



Mon @IP est personne et je demande une @IP à tout le monde

Auteur : Pascal Fougeray



source : Je ne sais plus ☺

1 Introduction

Le protocole DHCP répond à 3 problématiques :

1. Seuls les stations en service utilisent une adresse de l'espace d'adressage qu'il propose !
2. Toute modification des paramètres réseau telles que :

- (a) Adresse de passerelle : **Gateway**
- (b) Adresses des serveurs de noms **DNS**

est **rapportée au niveau des stations** lors de leur redémarrage sans que l'administrateur ait besoin de s'en occuper !

3. La modification des paramètres est centralisée au niveau du ou des serveurs DHCP.

Le protocole DHCP a été utilisé **la première fois en 1993**. IL est défini par la RFC1531 et a été, par la suite modifié et complété par les RFC1534, RFC2131 et RFC2132. Il fonctionne en IPv4 et en IPv6. Dans ce cas, il s'appelle DHCPv6 et les adresses peuvent être auto configurées, sans DHCP. (Voir le cours IPv6 si on a le temps de le faire...)

Il fonctionne en mode client/serveur et s'appuie sur le mécanisme de **requêtes DHCP**, traitées par le serveur et émises par les clients. Le moteur principal de ce protocole est adossé à la communication **BOOTP** en utilisant des trames **UDP**.

2 Bootstrap Protocol

Juste une petite explication pour information...

BOOTP pour **Bootstrap Protocol** est un protocole réseau d'amorçage, qui permet à une machine cliente **sans disque dur** de découvrir

- sa propre adresse IP,
- l'adresse d'un hôte serveur,
- et le nom d'un fichier à charger en mémoire pour exécution.

L'amorçage d'une station est une opération se produisant en 2 phases :

1. Détermination d'adresses et sélection du fichier de démarrage, c'est ici qu'intervient BOOTP.



2. Transfert du fichier de démarrage, le transfert utilisera typiquement les protocoles TFTP, SFTP ou encore FTP.

Le serveur BOOTP utilise le port 67 et le client BOOTP utilise le port 68. DHCP aussi.

BOOTP est sur la couche 3, couche réseau du modèle OSI alors que DHCP est sur la couche 7, couche application !

3 Définitions

DHCP comme **Dynamic Host Configuration Protocol** ou protocole de configuration dynamique des hôtes.

Ce protocole a pour rôle d'assurer la **configuration automatique des paramètres IP** d'un hôte en lui attribuant **automatiquement** une adresse IP et un masque de sous-réseau et éventuellement l'adresse du serveur DNS (voir chapitre suivant)

DHCP s'appuie sur UDP et 2 ports.

1. Le port **68** pour la source donc le client (**Bootpc**)
2. Le port **67** pour la destination donc le serveur (**Bootps**)

La conception initiale d'IP supposait la pré-configuration de chaque hôte connecté au réseau avec les paramètres TCP/IP adéquats : c'est l'**adressage statique** ou **IP fixe**.

Sur des réseaux de grandes dimensions ou étendues, où des modifications interviennent souvent, **l'adressage statique engendre une lourde charge de maintenance et des risques d'erreurs.**

Donc on a mis en place un système dynamique : **DHCP**

4 Le Mystère

Il y a un grand mystère avec DHCP !

Question : Comment une machine qui n'a pas d'IP peut lancer un protocole de couche 7 ?

Réponse : Elle s'octroie une IP, l'IP **0.0.0.0** qui représente **personne** ou **n'importe qui** sachant que tout le monde peut être n'importe qui mais que tout le monde n'est pas n'importe qui (J'adooooore la langue française ...) et va s'adresser à **tout le monde** donc **255.255.255.255**

Ceci est à connaître par cœur ou avec amour (non qu'en même pas ☺) pour le CT ☺

5 Le protocole

Il s'appuie sur UDP

Les différents types de paquets

DHCPDISCOVER : Permet de localiser les serveurs DHCP disponibles

DHCP OFFER : Réponse du serveur DHCP à un paquet DHCP DISCOVER.

DHCP REQUEST : Diverses requêtes du client.

DHCP ACK : Réponse du serveur contenant les paramètres réseau.

DHCP NAK : Réponse du serveur signalant au client que le bail est échu.

DHCP DECLINE : Annonce du client que l'adresse fournie est déjà utilisée.

DHCP RELEASE : Libération de l'adresse IP de la part du client.

DHCP INFORM : Demande de paramètres locaux de la part du client ayant déjà son IP.

