Sistemas Informáticos (Computer Systems)

Unit 11. Computer networks - Part 02

Authors: Sergi García, Alfredo Oltra

Updated March 2024

Licencia

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

📖 **Important**

❕ **Attention**

💬 **Interesting**

INDEX

Unit 11. Computer networks - Part 02

1. Connecting computers to a network

The first step to be able to use devices in a network is to connect them to it, and for this we need to know the available interfaces (NICs) and assign them an IP address. There are two ways to assign IPs, either dynamically or statically:

- **Dynamic**: A network device (a DHCP server) manages the IP distribution, When the device is connected to that network, it requests an IP to the DHCP server that assigns it based on certain rules. In this way, the incorporation of new devices is faster, and possible conflicts are avoided by assigning IPs equal to different nodes of the network). However, it is possible that between different connections to the network, the assigned IP is different
- **Static**: The IP of each device must be assigned manually. This complicates the addition of new devices and increases the probability of conflicts, but allows the IP of a computer to stay fixed over time.

1.1 Dynamic assignment

Because of its simplicity, it is the most common type of assignment within a network. It is used when we connect with our mobile to a wireless network (Wi-Fi or 4G) or with our desktop in our home network. In general, the router is the DHCP server.

📢 it is the one that is configured by default in the majority of operating systems that work as workstations, so that the user only has to connect the equipment to the network physically.

1.1.1 Linux systems

The first step is to know how many and which interfaces are available on our computer. To do this, from the terminal, we use the *"ifconfig"* command.

📖 **Important:** One of the most important Linux commands (as far as networking is concerned) is definitely *"ifconfig"*. With it, it is possible to configure and modify the configuration of the network interfaces.

With the *"-a"* option, we are shown information about all the interfaces that exist in our system. Each of them is named with two or three letters, followed by a number (similar to hard disk nomenclature). In the case shown in the figure there are 4 interfaces, 3 of them physical[1] (that is, they refer to hardware elements), and 1 logical *lo0*, which refers to the *loopback*, to the internal loop (others may appear, such as *vbox*, which refers to interfaces created by *virtual box*).

[1] en0, en1 referring to the ethernet interface and fw0 referring fire-wire interface

```
> ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
          options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
          inet 127.0.0.1 netmask 0xff000000
          inet6 ::1 prefixlen 128
          inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
          nd6 options=201<PERFORMNUD,DAD>
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
          options=10b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV>
          ether 10:9a:dd:71:1d:c6
          nd6 options=201<PERFORMNUD,DAD>
          media: autoselect (none)
          status: inactive
en1: flags=963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX> mtu 1500
          options=60<TSO4,TSO6>
          ether d2:00:1c:56:cf:c0
          media: autoselect <full-duplex>
          status: inactive
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
          lladdr 70:cd:60:ff:fe:c5:6c:fc
          nd6 options=201<PERFORMNUD,DAD>
          media: autoselect <full-duplex>
          status: inactive
```

Once we know the name of the interface to which we want to associate a dynamic ip (in our case we will choose en1), we must modify the file */etc/network/interfaces*

```
> sudo nano /etc/network/interfaces
```

And adding (or modifying if it already existed) a line that configures the interface. In our case we will modify the interface en1, so we would add

```
iface en1 inet dhcp
```

The last step is to restart the interface. We can do this in two ways: disabling and enabling the interfaces directly with the *ifconfig* command:

```
> sudo ifconfig en1 down
> sudo ifconfig en1 up
```

Or by stopping and booting the network system with the network system boot script[2].

```
> sudo /etc/init.d/networking stop
> sudo /etc/init.d/networking start
```

[2] A script is a program written using a command system, which usually aims to automate tasks

❕ **Attention:** There is a simpler way to restart the script using the restart parameter, but in some cases it may fail, so the safest option is to perform the process separately

📖 **Important:** Start scripts are scripts that are executed at the start of the operating system. Its objective is to start in a controlled way programs and daemons[3] that perform tasks in the system (for example, a database server or the control of the network).

In systems with GUI, it is possible to perform these actions graphically. The exact form depends on the system itself, but in general the idea is the same. To see an example, this video of how to configure a client in LUbuntu Desktop: https://www.youtube.com/watch?v=5rbeJTHzN5M

📖 **Important:** As you can see, the address of the DHCP server is nowhere indicated. How does the computer know who to request IP? The process works with a series of broadcast signals called *DHCP Discover, DHCP Offer, DHCP Request, DHCP ack*[4] that are performed using the 0.0.0.0 client address and as a destination 255.255.255.255 (broadcast).

