3TC-MAC
Networking Lab: Wireless Networks and Dynamic Host Configuration

Version _______: 1.2
Date_________: May 2023
Author________: Fabrice Valois

As usual, read carefully the subject before to start!

Goal of the lab (4 hours): During this lab, you will use several of your IP network skills. More, you will discover new networking tools, especially for dynamic host configuration but also for wireless network configuration. You will learn how to configure a Wi-Fi network, and you will observe some basic performance.

Part 0. Setup your system.

Step 0.0. Be careful when selecting the PC configuration.

Boot on the physical workstation `LIVE CD`.
The username/password is: `root/linadm`. You are now Super-User `Root`! 🦸

Step 0.1. Install the required packages.

First, you need to install the Ubuntu packages needed during this lab: the package `hostapd` for the wireless part of the lab, as well as the package `isc-dhcp-server` to administrate a DHCP server. Note that a DHCP client is already included in the Ubuntu distribution.

```
apt-get install <package_name>
```

In case you get an error message when installing a package, update the current installation:

```
apt-get update
```

Step 0.2. Remove several (dynamic) networking services

Remember your previous IP labs: you need to stop networking services (*e.g.*, default DHCP, DNS) in order to avoid conflict with our networking labs. Stop the networking manager:

```
service networking stop
nmcli networking off
```

Note that all the networking interfaces (`eth0`, `eth1`, `eth2`) are now off.

Several services are still running on your machines, including Dynamic DNS (DDNS), Multicast DNS (MDNS), NTP (Network Time Protocol) and IPv6 Neighbor Discovery. These services are managed by the `avahi` daemon. The goal of this daemon is to provide a kind of *plug and play* to access to your LAN services (*e.g.*, printers, file server). Stop this service to avoid conflict with your (future) DHCP server using the following commands:

```
service avahi-daemon stop
systemctl disable avahi-daemon.socket
```

Part 1. Configure your AP (Wi-Fi)

For this part, only use a Wi-Fi dongle (~~either a~~ Raspberry Pi WiFi Adapter ~~or a WirelessN Nano USB Adapter~~): it will be used to configure an access point (see Figure 1). Obviously, you can connect your own smartphone and tablet to your wireless network.

Figure 1. A workstation + an USB adapter = your AP.

Step 1.0. Discover the `ip` command.

The parameters that you will use most often are:

- `link (l)` – used to display and modify network interfaces.
- `address (addr/a)` – used to display and modify protocol addresses (IP, IPv6).
- `route (r)` – used to display and alter the routing table.
- `neigh (n)` – used to display and manipulate neighbor objects (ARP table).

You can get help and all the parameter options using:

```
ip link help
ip address help
...
```

Step 1.1. Installation and identification of the wireless interface id.

First, install the USB adapter on a PC. Identify the name of your wireless interface using:

```
iwconfig
```

This command is similar to `ifconfig` but for wireless interface with several options:

```
iwconfig interfaceName [essid X] [nwid N] [mode M] [freq F] [channel C]
[sens S ] [ap A ] [nick NN ] [rate R] [rts RT] [frag FT] [txpower T]
[enc E] [key K] [power P] [retry R] [modu M] [commit]
iwconfig --help
```

To get more information about this command:

```
man iwconfig
```

For an easiest administration, rename your wireless interface:

```
ip link set interfaceName name wlan0
```

Step 1.2. Up the interface!

Even if your wireless interface is available, it is not necessary running, the network is down. Try:

```
iwlist wlan0 scan
```

Run your wireless interface thanks to:

```
ip link set wlan0 up
```

Now, scan again your environment:

```
iwlist wlan0 scan
```

If it does not work (for some security or limitation reasons), it is probably because your wireless interface is locked. Solve this issue with:

```
rfkill unblock all
```

Step 1.3. Discover your (wireless) environment.

Identify several wireless networks (SSID), and look for the channel used, the encryption mode, the bit rates supported, the mode, the signal quality as well as the signal level. This command uses the scanning function of your Wi-Fi interface.

`iwlist` can also be used to display some additional information from a wireless network interface:

```
iwlist wlan0 scanning
iwlist wlan0 frequency
iwlist wlan0 rate
iwlist wlan0 keys
iwlist wlan0 power
iwlist wlan0 txpower
iwlist wlan0 retry
iwlist wlan0 event
iwlist wlan0 auth
iwlist wlan0 wpakeys
iwlist wlan0 modulation
iwlist --help
```

Only 11 channels are available, instead of 14. Explain.
Regarding to the French Regulation, what is your comment about the original `txpower` of the USB key? To respect the French Regulation, change the maximal transmission power for an indoor deployment:

```
iwconfig wlan0 txpower power(in dbm)
```

2 security keys are supported: 40 bits, 104 bits. Which security protocols are they related to?