Je ne vais pas tous les détailler, seuls 5 sont à connaître, **DORA** et **Release**

Imaginons le scénario suivant :

La VM est connectée à l'interface VBOXNET du HOST. Ce dernier fait aussi office de serveur DHCP sur le réseau 192.168.56.0/24. Réseau que l'on peut changer si on veut...

Si dans la VM on lance, **attention il faut être root**, les commandes **ifdown enp0s9** puis **ifup enp0s9** on obtient les messages suivants :



```
root@debian-12-GNS3~# ifdown enp0s9
Killed old client process
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
```

```
Listening on LPF/enp0s8/08:00:27:a9:cf:5b
Sending on LPF/enp0s8/08:00:27:a9:cf:5b
Sending on Socket/fallback
DHCPRELEASE of 192.168.56.108 on enp0s9 to 192.168.56.100 port 67
```

```
root@debian-12-GNS3~# ifup enp0s8
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
```

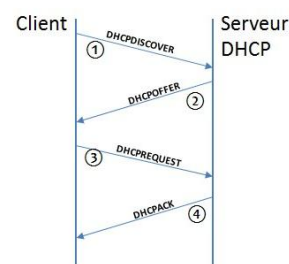
```
Listening on LPF/enp0s9/08:00:27:a9:cf:5b
Sending on LPF/enp0s9/08:00:27:a9:cf:5b
Sending on Socket/fallback
DHCPDISCOVER on enp0s9 to 255.255.255.255 port 67 interval 5
DHCPOFFER of 192.168.56.108 from 192.168.56.100
DHCPREQUEST for 192.168.56.108 on enp0s8 to 255.255.255.255 port 67
DHCPACK of 192.168.56.108 from 192.168.56.100
bound to 192.168.56.108 — renewal in 230 seconds.
```

On peut remarquer les 5 messages à connaître !

Si on fait une capture Wireshark on obtient la cela

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.108	192.168.56.100	DHCP	342	DHCP Release - Transaction ID 0x97dda235
4	16.036999999	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0x80e06e03
5	16.037725710	192.168.56.100	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x80e06e03
6	16.038030600	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0x80e06e03
7	16.042031756	192.168.56.100	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0x80e06e03

- Le premier paquet émis par un client est de type **DHCPDISCOVER**. Ce qui donne ...
 - Le serveur identifié répond alors par un paquet **DHCPOFFER** afin de soumettre une adresse IP à ce même client.
 - Ce dernier établit alors sa configuration et émet un paquet **DHCPREQUEST** afin de valider sa nouvelle adresse IP.
 - Le tout se terminant par un **DHCPACK**
- On parle de processus DORA (**D**iscover, **O**ffer, **R**equest, **A**ck)



1. Le client DHCP diffuse un paquet **DHCPDISCOVER**. Il s'agit d'un message diffusé aux stations du sous-réseau. Le seul équipement à pouvoir répondre est le serveur DHCP, ou un agent relais DHCP, dans le cas où le brin réseau n'est pas le même. Dans ce dernier cas, c'est l'agent qui transfère le message au serveur DHCP auquel il est rattaché.
2. Tout serveur DHCP présent sur le sous-réseau doit être en mesure de répondre, en émettant un paquet DHCPOFFER. Ce dernier fournit ainsi au client une adresse potentielle.
3. Le client, après avoir reçu le paquet **DHCPOFFER**, peut recevoir d'autres offres provenant d'autres serveurs DHCP. Dans ce cas, le client doit choisir l'adresse fournie par le serveur lui ayant répondu en premier et répond par un paquet DHCPREQUEST, contenant un identificateur du serveur. Cela permet aux serveurs DHCP recevant la diffusion de savoir quel est le serveur dont le message DHCPOFFER a été accepté par le client.
4. Les autres serveurs DHCP reçoivent ainsi le paquet **DHCPREQUEST**, émis par un client. Les serveurs qui n'ont pas été élus par le message DHCPREQUEST, se servent de ce dernier comme notification pour signifier le refus du client. Le serveur **sélectionné** stocke l'adresse IP du client dans



sa base de données et répond par un message DHCPACK. Lorsque le serveur DHCP ne peut fournir l'adresse promise, contenue dans le message DHCP OFFER initial, il envoie alors un message DHCPNAK.

Ci-dessous les 4 paquets plus en détails. Captures faites chez moi entre mon PC et ma Box ☺
Vous pouvez le faire chez vous, il suffit d'installer Wireshark.

