

SAE – Installation & Configuration d'un serveur DHCP

Antonin LEMAIRE, Thomas NGO, Jules RENAUD-GRANGE

Table des matières

Les objectifs du projet.....	2
Les outils utilisés.....	2
La mise en place du serveur DHCP.....	3
Lancement des machines et du serveur.....	3
Modification des fichiers de configuration du serveur.....	4
Fichier isc-dhcp-server.....	5
Fichier dhcpd.conf.....	5
Lancement du serveur DHCP.....	6
Fichier interfaces des machines.....	6
Analyse de trame : Attribution d'une adresse IP.....	8
Analyse de trame : Renouvellement d'une adresse IP.....	9
Mise en place d'adresses IP fixes.....	10
Les Avantages et les Inconvénients du serveur DHCP.....	11

Les objectifs du projet

L'objectif principal de ce projet est l'installation et la configuration d'un serveur DHCP.

Un serveur DHCP (Dynamic Host Configuration Protocol) est donc un serveur qui a pour rôle d'attribuer automatiquement des adresses IP aux différentes machines d'un réseau informatique, sans qu'elles n'aient besoin de configuration manuelle.

Ces adresses IP seront attribuées dynamiquement, mais nous verrons qu'il est aussi possible d'attribuer une adresse statique à une machine en particulier.

Les outils utilisés

Pour cela nous utiliserons l'application Netkit, un environnement permettant la mise en place et la réalisation d'expériences de réseau sur différentes machines virtuels.

Nous utiliserons aussi l'application Wireshark qui nous permettra de lire et d'analyser les paquets transmis entre les machines et le serveur DHCP.

De plus, n'ayant pas la bonne version d'Ubuntu nécessaire à l'utilisation de Netkit (20.04), nous avons utilisé le logiciel de virtualisation Oracle VM VirtualBox, afin de travailler sur une machine virtuelle ayant la bonne version.

La mise en place du serveur DHCP

Voyons maintenant comment nous avons mis en place le serveur DHCP et confirmé son fonctionnement par des analyses de trames.

Lancement des machines et du serveur

La première chose à faire (après avoir installé Netkit) est de lancer les différentes machines, ici nous en avons cinq : le serveur et quatre machines (M1 à M4).

Le réseau se présente donc ainsi :

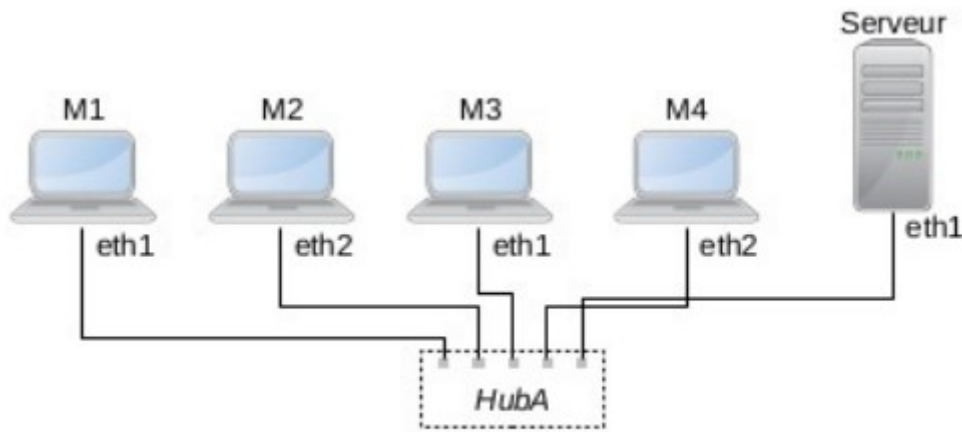


FIGURE 1 – Plan de réseau 1.

Afin de lancer chaque machine nous utilisons les commandes :

```
vstart M1 --eth1=A --mem=512
vstart M2 --eth2=A --mem=512
vstart M3 --eth1=A --mem=512
vstart M4 --eth2=A --mem=512
vstart Server --eth1=A --mem=512
```

Il est important de noter que les interfaces de chaque machine doivent toutes être connectées au même domaine de collision, ici appelé «A», sans quoi elles ne pourront pas communiquer entre elles même en ayant des adresses IP du même réseau.

