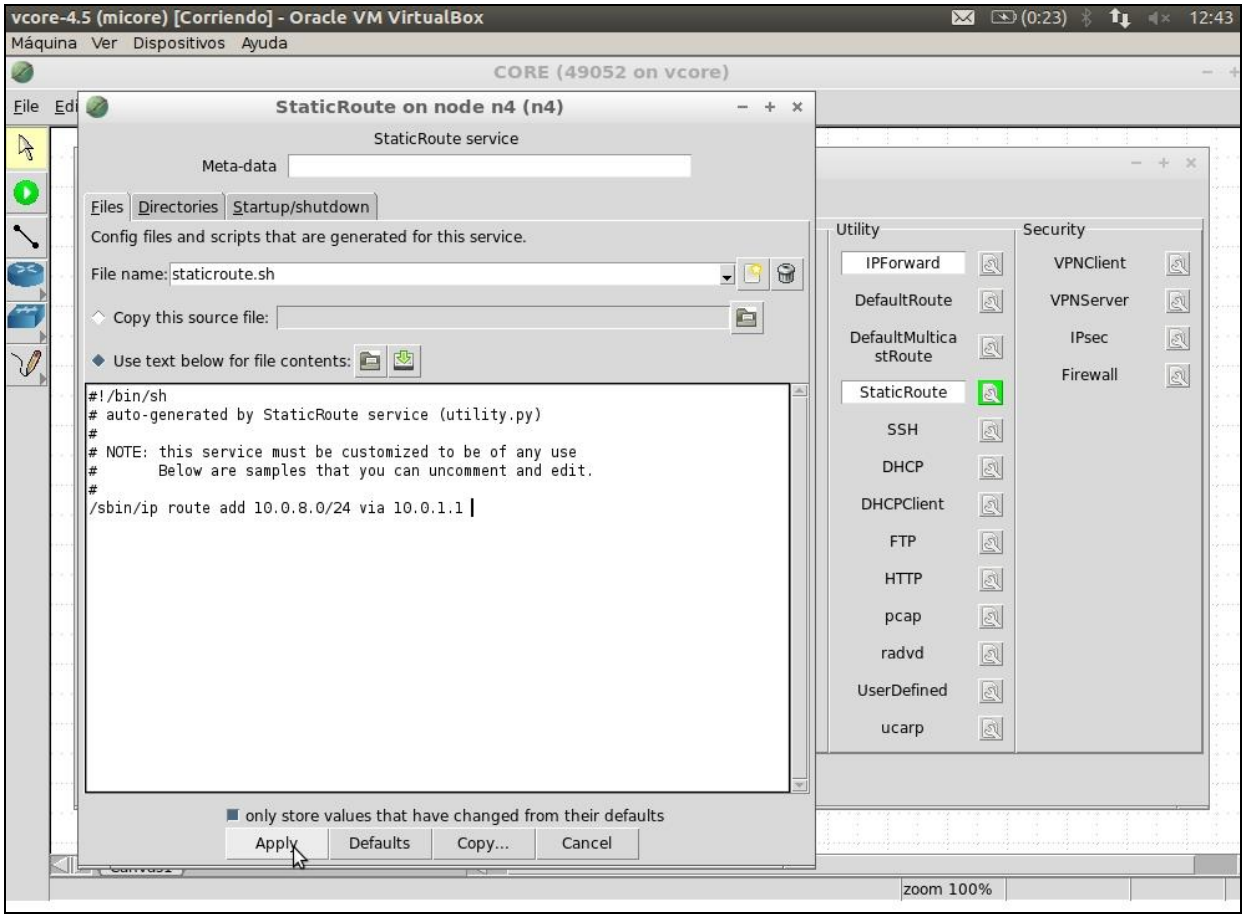

Figure 1: Example to add a route with Static Route

4. Requirements

To pass this assignment, you must perform the following tasks:

1. Design the **physical topology** of a branch network.

2. **Allocate IP addresses** to a branch: include the full addressing plan, i.e., addresses and masks (prefixes) associated with each subnet being deployed.

