DHCP - Dynamic Host Configuration Protocol

Michel FACERIAS

13 décembre 2023

Polytech Montpellier

Université de Montpellier

Table des matières

1 DHCP, or the story of a lazy NetOps, but not only that !

1.1 DHCP, a brief summary

According to Wikipedia, the **Dynamic Host Configuration Protocol** (DHCP) is a **network management protocol** used on IP networks for **automatically assigning IP parameters** to devices connected to that network, eliminating the need for individually configuring devices manually.

DHCP can be implemented on networks ranging in **size from residential networks to large campus networks**. Most of routers and residential gateways have DHCP server capability.

DHCP is **not the only way to have autoconfiguration**, but in fact, it has become so synonymous with that some believe it is also at the heart of regional ISP networks. Sometimes it is, but rarely.

DHCP services exist for networks running IPv4. Initially, IPv6 had so many autoconfiguration processes that it was not intended to use DHCP. Today, there is an IPv6 version of the DHCP protocol is commonly called DHCPv6.

1.2 DHCP in detail (or nearly so)

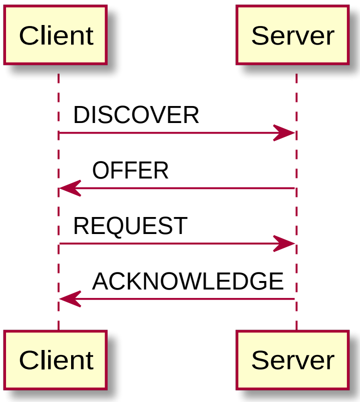
1.2.1 Concept : Network model

DHCP consists of **two network components**, a centrally installed network **DHCP server** and **DHCP client** instances of the protocol stack on each client device. When connected to the network, and periodically thereafter, a **client requests a set of parameters** from the server using DHCP.

The DHCP employs a connectionless service model, using UDP. It is implemented with two UDP port numbers for its operations :

- UDP port 67 is the port used by the server ;
- UDP port 68 is used by the client.

DHCP operations fall into four phases, often abbreviated as DORA for **discovery**, **offer**, **request**, and **acknowledgement**. Discovery phase uses **broadcast** to find a candidate server.



1.2.2 Concept : Dynamic allocation

A NetOps reserves a range of IP addresses for DHCP, and each DHCP client is configured to request an IP address from the DHCP server during network initialization.

The **request-and-grant** process uses a lease concept with a **controllable time period**, allowing the DHCP server to **reclaim and then reallocate** IP addresses that are not renewed.

1.2.3 Concept : Automatic allocation

The DHCP server **permanently assigns an IP address to a requesting client** from a range defined by the NetOps.

This is like dynamic allocation, but the **DHCP server keeps a table of past IP address assignments**, so that it can preferentially assign to a client the same IP address that the client previously had.

Most of the time, the server acts basically as **Automatic and Dynamic**.

1.2.4 Concept : Manual allocation

This method is also variously called **static DHCP allocation**, or **reservation**.

The NetOps **maps a unique identifier** (a client id or MAC address) for each client to an IP address, which is offered to the requesting client.

This method has many advantages for managing roaming clients or who have several network interfaces (wired and wireless) while always giving the same IP address.

1.3 Bibliographical resources

You will find a lot of information about DHCP. But I advise you to voluntarily restrict yourself to the following :

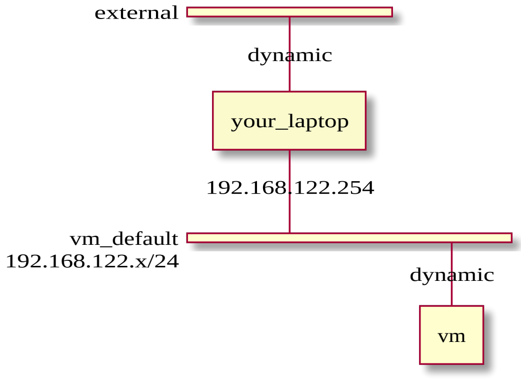
See more : https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
As always, a very good, precise and concise work of synthesis.

See more : https://www.isc.org/dhcp/ The main server's reference website.

And, of course, ***man*** is your best friend !