SAE – INSTALLATION & CONFIGURATION D'UN SERVEUR DHCP

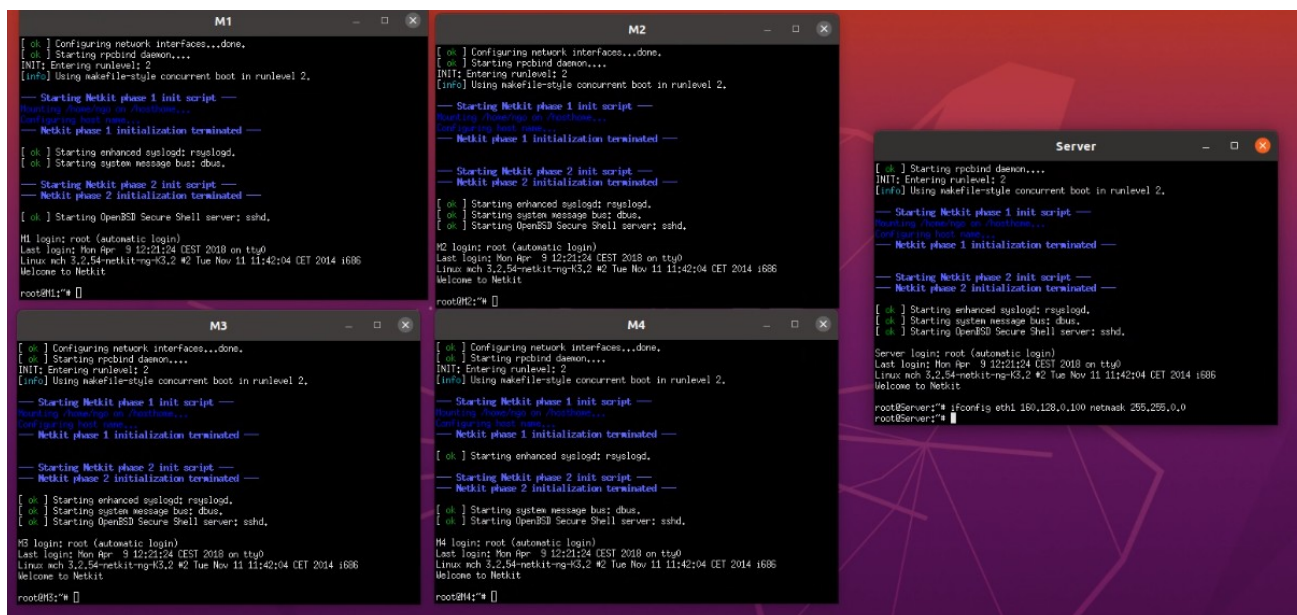
Nos différentes machines maintenant lancées, il est important de choisir l'adresse du réseau et son masque :

Nous avons choisi l'adresse de réseau 160.128.0.0 avec un masque 255.255.0.0.

Enfin nous avons attribué au serveur une adresse IP 160.128.0.100 avec le même masque via la commande :

```
ifconfig eth1 160.128.0.100 netmask 255.255.0.0
```

Voici donc ce que nous avons une fois tout cela effectué :



```
M1
[ ok ] Configuring network interfaces...done,
[ ok ] Starting rpcbind daemon....
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.

-- Starting Netkit phase 1 init script --
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenSSH Secure Shell server: sshd.
M1 login: root (automatic login)
Last login: Mon Apr  9 12:21:04 CEST 2018 on tty0
Linux mch 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit
root@M1:~#

M2
[ ok ] Configuring network interfaces...done,
[ ok ] Starting rpcbind daemon....
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.

-- Starting Netkit phase 1 init script --
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenSSH Secure Shell server: sshd.
M2 login: root (automatic login)
Last login: Mon Apr  9 12:21:04 CEST 2018 on tty0
Linux mch 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit
root@M2:~#

M3
[ ok ] Configuring network interfaces...done,
[ ok ] Starting rpcbind daemon....
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.

-- Starting Netkit phase 1 init script --
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenSSH Secure Shell server: sshd.
M3 login: root (automatic login)
Last login: Mon Apr  9 12:21:04 CEST 2018 on tty0
Linux mch 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit
root@M3:~#

M4
[ ok ] Configuring network interfaces...done,
[ ok ] Starting rpcbind daemon....
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.

-- Starting Netkit phase 1 init script --
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenSSH Secure Shell server: sshd.
M4 login: root (automatic login)
Last login: Mon Apr  9 12:21:04 CEST 2018 on tty0
Linux mch 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit
root@M4:~#

Server
[ ok ] Starting rpcbind daemon....
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.

-- Starting Netkit phase 1 init script --
[ ok ] Starting enhanced syslogd: rsyslogd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenSSH Secure Shell server: sshd.
Server login: root (automatic login)
Last login: Mon Apr  9 12:21:04 CEST 2018 on tty0
Linux mch 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 i686
Welcome to Netkit
root@Server:~# ifconfig eth1 160.128.0.100 netmask 255.255.0.0
root@Server:~#
```

Modification des fichiers de configuration du serveur

Sur le serveur nous avons deux fichiers à modifier afin de le faire fonctionner en tant que serveur DHCP :

Premièrement le fichier « /etc/default/isc-dhcp-server ».