3. The assignment of IP addresses must be performed in the most efficient way (i.e., address blocks should be left unassigned for future use, whenever possible).

4. Obtain the **routing tables** of routers and end systems connected to each network.

5. Save the implementation (IP addresses and routing tables assigned) in an **.imn file** and prepare a **.pdf file** enclosing the screenshots for each of the previous points as described in the following subsections.

4.1.Branch topology

It is mandatoy to show:

- All the routers
- 1 terminal per independent network (it is NOT necessary to include all the terminals of each subnet)
- IPv4 addresses of each interface
- Address range associated to each independent network (do not forget to add the interconnecting networks' ranges too)

CORE: View → Show → Everything but IPv6 and Link Labels

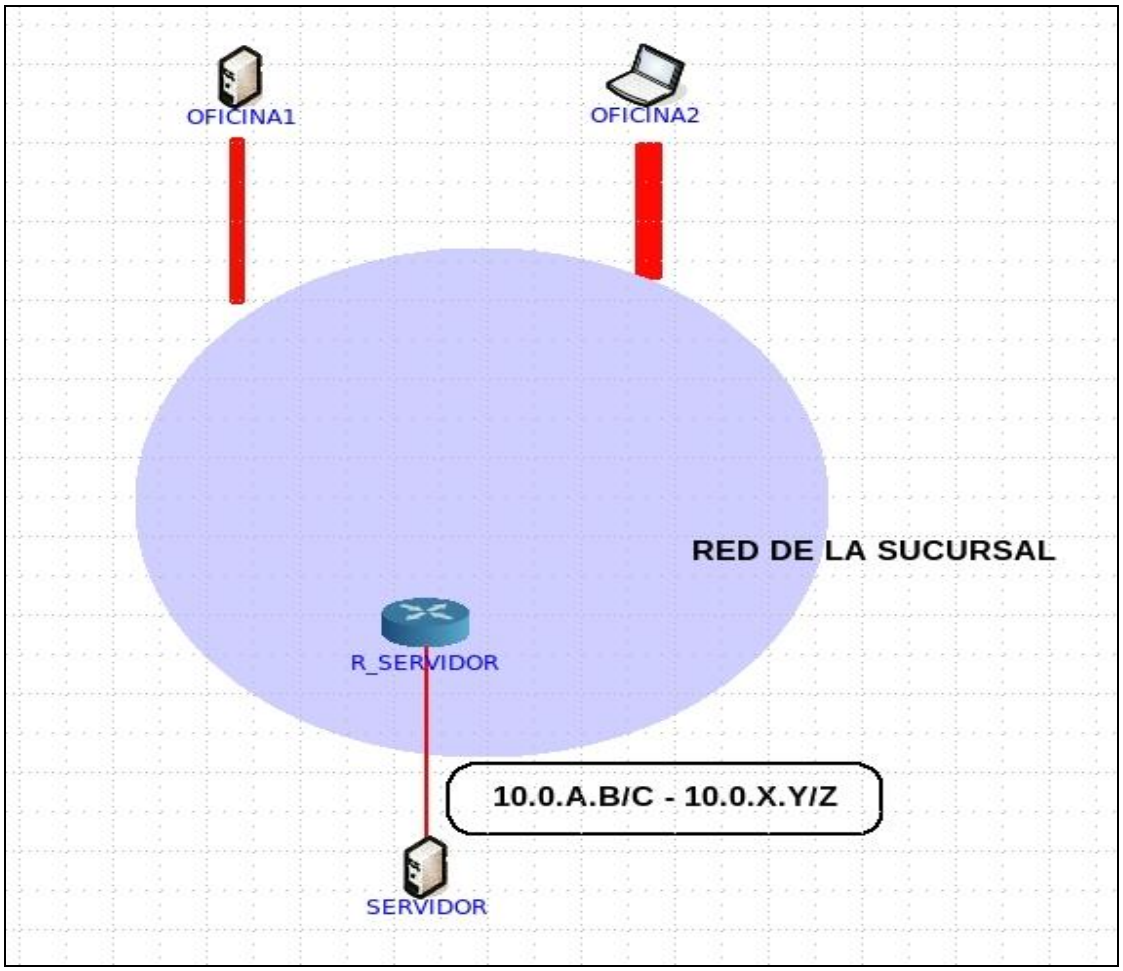

Figure 2: Screenshot example for the network topology (hidden network)

4.2.Routing tables

It is mandatory to show a screenshot per device, **showing the routes** defined in it as done in Figure 3.