— 1 Discover et 2 Offer

Cela fait DORA ;)

— 3 Request et 4 ACK

Et si on ne veut plus de l'IP, facile

— DHCP release

La commande **ifdown eth1** permet de faire tomber l'interface Eth1 et donc de libérer l'IP.

```
root@debian-10-etu:/etc/network# ifdown eth1
Killed old client process
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth1/08:00:27:a3:30:5c
Sending on LPF/eth1/08:00:27:a3:30:5c
Sending on Socket/fallback
DHCPRELEASE of 192.168.56.102 on eth1 to 192.168.56.1 port 67
root@debian-10-etu:/etc/network# dhclient eth1
```

6 Sous Linux

Sous Linux, si vous êtes sur une distribution de type Debian et si votre fichier **/etc/network/interfaces** contient des lignes telles que :

auto eth0 ← Ici eth0 représente le nom de l'interface et peut être autre !



iface eth0 inet dhcp

Alors votre interface demandera un IP à un serveur DHCP au démarrage !

Vous pouvez libérer cette IP à l'aide de la commande **dhclient -r** (r comme **release**)

Vous pouvez obtenir un IP, si auto eth0 n'est pas positionné dans le fichier **/etc/network/interfaces**, à l'aide de la commande **dhclient eth0**.

...

Quand vous **lancez** cette commande, il faut être root !

Il est possible de modifier le **timeout**, c'est à dire le temps max que va attendre le client. Par défaut il est positionné à 300s soit 5mn.

C'est à modifier dans le fichier **/etc/dhcp/dhclient.conf**, voir la ligne **timeout=xxx** (xxx en secondes)

Si le client ne trouve pas de serveur, il abandonne et s'alloue une adresse **169.254.0.0/16**, adresses locales auto configurées (**APIPA**) (Voir cours IP ☺)

Pour les distributions de type Redhat telles Centos, Fedora ...

Voir le site Web :

https://access.redhat.com/documentation/fr-fr/red_hat_enterprise_linux/7/html/networking_guide/index

Ou le PDF : Qui fait plus de 200 pages...

https://access.redhat.com/documentation/fr-fr/red_hat_enterprise_linux/7/pdf/networking_guide/red_hat_enterprise_linux_7-networking_guide-fr-fr.pdf

7 Installer un serveur DHCP

Ce n'est pas très compliqué si on se contente du minimum, cela peut devenir plus compliqué si on veut le maximum ☺

7.1 Sous Linux

Il existe plusieurs serveurs DHCP possibles sous Linux.

Il n'est pas "compliqué" d'installer un serveur DHCP sous linux

En TP nous allons le faire. On a deux possibilités entre autres

1. **dnsmasq** qui fait serveur DNS et serveur DHCP, il est léger et fonctionne très bien pour nous ?

<https://thekelleys.org.uk/dnsmasq/doc.html>

Une bonne doc bien écrite : https://doc.ubuntu-fr.org/configuration_serveur_dns_dhcp

ou

2. **isc-dhcp-server** qui lui ne fait office que de serveur DHCP : <https://www.isc.org/dhcp/>

Voyons rapidement l'utilisation et la configuration de **isc-dhcp-server**

Je vous propose de le faire avec **isc-dhcp-server**, dans la VM !

1. Voyons si le port est ouvert à l'aide de la commande **netstat -lnpu | grep 67**

2. Pourquoi 67 ?

3. Pas de port ouvert, installons : **apt install isc-dhcp-server**

```
Generating /etc/default/isc-dhcp-server...
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/isc-dhcp-server.service
```

```
/lib/systemd/system/isc-dhcp-server.service.
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/isc-dhcp-server6.service
```

```
/lib/systemd/system/isc-dhcp-server6.service.
```

```
Traitement des actions différées (« triggers ») pour man-db (2.10.2-2) ...
```

On voit qu'on a un service qui est créé : **/lib/systemd/system/isc-dhcp-server.service**.

Il y a aussi IPv6 Mais on ne va pas s'en servir !

4. Voyons si le démon tourne : **systemctl list-units --type=service | grep dhcp** renvoie :

```
root@PAF:~# systemctl list-units --type=service | grep dhcp
```

```
isc-dhcp-server.service      loaded failed failed    ISC DHCP IPv4 server
```

```
isc-dhcp-server6.service    loaded failed failed    ISC DHCP IPv6 server
```



Normal nous n'avons pas configuré le serveur !