Ensuite le fichier « /etc/dhcp/dhcpd.conf ».

Afin de les lire et d'écrire dedans nous utiliserons la commande :

```
nano chemin_du_fichier
```

Fichier isc-dhcp-server

Ce fichier sert à configurer les interfaces d'écoute du serveur, pour cela, rien du plus simple :

Il suffit de rajouter l'interface en question (ici, eth1) tout en bas du fichier, entre les guillemets.

```

Server
GNU nano 2.2.6      File: /etc/default/isc-dhcp-server      Modifie
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth1"

File Name to Write: /etc/default/isc-dhcp-server
G Get Help      M-I DOS Format  M-A Append     I-B Backup File
C Cancel        M-M Mac Format  M-P Prepend
  
```

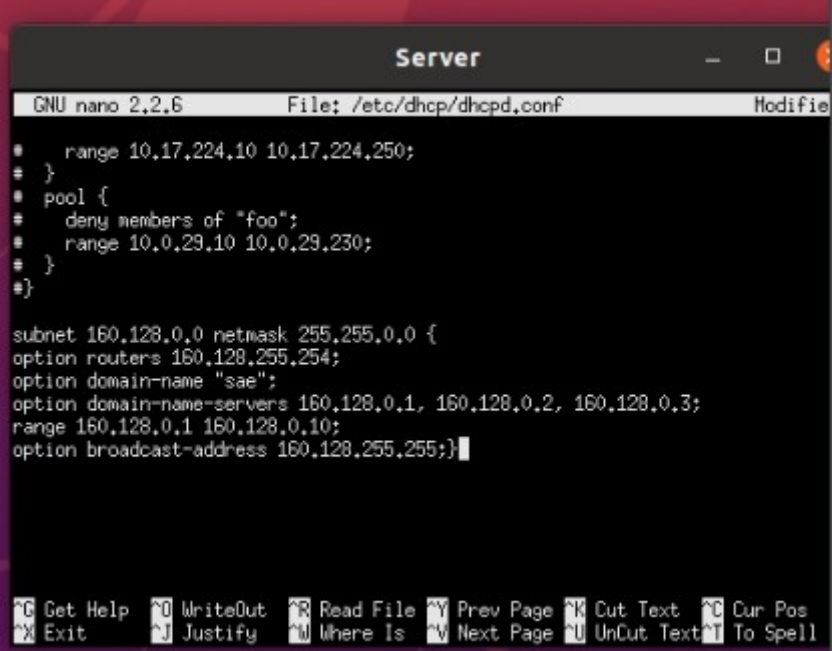
Fichier dhcpd.conf

Ce fichier sert à définir plusieurs paramètres liés au serveur DHCP, tel que le nom de domaine, les adresses des serveurs DNS, l'adresse de diffusion, mais surtout, ce qui nous intéresse le plus, la plage des adresses utilisables.

Nous avons ici différentes contraintes à respecter :

1. La plage des adresses doit contenir précisément 10 adresses, Nous avons donc choisi la plage de l'adresse 160.128.0.1 à l'adresse 160.128.0.10.
2. Le serveur doit fournir aux machines leur passerelle par défaut, qui doit avoir la plus grande adresse machine du réseau, soit 160.128.255.254. (c'est une simple adresse d'exemple, la machine ne sera pas créée).
3. Le serveur doit fournir aux machines le nom du domaine auxquels elles appartiennent, nous avons choisi le nom de domaine « sae ».
4. Le serveur doit fournir aux machines l'adresse IP des trois serveurs DNS du domaine, les trois adresses les plus petites du réseau, soit 160.128.0.1, 160.128.0.2 et 160.128.0.3.