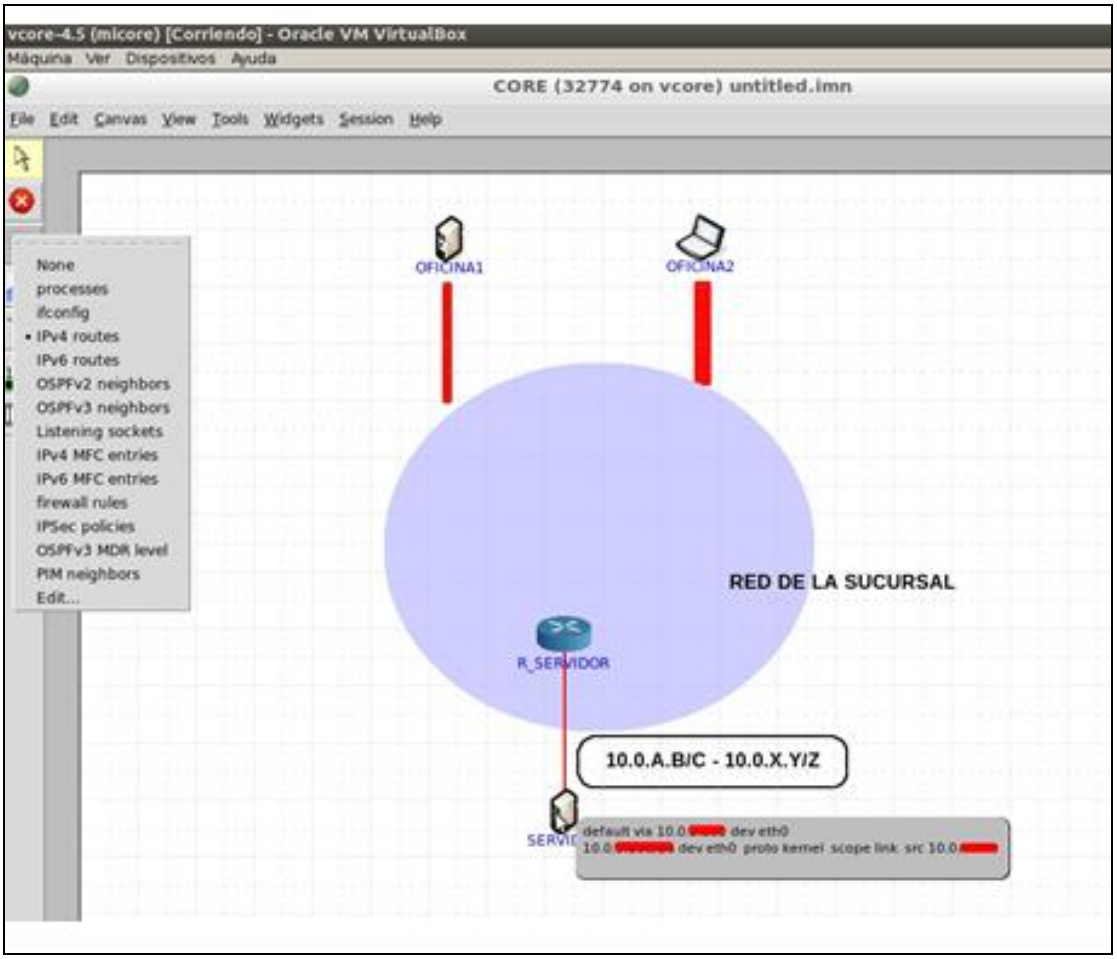

Figure 3: Screenshot example for the routing table of a device in the server network.