1.1.2 Windows systems

If you want to know how many and which interfaces are available on our computer in a Windows system, you can use the command "*ipconfig*".

```
> ipconfig
> ipconfig /all
```

You can configure a dynamic IP in Windows systems following those steps from this video: https://www.youtube.com/watch?v=kAEZDUV9VuI

1.2 Static assignment

The configuration of the static IP is not very complex, but it does require knowledge about the topology of the network. As in the dynamic configuration, DHCP handles all the work, in this case it is the user who has to configure all the data manually.

The data we need are the IP to be assigned (taking into account that it is not already used by any other equipment), the network mask and the address of the gateway, i.e., the router that will output the outside to that net.

1.2.1 Linux systems

Similarly to the dynamic assignment, we modify the file */etc/network/interfaces* but in this case adding to information necessary for the static operation:

```
iface en1 inet static
address 192.168.20.5
netmask 255.255.255.0
gateway 192.168.20.1
```

To do it graphically, a LUbuntu example: https://www.youtube.com/watch?v=ccBgbiju_DM

[3] In Linux, a daemon is a service, that is, a program that is executed in the background, in a transparent way to the user. For example an antivirus

[4] http://www.thegeekstuff.com/2013/03/dhcp-basics/

1.2.2 Windows systems

In Windows systems, to configure static IP, you have to go to the same place that you went to configure dynamic IP.

In that place, instead of choosing *Obtain IP automatically*, you should introduce manually IP, network mask and gateway. You can see the process in this video:

https://www.youtube.com/watch?v=kSHunPYosi0

2. Locating resources on the network

Each device connected to an IP network has an IP address, but remembering that set of numbers to be able to communicate between devices is something that is very complicated. Name resolution is the process of mapping IP addresses to host names, making it easier to identify resources on a network. For instance, it is easier to remember www.google.com than 216.58.211.238.

2.1 Assigning your computer name

In order to access the devices by name, this must be assigned one. From Linux systems, we have to change the file "*/etc/hostname*".

```
 > sudo nano /etc/hostname
```

In Windows 10 systems, open *Settings* and go to *System* > *About*

📖 **Important:** You can use letters, numbers and hyphens, but no spaces.

2.2 Relate names and IP locally

The easiest way to relate names with IP is using the *hosts* file. This file contains lines of text that are made of IP addresses followed by one or more host names. Each field is separated by white space (blanks or tabulation characters).

```
192.168.20.6 mortadelo-computer
192.168.20.7 filemon-computer
192.168.20.8 zipi-computer
192.168.20.9 zape-computer
192.168.20.10 carpanta-computer
```

This file is located in */etc/hosts* in Linux systems or in *[X]:\Windows\System32\Drivers\etc* (where [X] is the until where the Windows system is installed, usually *C*.

📖 **Important:** In Windows systems, the *hosts* file can not be directly modified for security reasons. To modify it is necessary to make a copy in another folder, modify it and then, replace the original with the modified using the administrator permissions

2.3 DNS

The hosts file can solve the problem of the location of names within a small and controlled environment such as a local network, but when we need to resolve names of internet servers, this solution is not feasible.

To do this, there are the so - called DNS servers, which provide a name - IP relationship. Obviously a single server can not resolve all the names of the entire internet, so if one server can not resolve, it will forward the request to another.

In general these servers are assigned by the ISP, but there are other public DNS servers like Google (8.8.8.8 and 8.8.4.4)

On Linux systems, these servers are indicated using the file */etc/resovl.conf*

```
nameserver 127.0.0.1
nameserver 172.16.1.254
```

Be careful with the modifications made to this file. In general, the DHCP server itself will assign not only the IP, but also the DNS servers. These modifications can override those performed manually. To keep this data, we can edit the "*/etc/network/interfaces"* with information that we want to add to the "*resolv.conf"* file.

```
iface en1 inet static
address 192.168.20.5
netmask 255.255.255.0
gateway 192.168.20.1
dns-nameservers 172.16.1.254,8.8.8.8
```

In Windows systems, you can configure DNS using its GUI following this tutorial:
https://www.youtube.com/watch?v=y1SEaJlYAG0

❕ **Attention:** In Windows example, we are using Google public DNS that are 8.8.8.8 and 8.8.4.4. They are easy to remember. You can find more information in the Wikipedia Google Public DNS.