To get more information about this command:

```
man iwlist
```

Step 1.4. From a USB adapter to an AP.
Finally, the command `iw` is used to show and manipulate wireless devices and their configuration. It provides more technical information than `iwlist`. Verify that your USB adapter is able to support the access point mode:

```
iw list
```

Look for the section Supported interface modes: `AP` means that your USB adapter can be an access point, whereas `managed` means that it can be a wireless client. Take time to verify the supported frequencies, the bitrates, the ciphers, etc.

Step 1.5. Configure the daemon.
The `hostapd` daemon is required to set up your Linux PC as an access point.
Create the file `/etc/hostapd/hostapd.conf` with the following information:

```
interface=wlan0               # WLAN interface for wi-fi
driver=nl80211                # Classical Linux driver for 802.11
ssid=YourCrazyName            # SSID
hw_mode=g                     # Wi-Fi mode (b:IEEE 802.11b g = IEEE 802.11g)
channel=6                     # Channel used
auth_algs=1                   # Authentification 1=wpa, 2=wep, 3=both
wpa=2                         # WPA2 support
wpa_key_mgmt=WPA-PSK          # Pairwise Shared Key: key management for
                              # authentication
rsn_pairwise=CCMP             # Ciphering and Integrity protocols for WPA2
wpa_passphrase=somepassword   # Private key, only in personal mode
                              # Professional use: certificate only!
max_num_sta=1                 # Maximum number of stations allowed
```

Now, run your AP:

```
hostapd /etc/hostapd/hostapd.conf
```

Step 1.6. Be connected!
Check if your smartphone is able to be associated to your wireless network (see Figure 2). Try to be associated and check the control information from the AP side when you use a wrong security key, then a good one. Check the IP configuration of your smartphone.
Try to connect a second device to your Wi-Fi.

Figure 2. A smartphone connected to your AP.

Step 1.7. Observe the (control) trafic.
Without any device connected to your AP, observe the (control) traffic using `wireshark` on the `wlan0` interface. Then, connect a device to your AP. Observe the (control) traffic.
Why do you see `Ethernet` frames?

To observe 802.11 frames, the USB Adapter should run in a monitor mode. In this case, all the 802.11 headers could be observed, as well as the control frames (*e.g.*, beacon, association request). Unfortunately, this mode is not ~~well~~ supported on our platform… 😭

Step 1.8. Limit the access to your Wi-Fi.

To limit the use of your wireless network, a simple access control can be enabled. In the file:

```
/etc/hostapd/hostapd.conf
```

add:

```
macaddr_acl=1                 # access list via address-based authentication
                              # 0: accept unless in deny list
                              # 1: deny unless in accept list
                              # 2: use external RADIUSserver
accept_mac_file=/etc/hostapd/hostapd.accept
```

Here is an example for `/etc/hostapd/hostapd.accept`:

```
00:0a:f7:2d:6a:b4
00:0a:f7:2d:79:dd
…
```

In your testbed, consider the MAC address of your smartphone. Stop, then start the `hostapd` daemon. Try to be associated.

By the way, your are able to connect any devices to your AP using the three steps of 802.11: scanning, authentication, association. Congratulations! Unfortunately, the IP configuration is missing!

Part 2. Create a (static) LAN with the `ip` command

During your first journey to discover IP, you used `ifconfig` to configure the networking interface. If this command is widely used, it is unfortunately deprecated since several years. Instead, the command `ip` should be used. Let's go to configure your two LANs using this command (see Figure 3)!

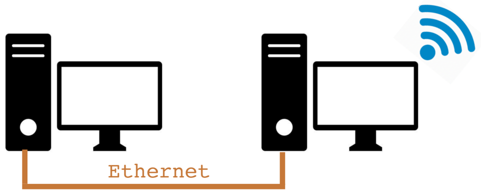

Figure 3. A small wired network.

Step 2.1. Configure your Ethernet interfaces.

List all the networking interfaces of your computer using:

```
ip link show
```

Verify the ones which are running:

```
ip link ls up
```

To bring the network interface `interfaceName` up, use :

```
ip link set interfaceName up
```

Configure an IP address to this interface

```
ip addr add ip_address/netmask dev interfaceName
```

Verify your configuration:

```
ip addr show
```

Use `ping` to confirm the network connectivity of your LAN.

Step 2.2. From wireless to wired, and vice-versa.

Before to discover the dynamic host configuration of your wireless device in practice, configure manually the IP address of your smartphone. Provide also an IP address to `wlan0`. Enable the routing for the PC hosting the AP (if you do not remember how do it, check your first 3TC-IP lab). Then, to add route using `ip`, use:

```
      ip route add ip_address/nemtask dev interfaceID
or    ip route add ip_address/netmask via GatewayIP
or    ip route add default dev interfaceId
or    ip route add default via GatewayIP
```

To check your routing table:

```
      ip route list
```

Verify the connectivity between the wireless network and the wired one.