4.3.Traceroute command

It is mandatory to show s screenshot of the execution of the **traceroute** command from an element in **Office 1** and an element in the **server network** (both a screenshot of the Two-node Tool or a screenshot of a traceroute performed from the terminal is fine).

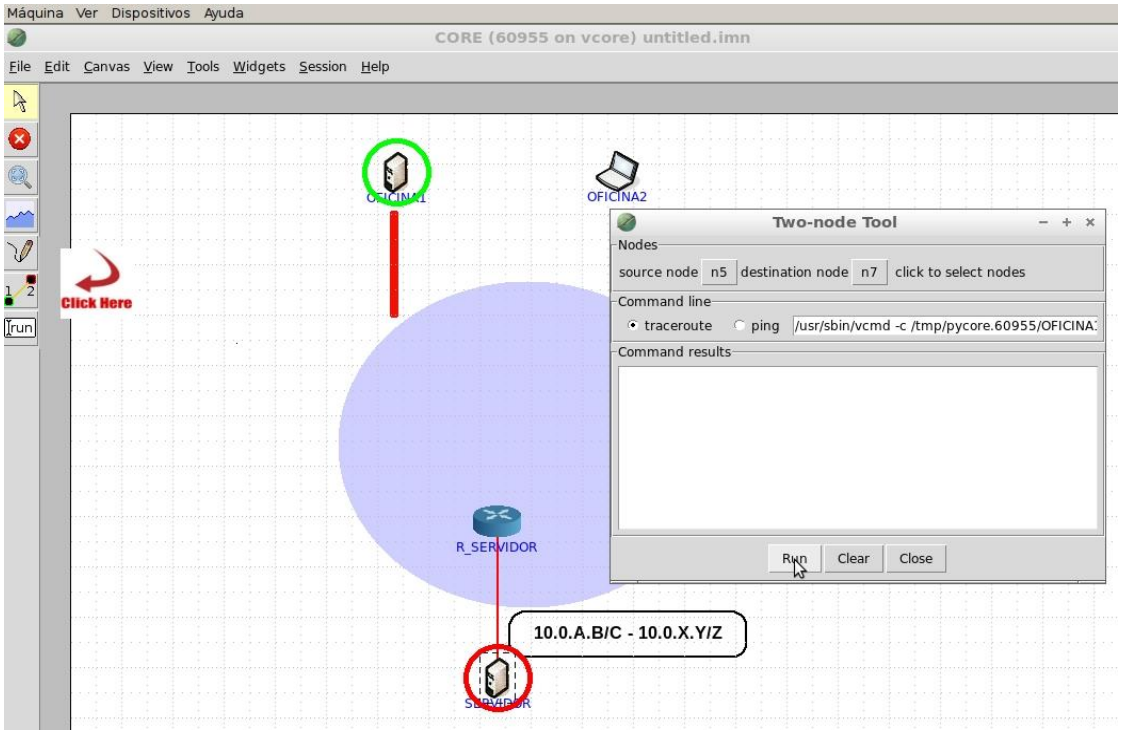

Figure 4: traceroute screenshot example

4.4.Save .imn file: File → Save as imn...

Once all these tasks have been completed and the emulation has been stopped, you have to **save the file in *.imn* format**. This file must be uploaded to the activity enabled for it in Aula Global, along with the *.pdf* file with the screenshots (upload them all into a zip file).

5. Delivery

The *.imn* and the *.pdf* files together with the screenshots specified in section 3 will be delivered through Aula Global. The name of the uploaded file should follow this format:

RO-PdC3-*[Campus][Group]-[NIA].[imn|pdf|zip]*

where NIA is the NIA used to configure the branch.

Valid example: **RO-PdC2-L81-100055221.zip**

It is **NOT** necessary to deliver any physical memory.

NOTE: the exercise will **only** be delivered by one student (the one whose NIA is included in the name of the file that is the same used to configure the branch). Obviously in the cover page of the .pdf delivered, both names and NIAs will be included if the exercise have been done in couples.