3. SECURITY

3.1 Firewall

A firewall is a system (it can be implemented using software or hardware) that monitors and controls the incoming and outgoing network traffic. You have configured a series of trust and non-trust rules that apply to each package in a way that lets it pass depending on whether it accomplishes any of those rules.

3.1.1 Linux systems

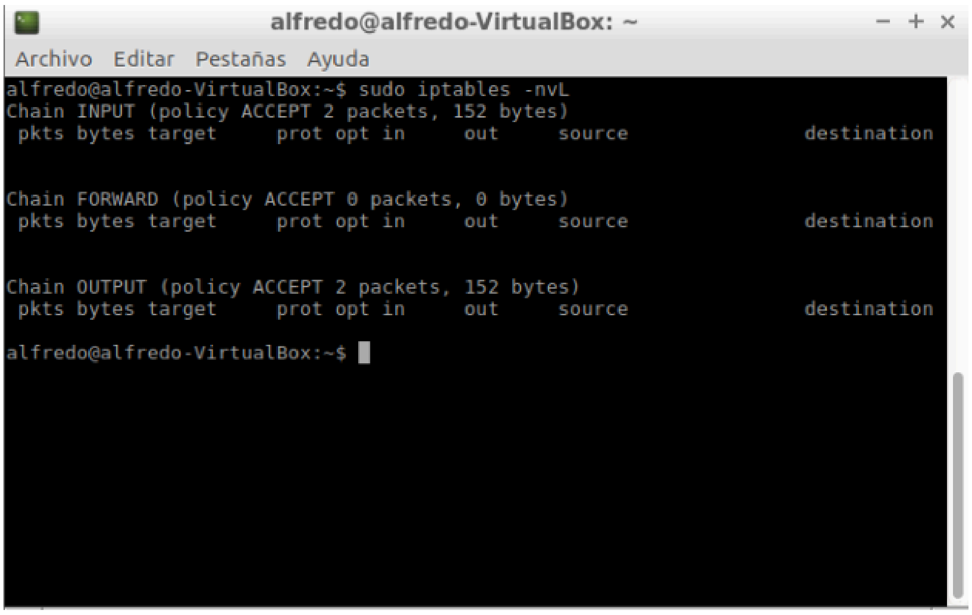
Linux includes a native firewall built inside the kernel. That firewall is controlled by the "*iptables"* command (you can install it via "*sudo apt install iptables -y*"). The IP tables are a set of tables that tell the kernel how to process incoming packets. Each table has a distinct function. For example, the filter table (the default table) provides commands to filter and accept or drop packets and in this way behave as a firewall. The NAT table provides commands to translate (modify) source or destination IP addresses, and the mangle table provides commands to modify packet headers.

Each table contains *chains,* which are sets of packet rules or policies. The table represents what to do with the packets, and the chain represents at which stage of the TCP/IP stack the operation must be done. For example, the filter table contains built-in chains called INPUT, FORWARD and OUTPUT, and support the policies ACCEPT, DROP and REJECT (and others). A packet DROP rule configured underneath the INPUT chain will direct the kernel to DROP packets that are received on a particular interface.

📖 **Important:** In our case we will only work with the filter table which is also the default *iptables* table. To choose the table to work with, you must use the *-t* option

To list the filter table:

```
iptables -nvL
```



```
alfredo@alfredo-VirtualBox:~$ sudo iptables -nvL
Chain INPUT (policy ACCEPT 2 packets, 152 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 2 packets, 152 bytes)
 pkts bytes target     prot opt in     out     source               destination

alfredo@alfredo-VirtualBox:~$
```

Figure 1 - Command "iptables -nvl"

📖 **Important:** By default, no rules are defined and the default policy for each chain is ACCEPT, so the *"iptables"* allow all packets to pass through the kernel

One more safe way to work is to change the default mode from ACCEPT to DROP, at least when it comes to incoming packets. To do it:

```
 sudo iptables -P INPUT DROP
```

This command changes the default INPUT policy (-P) to DROP. In this way, when you list the table again:

From this moment, our computer will reject all incoming packages.

This solution can be drastic since the communication with the outside can be impossible if we do not allow the entry of any package. The interesting thing is to add rules that are opening elements according to our needs. This opening can be done in several ways:

- **Open one interface**
- `iptables -A INPUT -i lo -j ACCEPT`
- append (A) a rule to the INPUT chain. The policy is ACCEPT and the rule is applied to the loopback (lo) interface (-i).