On pourrait penser que la plage d'adresse entre en conflit avec les adresses des serveur DNS, mais celles-ci ne sont en réalité que des exemples et ne seront pas créées.



```

GNU nano 2.2.6      File: /etc/dhcp/dhcpd.conf      Modifie
#   range 10.17.224.10 10.17.224.250;
#   }
#   pool {
#       deny members of "foo";
#       range 10.0.29.10 10.0.29.230;
#   }
#}

subnet 160.128.0.0 netmask 255.255.0.0 {
option routers 160.128.255.254;
option domain-name "sae";
option domain-name-servers 160.128.0.1, 160.128.0.2, 160.128.0.3;
range 160.128.0.1 160.128.0.10;
option broadcast-address 160.128.255.255;}
```

Lancement du serveur DHCP

Maintenant les différents fichiers configurés, nous pouvons lancer le serveur, mais avant ça nous devons activer une capture tcpdump sur l'interface du serveur afin de pouvoir ensuite analyser les trames.

Pour cela nous utilisons la commande :

```
tcpdump -i eth1 -s 0 -w /hosthom/nom_du_fichier.cap &
```

Nous pouvons finalement lancer le serveur DHCP via la commande :

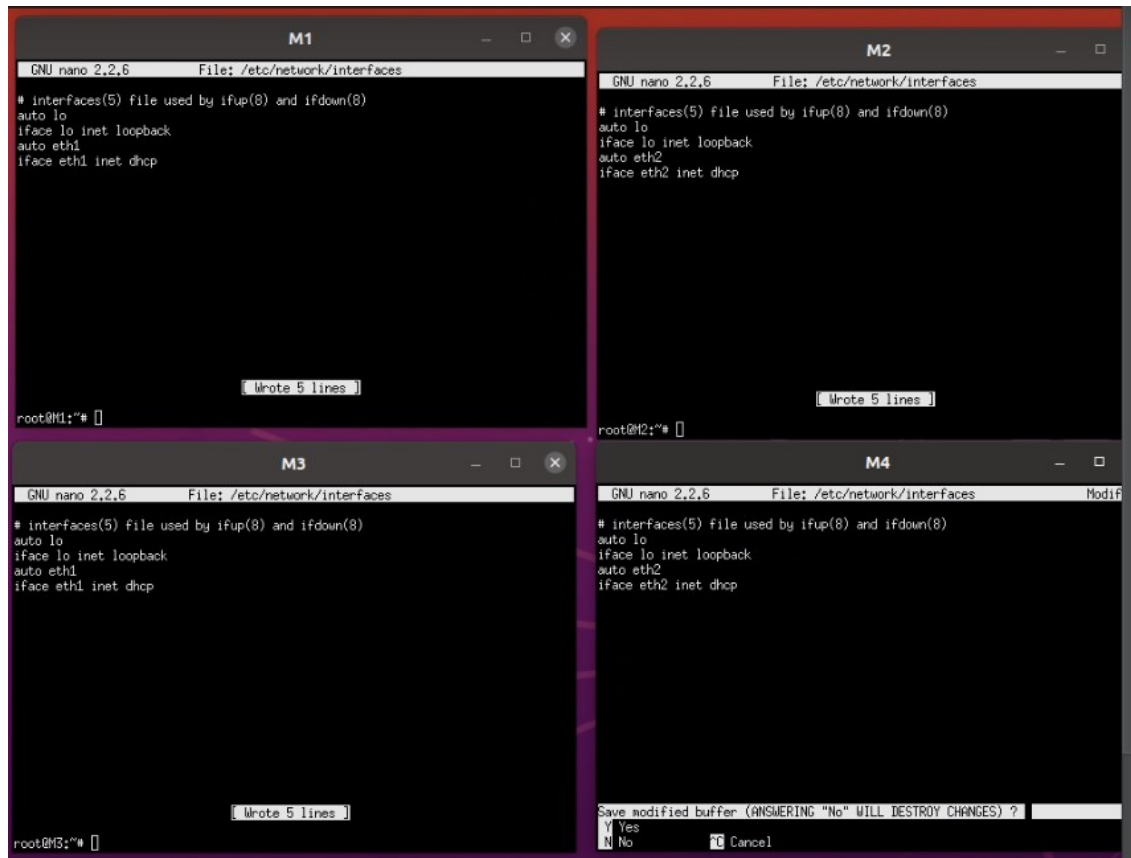
```
service isc-dhcp-server start
```

Fichier interfaces des machines

Le serveur est bien lancé, mais aucune des machines n'a été attribuée d'adresses IP, cela est tout à fait normal, c'est au machine de faire la demande, c'est pourquoi nous devons modifier le fichier /etc/network/interfaces de chaque machine afin qu'ils fassent la demande (et le renouvellement) d'un adresse IP eux-même.

