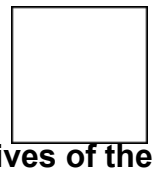

Practice 3

bjectives of the practice

- Network configuration on Linux and Windows systems

rior knowledge:

- ☐ Networks: ranges and masks.
- ☐ IP routing and routing tables.

ntroduction:

To configure a device (end node) in a network generally requires 4 parameters:

- **IP of the equipment**: used to identify the equipment.
- **Mask**: indicates with which equipment it can communicate directly.
- **Gateway**: to know to which computer (or computers) you need to send to in order to leave your network.
- **DNS server**: to convert names in text to IPs.

Usually this information is obtained automatically through a request to the DHCP server, but in this practice we are going to see how all these parameters can be configured on Linux machines and how it influences the network messages we can send.

You will also learn how to examine the Windows configuration via the command line.

Task 1: Network information and configuration commands on Windows and Linux

Step 1: Configure the virtual machine

Run **VirtualBox** and browse to the pl-ubuntu-17.10-gnome-amd64_2022-2023 virtual machine. Configure the network interface of the virtual machine in *Bridge Adapter* mode, and make sure that the active interface is the real network interface. Also in the *Advanced* section (in the same network section) click the button to choose a new random MAC. After configuring it, start it. Once booted, access the graphical interface as **usuario** (password: **usuario**) and in a terminal change to the **root** user with the command `sudo su` (password: **usuario**).

Next, stop the network service from interfering by running `systemctl stop NetworkManager` on the terminal and edit the `/etc/resolv.conf` file and delete the lines beginning with `nameserver`.

Step 2: Obtaining network information on Windows and Linux

Exercise 1. The Windows `ipconfig` and Linux `ip address` commands display information about the machine's network interfaces. Run those commands on a Windows console and on the Linux virtual machine, look up its physical interface information and identify its associated IP, mask and gateway, and fill in the table below.

	Windows	Linux
IP	*192.168.166.41*	*N/A*
Mask	*255.255.252.0*	*N/A*
Gateway	*192.168.167.254*	*N/A*

Is the Linux of the virtual machine on the same IPv4 network as the Windows of the host machine? Why?
No, because right now we don't have neither an IP nor a Mask nor a Gateway as we are using a bridge adapter.

What does the "`lo`" interface on the virtual machine refer to (look up information on the Internet)?
Is a special, virtual network interface that the computer uses to communicate with itself. It is used mainly for diagnostics and troubleshooting, and to connect to servers running on the local machine.

Step 3: Linux network configuration

Exercise 2. If we want the Linux machine to have the same IP as the Windows machine, but changing the second most significant bit of the hosts reserved part of its address,
What should the Linux IP be? *192.168.167.41*

Configure the IP and subnet mask on Linux with the following command:

- `ip address add <dirIP>/<prefix> dev <device>`

  ip address add 192.168.167.41/22 dev enp0s3

where the device value will be **`enp0s3`**, the dirIP value will be the one you just calculated; and the mask will be the same as in Windows. Run the **`ip address`** command again (without any additional parameters) to see what changes have occurred and that everything is correct.

Exercise 3. Now try to `ping`[1] from Linux to the loopback IP (127.0.0.1), to the IPs of your machine (Windows and Linux), to the Professor's IPs (Windows and Linux) and to a machine outside the network (try both by name **www.lcc.uma.es** and by IP: **150.214.108.11**).

Which ones work and which ones don't work?
I can't ping from Linux to both machines outside the networks. The rest of pings are allowed.

Observe the routing table of your Linux virtual machine with the **`ip route`** command.

How does this table explain why some of the above pings work and others do not?
Because the last two pings are made to IPs outside the network and the only entry on the table is inside the network.

In addition to consulting the routing table, we can modify it with the **`ip route`** command:

I. Add a default routing entry using **`ip route add default via <Gateway>`** (as gateway value use **192.168.167.254**). Retest the pings that failed in exercise 3 and discuss why some of them now work and some did not before.
Now, the ping to 150.214.108.11 works, but not the ping to www.lcc.uma.es. We are actually sending the message to our router and the first ping is in the routing table of this router.

II. Finally, edit the **`/etc/resolv.conf`** file and add the line **`nameserver 150.214.57.7`** at the end as the DNS server. Why do you think all the pings are working now, and why do you think the ones that failed before are working?
Now, all the pings work, because we have an entry on the ip table that goes to a machine that can route us to www.lcc.uma.es.

Step 4: Routing on Windows

Exercise 6. When a message is sent outside your local network, two queries are made to your routing table:

- First, we look for the entry that takes us to the final destination. Being external, the default entry will be chosen, which tells us to send to the gateway (your router).
- Then we look for the entry to reach our router (the entry that allows us to communicate with the equipment in our network) which will tell us that this communication can be done by direct delivery.