- **By protocol**
- `iptables -A INPUT -i eth1 -p udp --sport 68 --dport 67 -j ACCEPT`
- append (A) a rule to the INPUT chain. The policy is ACCEPT and the rule is applied to the ethernet1 (eth1) interface (-i) for the protocol (-p) udp. Besides, the client sends the packet for the port (-sport) 68 and the server is listening on the port (-dport) 68

- **By state**
- `iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT`
- iptables can control the connection state, that is, it can group the packets in connections according to their IP addresses and source / destination ports and understand if the connection is being started by the local machine or a remote machine. In the example, we append (A) a rule to the INPUT chain. The policy is ACCEPT and the rule is applied to the ethernet1 (eth1) interface (-i) if the state (-m) is ESTABLISHED (connections that we have originated) or RELATED (created by the existing ESTABLISHED connections)

3.1.2 Windows systems

Default firewall in Windows systems is very limited compared to iptables on Linux, but for simple configurations it could be useful.

You can configure it using the Windows GUI following this tutorial:

https://www.youtube.com/watch?v=M4QhgpXZB6E

4. Remote Access

At the administrative level, one of the first advantages of networks is the possibility of managing a computer remotely. There are many options, but we will focus on two of the most used.

📖 **Important:** An IP direction could have a lot of services. To distinguish which service we are connecting, each service has a different port. Most common ports are 80 for web servers, 22 for ssh, 25 for SMTP, etc.

More information in https://en.wikipedia.org/wiki/Port_(computer_networking)

4.1 SSH

The first existing option was *telnet*. *Telnet* is a program that allows to connect via terminal with another system. Simply indicate the IP and, with the proper credentials, are allowed to work remotely in the terminal of another computer.

However, *telnet* is no longer used because it has several security problems, the most important being that the connection is not encrypted so that data such as passwords could be obtained.

To solve this problem, it appears the *Secure Shell* (SSH) protocol family of tools for remotely controlling or transferring files between computers safely. Among the tools in the family are the remote connection (the style of telnet but secure) or the secure copy. The system consists of a daemon (*sshd*) in the computer to which we want to access and a client in the function of the type of access that is used (*ssh* for remote control, *scp* to copy ...).

In the server (the computer to which we want to access) we need:

1. Install the ssh daemon server

```
sudo apt-get install openssh-server
```

2. Open the corresponding port to receive packets. In this case, the protocol is TCP and the default port is 22

```
iptables -A INPUT -i eth1 -p tcp --dport 22 -j ACCEPT
```

In the client, we need to install the ssh client

```
sudo apt-get install openssh-client
```

To connect, from the client

```
> ssh user@192.145.6.23
```

You will get a warning message asking if you trust the digital signature of the remote computer. If you trust it, the signature will be stored in a hidden file called *.ssh/known_hosts*, and you won't get more warnings when you connect to this server in the future unless the fingerprint of the remote server changes, which can be a signal that someone is intercepting your connection. Then the program will ask you for your password[5] on the remote computer and a session will start.

4.2 Team Viewer

TeamViewer is one of the most common third-party tools for remote management. In this case, its functionality is not direct from one computer to another, but the connection is made through the TeamViewer's servers.

It does not require much configuration, simply install the program on both computers and then each computer will connect to the Team Viewer's servers on the Internet, and it will create a management account whose ID and password will be displayed on screen. You will have to type those credentials when you want to connect to each computer remotely.

You can watch a video about the process in the Moodle course.

5. SHARED RESOURCES

At the user level, one of the great advantages of networks is the possibility of sharing resources, especially files and printers. As always, there are several options to perform this sharing, with NFS, but in this unit we will work with SAMBA, a system that will allow us to share resources between Linux and Windows systems

5.1 SAMBA

Today, much of the functionality of SAMBA is transparent to the end user: a Linux user can open a file browser on the network, search Windows computers and access those elements that have been shared from Windows.

Even so, in many situations we will not be able to use the graphical interface or require a more detailed configuration.

In the platform are linked two videos (https://www.youtube.com/watch?v=zTujwRSsIBw and https://www.youtube.com/watch?v=p2r0kIB_ItE) about the installation and configuration of SAMBA.

6. BIBLIOGRAPHY

[1] Como funciona un servicio HDCP
http://windowserver.wordpress.com/2013/09/20/cmo-funciona-el-servicio-dhcp-incluye-capturas-de-red/

[2] Computer networks. S. Tanenbaum Andrew. Pearson. 2010

5 There are several ways to authenticate: login and password, private / public key ...