5. Une fois installé, il faut définir les interfaces d'écoutes dans le fichier **/etc/default/isc-dhcp-server**.

(a) **nano /etc/default/isc-dhcp-server**

(b) **INTERFACESv4="tap0"**

(c) Cela nécessite un redémarrage donc :

i. **systemctl stop isc-dhcp-server.service**

ii. **systemctl start isc-dhcp-server.service**

6. Voyons si cela fonctionne : **systemctl status isc-dhcp-server.service**

7. Puis Il faut saisir la configuration dans **/etc/dhcp/dhcpd.conf**

Le fonctionnement de la configuration est simple.

(a) SAV : **cp dhcpd.conf dhcpd.conf-sav**

(b) **nano -l dhcpd.conf**

(c) Dé commenter les lignes 33 et 34 et mettre 172.31.1.0 Rx de TAP0

```
subnet 172.31.1.0 netmask 255.255.255.0 {
    range dynamic-bootp 172.31.1.3 172.31.1.253;
}
```

8. Relançons correctement le serveur

root@PC :/etc/dhcp# **systemctl stop isc-dhcp-server.service**

root@PC :/etc/dhcp# **systemctl start isc-dhcp-server.service**

root@PC :/etc/dhcp# **systemctl status isc-dhcp-server.service**

Renvoie :

```
root@PAF:/etc/dhcp# systemctl status isc-dhcp-server.service
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; preset: enabled)
   Active: active (running) since Thu 2022-12-15 09:43:23 CET; 5s ago
     Docs: man:dhcpd(8)
    Main PID: 11105 (dhcpd)
      Tasks: 1 (limit: 38288)
    Memory: 1.5M
       CPU: 32ms
    CGroup: /system.slice/isc-dhcp-server.service
            └─11105 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf tap0
```

9. Voyons si le daemon tourne : Son PID vient de nous être donné, voyons voyons : **ps aux | grep dhcpd** renvoie :

root@PAF:/etc/dhcp# **ps aux | grep dhcpd**

```
dhcpd      11105  0.0  0.0  6392  4632 ?        Ss   09:43   0:00 dhcpd -user dhcpd -group dhcpd
root       11119  0.0  0.0  11620  2272 pts/3    R+   09:47   0:00 grep --color=auto dhcpd
```

root@PAF:/etc/dhcp# **netstat -lunp | grep 11105**

```
udp        0      0 0.0.0.0:67          0.0.0.0:*           11105/dhcpd
```

root@PAF:/etc/dhcp#

10. Allumons le PC et faisons une capture Wireshark (à placer avant d'allumer le PC !)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xcff20e14
2	0.650521	172.31.1.254	172.31.1.3	DHCP	342	DHCP Offer - Transaction ID 0xcff20e14
3	0.999846	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xcff20e14
4	1.006357	172.31.1.254	172.31.1.3	DHCP	342	DHCP ACK - Transaction ID 0xcff20e14

On a un beau Dora, le serveur IP est bien 172.31.1.254 il offre et le client qui avait 0.0.0.0 obtient 172.31.1.3

Tout est OK;)

11. Un **ifconfig** sur le PC renvoie

gns3@box:~\$ **ifconfig eth0**

```
eth0      Link encap:Ethernet  HWaddr 0C:9E:51:54:00:00
```

```
          inet addr:172.31.1.3  Bcast:172.31.1.255  Mask:255.255.255.0
```

gns3@box:~\$ **cat /etc/resolv.conf**

```
nameserver 192.168.0.2
```



On peut aller plus loin dans la configuration.

Il est possible d'attribuer de façon automatique une adresse IP à un ordinateur spécifique, par exemple les serveurs de l'entreprise, Web, Mail, NFS etc ...

Modifions le fichier de configuration comme cela : ATTENTION, il faudra modifier l'@Mac dans cette conf en regardant l'@ MAC du PC en question !!!