Part 3. Some basic performance observations.

Step 3.0. Stability and delay comparisons.

Considering the Figure 4, and using the basic `ping` tool:

- compare the latency of your wireless network with the Ethernet network. Explain the difference ;
- compare also the *stability* of these two networks. Explain.

Consider several scenarios, such as:

- your smartphone close to the AP,
- increase the distance between your smartphone and the AP,
- associate several smartphones to your AP,
- use the same channel for several (all ?) APs,
- etc.

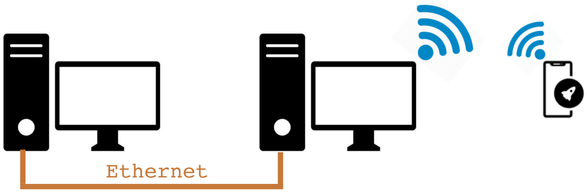

Figure 4. The final topology, and the performance evaluation scenario.

Now, it’s time to:
1. configure a dynamic IP address allocation for the wireless devices ;
2. enable a routing from your wireless network to the wired one, and vice-versa.

Part 3. Dynamic Host Configuration Protocol

Step 4.0. A big picture of DHCP.

DHCP is an automatic way to configure the networking protocol stack of a workstation, meaning that DHCP provides IP address, netmask, routing table or default gateway information as well as DNS name server. DHCP is used in open wireless networks as well as in `eduroam`, or in Ethernet networks. ADSL boxes often uses DHCP to provide IP configuration to smartphones, tablets, etc. As in client/server architecture, DHCP operates into two modes:

1. Either periodically, the DHCP server broadcasts control information to the whole subnetwork in order to announce its presence. Then, a client requests configuration in unicast to this server.
2. Or, when a new workstation is connected to the network, without networking information, it broadcasts a request to look for a DHCP server. The DHCP server provides configuration in unicast to this workstation.

When a configuration is provided from the DHCP server, it is only for a limited duration: the lease time.

Step 4.1. Be ready to monitor!
Run `wireshark` on your AP (interface `wlan0`) to capture the packet exchanges (or monitor the port 24 of your switch to capture both the client and server exchanges), and keep this window active. Use the UDP filter to monitor DCHP packets (more precisely, ports 67 and 68).

Step 4.2. Configure the DHCP server.
On the wireless network interface of your workstation, configure the DHCP server by modifying two configuration files, as in the following examples. First, save a copy of the current `/etc/dhcp/dhcpd.conf` file. Second, use this minimal configuration file:

```
File 1 /etc/dhcp/dhcpd.conf :
subnet 192.168.y.0 netmask 255.255.255.0 {
   option routers 192.168.y.1;             # Default gateway
   option subnet-mask 255.255.255.0;       # subnet mask
   range 192.168.y.2 192.168.y.100;         # IP adresses range
   default-lease-time 600;                 # lease time (seconds)
   max-lease-time 7200;                    # maximum lease time (id)
}
```

Do not hesitate to change and test different lease time values to verify its impacts.
Now, parse the initial version of the configuration file to learn more about the different options available:

```
/etc/dhcp/dhcpd.conf.copy
```

Check the file `/etc/sysconfig/dhcpd`. What do you learn from the comments? Do you need to configure first the networking interface of the DHCP server or this interface is also dynamically configured?

Now, specify on which networking interface the DHCP server should be available. Add the following information in the file `/etc/default/isc-dhcp-server` :

```
INTERFACES="wlan0"
```

Step 4.3. Start the DHCP service on your PC with the AP (router):

```
service isc-dhcp-server start
```

Step 4.4. In case of a client which is not your smartphone, to request a configuration on the interface `interfaceName` use the following command:

```
dhclient interfaceName
```

You can cancel a lease using: `dhclient -r interfaceName`
To renew an IP address: you first cancel the current one, then you request a new one:

```
dhclient -r interfaceName
dhclient interfaceName
```

Observe before/after the networking configuration of the clients.
Use `ping` to validate the configuration on the subnet2.

Step 4.5. Observe the DHCP packet exchanges carefully: how many control packets are exchanged? Illustrate using a time sequence diagram. Check the IP address source & destination, check the information into the DHCP packets. Motivate the use of the IP address `0.0.0.0`.

Step 4.6: At the DHCP server side, verify the DHCP clients which are configured thanks to this file:

```
/var/lib/dhcpd/dhcpd.leases
```

Step 4.7. configuration with dedicated (and static) @IP ↔ @MAC associations
Modify `/etc/dhcp/dhcpd.conf` in order to statically associate a given @IP to a given @MAC:

```
host client1 {
      hardware ethernet 00:80:23:a8:a7:24;
      fixed-address 192.x.y.10;
}
```

To avoid to allocate an IP address to unknown clients, use this option in `/etc/dhcp/dhcpd.conf`:

```
deny unknown-clients
```