2 Building your own sandbox

2.1 The VM virtual network



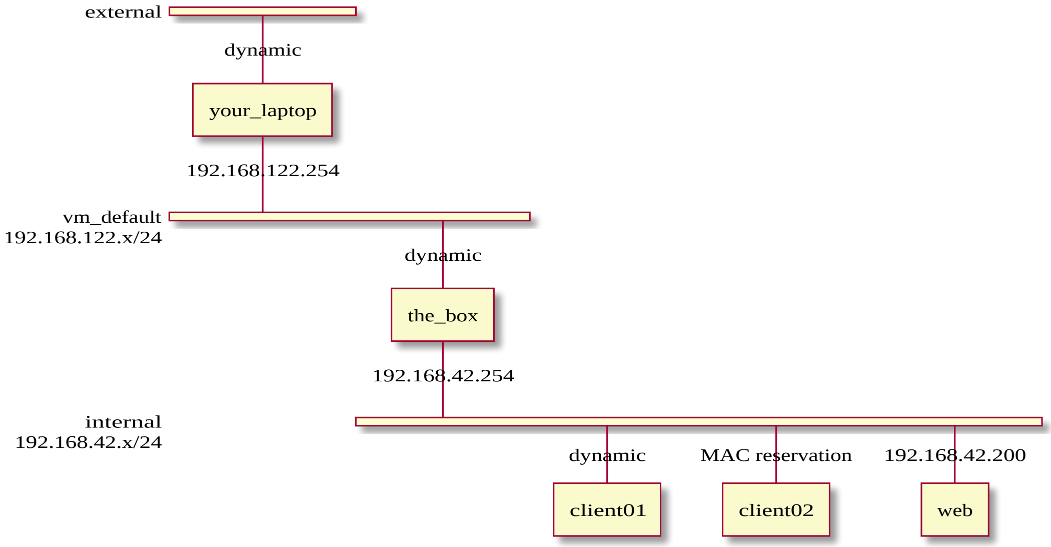
The virtualisation tool you normally have at your disposal reproduces this network diagram :
- your laptop acts as the **hypervisor** ;
- the hypervisor implements **VMs hosting** ;
- the hypervisor provide **virtual network resources** :

- The ***default*** **network** ;
- The **routing process** from this network to your external network (i/e routing and NAT) ;
- A **DHCP service** for VMs ;
- A **DNS service** for the VMs.

As the two last services are implemented by a small (but powerfull) integrated tool (*dnsmasq*), you must build an **isolated network**. On this network, you will **provide your own DHCP service**.

2.2 Building a full sandbox

The network model you need is described in the following figure.



2.2.1 To do : Create the *internal* network

Using *virtmanager*, right click on the local VMs domain, chose *detail*, then *networks*, and create an isolated network named ***internal***.

Deactivate IPv4 configuration.

2.2.2 To do : Create *the_ box*

Clone your VM template to have a VM named *the_ box*.

As this VM is going to simulate a residential ISP gateway, we need :

- one network cards connected on *default* network, using dynamic configuration from this network ;
- an other network card connected on *internal* network, statically addressed on 192.168.42.254/24 ;

Assuming *eth0* is the interface linked to the network *default* and *eth1* is those linked to the network *internal*, */etc/network/interfaces* must look like :

```
...
# The loopback network interface
auto lo
iface lo inet loopback

# uplink to "default" VMs network
auto eth0
iface eth0 inet dhcp
```

```
#       downlink to "internal" network
auto eth1
iface eth1 inet static
  address 192.168.42.254/24
```

At this step, *the_box* should *ping* all the internet !

2.2.3 To do : Create *client01*

Clone your VM template to have a VM named *client01*. This VM **must only be connected** to the *internal* network.

This VM is going to simulate a basic client. At the end, it should be autoconfigured by DHCP installed on *the_box*, but not at this time.

Assuming *eth0* is the *internal* network interface, you can force an IP address using this command :

```
# ip address add 192.168.42.100/24 dev eth0

# ip route add default via 192.168.42.254
```

At this step, ***client01* should *ping* 192.168.42.254 and nothing else**, even adding a default route !

2.2.4 To do : Make *the_box* a gateway

Remember that we need :

- a routing capability ;
- a NAT process.

To activate the routing capability, edit */etc/sysctl.conf* and allow IPv4 forwarding as this :

```
        ...
        # Uncomment the next line to enable packet forwarding for IPv4
        net.ipv4.ip_forward=1
        ...
```

Then, assuming *eth0* is the interface linked to the network *default* (the output to the internet), invoke these commands :

```
        # apt install iptables
        ...
        # iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

But be careful, **this action will stop at the next reboot**. So you need to make "persistent" this rule by installing this package and configure it to **save IPv4 rules only**.