Sous GNS3 il est possible d'affecter une @MAC à un Host

ATTENTION nous sommes en virtuel pas en réel, en réel je rappelle que c'est impossible !!!

```
host Serveur_Web {
    hardware ethernet 16:64:33:51:CA:FE;
    fixed-address 172.31.1.16;
}
```

Redémarrer le serveur DHCP et **vérifier** que ça fonctionne

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xdcce1620
2	0.537836	172.31.1.254	172.31.1.16	DHCP	342	DHCP Offer - Transaction ID 0xdcce1620
3	1.000652	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xdcce1620
4	1.000948	172.31.1.254	172.31.1.16	DHCP	342	DHCP ACK - Transaction ID 0xdcce1620

J'ai bien l'IP 172.31.1.16 et l'@MAC est bien

```
gns3@box:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 16:64:33:51:CA:FE
          inet addr:172.31.1.16  Bcast:172.31.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:684 (684.0 B)  TX bytes:684 (684.0 B)
```

On peut aussi indiquer **l'IP du serveur DNS du site et de la passerelle par exemple !**

```
subnet 172.31.1.0 netmask 255.255.255.0 {
    range dynamic-bootp 172.31.1.3 172.31.1.253;
    option domain-name-servers 122.31.1.2; # DNS
    option routers 192.168.1.254; # Gateway
}
```

On peut aussi interdire de donner une IP à des stations dont on ne connaît pas les @MAC

7.2 Avec un routeur Mikrotik

On n'a pas encore étudié cette "bête"...

Mais dans de nombreuses petites entreprises ou chez vous c'est le routeur (votre box) qui fait office de serveur DHCP.

Alors c'est ce que nous ferons aussi en TP. Sans avoir vu le principe du routage !

Si vous voulez savoir comment paramétrer un routeur de type **Mikrotik** de manière qu'il soit serveur DHCP c'est fort simple, tout est là :

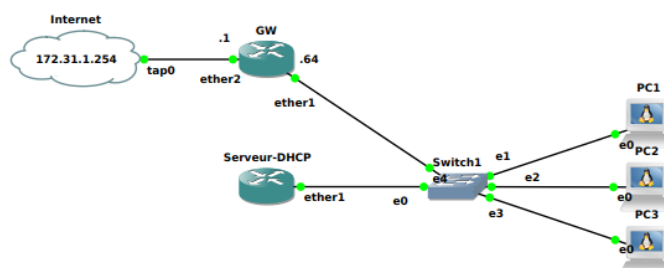
https://wiki.mikrotik.com/wiki/Manual:IP/DHCP_Server

Voici un exemple de configuration minimale

1. Le projet GNS3

Dans cette structure nous avons un Serveur DHCP réalisé à l'aide d'un routeur Mikrotik et d'un Passerelle (GW) qui est aussi un routeur Mikrotik.

Les 3 PC1 à 3 sont des clients DHCP sous Linux.



2. La conf du routeur

Ici le routeur ne fait pas office de routeur mais de serveur DHCP tout comme votre Box chez vous.

Je ne vous demande pas de savoir la réécrire mais simplement de savoir la lire.

On peut installer un serveur DHCP par exemple sur une machine de type Linux ou Windows et ce ne serait pas les mêmes commandes mais le principe serait le même !!!

Un lien pour savoir le faire sous debian : https://wiki.debian.org/fr/DHCP_Server

— # Par défaut sur les routeurs Mikrotik l'interface **ether1** est client **dhcp**, donc on retire !

```
[admin@MikroTik]/ip dhcp-client remove 0
```

— # Donner une adresse IP à l'interface ether1

```
[admin@MikroTik] ip address add address=192.168.0.1/24 interface=ether1
```

— # Puis on lance la configuration du serveur DHCP

```
[admin@MikroTik] ip dhcp-server> setup
```

Select interface to run DHCP server on

— # Sur quelle interface il rend ce service ou quelle interface écoute le port pour les requêtes DHCP
dhcp server interface : **ether1**

— # Quel réseau ?

Select network for DHCP addresses

dhcp address space : **192.168.0.0/24**

— # Quelle est @IP de la passerelle

Select gateway for given network

gateway for dhcp network : **192.168.0.1**

la plage d'@IP que le serveur donne

Select pool of ip addresses given out by DHCP server

addresses to give out : **192.168.0.3-192.168.0.254**

— # L'@IP de l'éventuel serveur DNS du site

Select DNS servers

dns servers : **192.168.0.2**

#Combien de temps l'@IP donnée est valide, ici un jour

Select lease time

lease time : **1d** <- correspond à 1 jour !