Look at the Windows routing table with the **`route PRINT -4`** command. Which rows in that table represent the following entries?

a) Input that allows it to communicate with a computer on its own physical network (different from yours).
192.168.167.255 255.255.255.255 En vínculo 192.168.166.41 281
b) Default entry.
0.0.0.0 0.0.0.0 192.168.167.254 192.168.166.41 25

Step 5: Obtaining more information

Another useful command that gives us information about the connections is **`netstat`** (it works on both Windows and Linux). This command gives you information about the connections you have on your computer and even which programs are generating them. This is covered in Topic 4. It also allows you to examine statistics on messages sent and received using the **`-s`** option. Run the command **`netstat -s`** in Windows and by analysing the IPv4 and

[1] `Ping` with the `-c 1` option to send only one ICMP message.

ICMPv4 (ICMP over IPv4) statistics, answer the following questions (a brief description of each statistic can be found at this link). It is recommended to take a screenshot and analyse the data on it as the statistics change over time:

```
Estadísticas de IPv4

  Paquetes recibidos                    = 448991
  Errores de encabezado recibidos       = 0
  Errores de dirección recibidos        = 1737
  Datagramas reenviados                 = 0
  Protocolos desconocidos recibidos     = 0
  Paquetes recibidos descartados        = 23631
  Paquetes recibidos procesados         = 235840
  Solicitudes de salida                 = 109339
  Descartes de enrutamiento                     = 0
  Paquetes de salida descartados        = 241
  Paquetes de salida sin ruta           = 37
  Reensamblados requeridos              = 2968
  Reensamblados correctos               = 989
  Reensamblados erróneos                = 0
  Datagramas correctamente fragmentados = 0
  Datagramas mal fragmentados           = 0
  Fragmentos creados                    = 0
```

a) [IPv4 Statistics] What statistic tells us how many packets we have discarded and what percentage of the total number of datagrams received?
The “paquetes recibidos descartados” & “paquetes de salida descartados”. 5.3168% of packets are discarded.

b) [IPv4 Statistics] What statistic does the CHECKSUM error detection pick up? How many potential packets have that error?

```
Estadísticas ICMPv4

                                    Recibidos  Enviados
  Mensajes                           141         307
  Errores                            0           0
  Destino inaccesible                121         119
  Tiempo agotado                     0           0
  Problemas de parámetros            0           0
  Paquetes de control de flujo       0           0
  Redirecciones                      0           0
  Respuestas de eco                  16          4
  Ecos                              4           184
  Marcas de tiempo                             0           0
  Respuestas de marca de tiempo                0           0
  Máscaras de direcciones            0           0
  Máscaras de direcciones respondidas  0         0
  Solicitudes de enrutador    0                0
  Anuncios de enrutador       0                0
```

c) [ICMPv4 Statistics] What type of ICMP error reporting messages have you received?
Unreachable destination.

d) [ICMPv4 Statistics] Indicate which commands you could use on your machine to force more ICMP responses of at least two different types to be generated.
ping command with the flag to set the max TTL

APPENDIX 1: Useful Commands:

Commands related to network elements that are usually installed by default on most computers. This is not an exhaustive list, and there are many others that usually require explicit installation.

To get help from them you can use the **man** (Linux and Mac) or put the **/?** option (Windows). Commands marked with * on Windows mean that they may not be installed by default.

ping (Windows, Linux and Mac):
- Allows sending an ICMP message to a remote machine to check if it is active and the times.
- Caution: ICMP packets are often filtered by intermediate nodes and not receiving a reply does not mean that the remote machine is not active.
- By using options, certain parameters of the sent IP datagram can be set: time-to-live, size, non-fragment bit...

tracert (Windows) **traceroute** (Linux and Mac):
- It provides a list of the intermediate nodes through which it passes.
- Caution: when * is displayed, it indicates that the intermediate node is not providing information.

netstat (Windows, Linux and Mac):
- Provides information on the use of various network protocols and connections (Topic 4).

arp (Windows, Linux and Mac):
- Allows querying and modifying the ARP cache table (equivalences between logical and physical addresses).

ipconfig (Windows) **ifconfig** (Linux and Mac):
- Allows you to consult and modify the basic configuration of the network.

route (Windows, Linux and Mac):
- Allows the routing table to be consulted and modified.

ifconfig/arp/route/netstat (Linux):
- Old tools for configuring and obtaining network information (now replaced by the ip command).

netsh (Windows):
- It allows you to consult and modify a large part of the system's network elements.

nslookup (Windows, Linux and Mac):
- Allows for very basic DNS queries (Topic 5).

dig (Windows*, Linux and Mac):
- Similar to **nslookup** but allows much more detailed queries and provides more information.

ssh (Windows*, Linux and Mac):
- Allows remote and encrypted connection to a remote computer. Once connected you can execute commands on the remote computer (Topic 5).

telnet (Windows, Linux and Mac):
- Similar to **ssh** but communications are used unencrypted which makes it very vulnerable.
- Another use is to allow TCP connections (Topic 4) to any port and use it to access/debug certain services.

curl/wget (Windows*, Linux and Mac):
- Allows downloading resources via HTTP (Topic 5) through the command line.
- It is very useful for automating access to REST services or web pages.