```
        # apt install iptables-persistent
```

At this step, ***client01* should ping all the internet** !

3 DHCP server installation and configuration

3.1 Basic installation

3.1.1 To do : Deploying the package

On *the_box*, perform the followed commands :

```
        # apt update && apt upgrade
        # apt install isc-dhcp-server
```

Your installation ends with an error, and that's normal

3.1.2 To do : Debugging the configuration

All server should be launched on an interactive command line.

Read *man dhcpd* and find the way to run *dhcpd* (the DHCP server) as a foreground process instead of as a daemon in the background. Run it, and find the error.

Change to propose to your clients :

— 192.168.42.1 to 192.168.42.99 addresses ;

— 192.168.42.254 as default gateway ;

— *dopolytech.fr* as default domain ;

— the DNS server used by *the_ box* as DNS server.

Edit `/etc/dhcp/dhcpd.conf`, but before, make a copy as *dhcpd.conf.orig*. To find the DNS server used, have a look at `/etc/resolv.conf`.

Reload *dhcpd* on foreground to validate your config file.

Then test the result on *client01*, assuming *eth0* is the name of its network interface :

```
        # dhclient eth0
```

3.2 DHCP reservation

3.2.1 To do : Add an new *client02*

Clone a new VM named *client02*.

Make a MAC/IP mapping in `/etc/dhcp/dhcpd.conf` to offer always the same IP address to *client02*.

Try first to chose an IP out of the range. Then try an IP in the range.

Conclude on the right way to make DHCP reservation.

3.2.2 To do : Propose a template DHCP configuration

Remember rules and good practices in IP addressing.

Assuming you have to plane 192.168.42.0/24 network, propose address range for :

— unknown roaming client ;

— known roaming client with reservation base on MAC/IP mapping ;

— statically addressed servers, including the default gateway ;

Create a statically addressed *web* VM and install an http server on it.

3.3 Conclusion

3.3.1 To do : Clean the configuration file

Your configuration file was build using the template given in the package. There are à lot off unused lines !

Clean the file, and only conserve active lines. Add your own comments before each line after learning what they do, and make some tests for it.

Then, add your template DHCP configuration as comments at the begining of this file, to improve your configuration file.

You should push this file by mail, named *dhcpd-lastname_firstname.conf*. This is to assess your work !

4 DHCP client configuration

Using the **DHCP protocol as a client is extremely simple**. All necessary information for the network connection is passed on in the DHCP *offer* datagram :

- IP address and network mask ;
- Default route (i/e gateway) ;
- Other routes, if needed ;
- DNS servers ;
- DNS default domain.

4.1 Linux basic client howto

Assuming :

- *client02* is using DHCP (Normally, it should !) ;
- your network provider (ISP) is *Eduroam*.

See more : https://en.wikipedia.org/wiki/Eduroam ***Eduroam*** (education roaming) is an international Wi-Fi internet access roaming service for users in research, higher education and further education.

4.1.1 To do : Analysis of `/etc/resolv.conf`

This job will be made on the VM the_box because it uses DHCP protocol, and it is the closest one to your ISP's network.

Show the content of `/etc/resolv.conf`, and try to understand it. Yours should looks like this one :

```
# cat /etc/resolv.conf
domain umontpellier.fr
nameserver 193.51.152.152
nameserver 193.51.152.153
```

4.1.2 Question : Analysis of *dhclient* behaviour

What gives this information to *the_box* ?

You should use `man resolv.conf` and `man dhclient` to help you.

4.1.3 To do : Try to resolve some domain names

Then, use the `host` command to resolve IP addresses :

```
# host www.umontpellier.fr
www.umontpellier.fr has address 193.51.152.74

# host www
www.umontpellier.fr has address 193.51.152.74
```

4.1.4 Question : Analysis of FQDN complétion

What happened when the name you gave is not a FQDN ?

4.2 How to hack *dhclient* behaviour ?

If you are a **roaming user**, it can be **useful to maintain the same behaviour** regardless of where you use your computer.

For example, **always using a UQDN** will make it **easier to access** your company's resources.

⚠ CAUTION, what is seen here is confirmed by the `host` command. But your **browser has internal mechanisms** to replace a UQDN with the result of a search engine.

To obtain the same result with your browser, **you will have to modify its configuration**.

4.2.1 To do : Make my own domain name a priority

On *client02*, we want that all UQDN will bee complete by your own domain name.

Edit `/etc/dhcp/dhclient.conf`, and add a supersede option at the end :

```
$ cat /etc/dhcp/dhclient.conf
...
supersede domain-name "facerias.org";
```

Assuming `eth0` is the network interface used by *client02*, then force *dhclient* to reload :

```
# dhcient eth0
```

4.2.2 Question : Why /etc/resolv.conf is different ?

Have a look at /etc/resolv.conf and try to explane.

4.2.3 To do : Try to resolve the same domain names again

Retry to resolve :

— a FQDN : *www.umontpellier.fr*

— a UQDN : *www*

and www

4.2.4 Question : How it works together ?

Try to establish the chain of actors between the DHCP server and the resolution with the *host* command.