1. Conf, partie serveur DHCP, du routeur renvoie

Il donne

— une IP comprise entre 192.168.0.3 et 192.168.0.254

— Pour une durée de 1 jour

— l'@IP du DNS : 192.168.0.2

— l'@IP de la GW : 192.168.0.1

```
[admin@MikroTik] > /export
```

...

```
/ip pool
```

```
add name=dhcp_pool0 ranges=192.168.0.3-192.168.0.254
```

```
/ip dhcp-server
```

```
add address-pool=dhcp_pool0 disabled=no interface=ether1 lease-time=1d name=dhcp1
```

```
/ip dhcp-server network
```

```
add address=192.168.0.0/24 dns-server=192.168.0.2 gateway=192.168.0.1
```

2. La capture Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
4	27.932010	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x39114c33
6	28.576833	192.168.0.1	192.168.0.253	DHCP	342	DHCP Offer - Transaction ID 0x39114c33
7	28.946711	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x39114c33
8	28.954747	192.168.0.1	192.168.0.253	DHCP	342	DHCP ACK - Transaction ID 0x39114c33

On voit les 4 paquets DORA !

3. Sur le PC1

On voit qu'il a bien récupéré

— **une @IP 192.168.0.253/24**




```

— un DNS d'@IP 192.168.0.2/24
— une route par défaut vers la GW (Passerelle) d'@IP 192.168.0.64/24
gns3@box:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 0C:39:90:2F:00:00
          inet addr:192.168.0.253  Bcast:192.168.0.255  Mask:255.255.255.0
...
gns3@box:~$ cat /etc/resolv.conf
nameserver 192.168.0.2
...
gns3@box:~$ ip route ls
default via 192.168.0.64 dev eth0
127.0.0.1 dev lo  scope link
192.168.0.0/24 dev eth0  proto kernel  scope link  src 192.168.0.254

```

**Cela fonctionne puisque PC1 peut pinguer 172.31.1.1 mais pas le PC (TAP0) 172.31.1.254.
Vous aurez la réponse à cette problématique plus tard**

Ce n'est pas PC1 qui ne sait pas trouver le PC mais le PC qui ne sait pas trouver PC1!!!

```

gns3@box:~$ ping 172.31.1.1
PING 172.31.1.1 (172.31.1.1): 56 data bytes
64 bytes from 172.31.1.1: seq=0 ttl=64 time=13.835 ms
^C
— 172.31.1.1 ping statistics —
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 13.835/13.835/13.835 ms
gns3@box:~$ ping 172.31.1.254
PING 172.31.1.254 (172.31.1.254): 56 data bytes
^C
— 172.31.1.254 ping statistics —
3 packets transmitted, 0 packets received, 100% packet loss
gns3@box:~$

```

4. Si on désire connaître les @IP offertes par le serveur DHCP, il suffit de lancer sur le routeur Mikrotik la commande : **/ip dhcp-server lease print**

```

[admin@MikroTik] /ip service> /ip dhcp-server lease print
Flags: X – disabled, R – radius, D – dynamic, B – blocked
#  ADDRESS          MAG-ADDRESS  HOST-NAME  SERVER  RATE-LIMIT  STATUS
0 D 192.168.0.254    0C:39:90:2F:00:00 box        *2      bound
1 D 192.168.0.253    0C:39:90:2F:00:00 box        dhcp1    bound
2 D 192.168.0.252    0C:82:31:6E:00:00 box        dhcp1    bound
3 D 192.168.0.251    0C:93:4F:1F:00:00 box        dhcp1    bound

```

Je rappelle que dans la langue de Shakespeare : **To bind veut dire lier ou attacher**

Vous n'avez pas tout compris ?
Ce n'est pas encore trop grave !

Nous verrons cela plus en détails dans le TD et le TP!!!

8 Conclusion

Ce qui est à retenir et à savoir par cœur!!!

Le protocole DHCP s'appuie sur le protocole DHCP qui lui même s'appuie sur le protocole IP

1. La première **question** piège et méchante ☹

Comment le **host** qui n'a pas d'@IP fait-il pour s'appuyer sur le protocole IP pour faire une requête DHCP au serveur DHCP ?

Réponse : il s'octroie l'@IP n'importe qui **0.0.0.0/0**

Un serveur DHCP attribue



2. La seconde **question** piège et méchante ☺

Comment le **host** qui ne connaît pas l'@IP du serveur fait-il pour le trouver et faire une "belle" requête DHCP au serveur DHCP ?

Réponse : il envoie sa requête à tout le monde en utilisant l'@IP **255.255.255.255**

Un serveur DHCP attribue

- **une @IP**
- **un masque de réseau**
- **l'@IP du DNS du site local !**
- **l'@IP de la passerelle (GW) du site local afin de pouvoir quitter le LAN auquel il appartient et d'aller sur Internet ou autre part !**