SAE – INSTALLATION & CONFIGURATION D'UN SERVEUR DHCP

Dans chaque machine nous accédons au fichier en question avec la commande nano, et y rajoutons les deux lignes :



Afin d'avoir une adresse IP dynamique, on tape dans chaque machine la commande :

```
ifup ethx
```

(x : numéro de l'interface)

Ensuite nous pouvons vérifier si les machines ont bien une adresse IP avec la commande :

```
ifconfig -a
```

Qui affiche les interfaces et l'adresse IP qui y est liée.

Finalement nous pouvons vérifier si la communication entre les différentes machines est bien possible avec des ping, chaque machine a bien reçu son adresse et peut communiquer, nous pouvons alors terminer la capture tcpdump avec la commande :

```
killall tcpdump
```

Analyse de trame : Attribution d'une adresse IP

Nous pouvons maintenant ouvrir, à l'aide de Wireshark, le fichier cap que nous a donné la capture tcpdump afin d'analyser les trames liées à l'attribution d'une adresse IP.

On peut observer que l'attribution d'une adresse IP par un serveur DHCP se fait par 4 messages :

DHCP Discover, Offer, Request & ACK (acknowledgement).

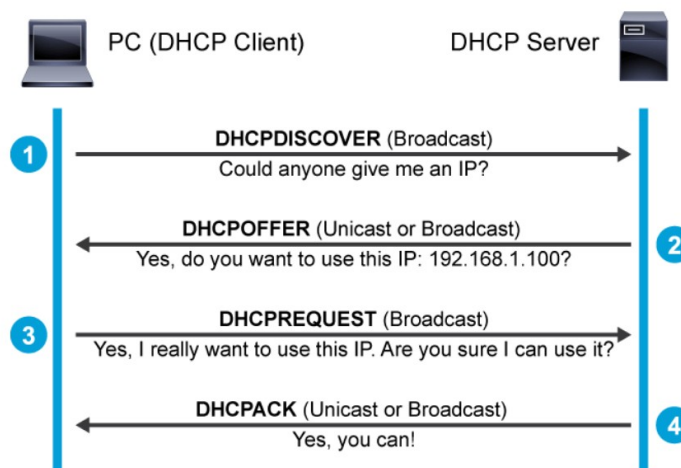
5	63.215401	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x4d6d5344
6	63.215536	160.128.0.100	160.128.0.2	ICMP	62 Echo (ping) request id=0x4c35, seq=0/0, ttl=64 (no response found!)
7	63.229581	::	ff02::16	ICMPv6	90 Multicast Listener Report Message v2
8	63.229582	::	ff02::1:ff00:c41e	ICMPv6	78 Neighbor Solicitation for fe80::a017:b6ff:feb0:c41e
9	64.224787	160.128.0.100	160.128.0.2	DHCP	342 DHCP Offer - Transaction ID 0x4d6d5344
10	64.225966	0.0.0.0	255.255.255.255	DHCP	342 DHCP Request - Transaction ID 0x4d6d5344
11	64.226150	160.128.0.100	160.128.0.2	DHCP	342 DHCP ACK - Transaction ID 0x4d6d5344

Discover : La machine demande au réseau (via l'adresse de broadcast) une adresse IP.

Offer : Le serveur DHCP propose une adresse IP actuellement disponible parmi la plage d'adresse.

Request : La machine demande à utiliser cette adresse.

ACK : Le serveur DHCP confirme que la machine peut utiliser l'adresse en question.



En observant les adresses MAC et IP des messages DCP Discover et Request, envoyés par la machine demandeuse d'IP, on peut voir que l'adresse MAC source est bien celle de la machine, mais l'adresse IP source est de 0.0.0.0, car cette machine n'en a pas encore une attribuée sur le réseau.

De plus, les adresses de destination sont celles de broadcast :

IP : 255.255.255.255 & MAC : ff:ff:ff:ff:ff:ff

```

▶ Ethernet II, Src: a2:17:b6:b0:c4:1e (a2:17:b6:b0:c4:1e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
▶ User Datagram Protocol, Src Port: 68, Dst Port: 67
▶ Dynamic Host Configuration Protocol (Discover)
    
```

La raison à l'utilisation de l'adresse de broadcast est simple, la machine n'a aucune idée de l'adresse du serveur DHCP sur le réseau, elle envoie donc la demande en broadcast, toutes les machines voient le message mais seul le serveur DHCP sera intéressé et y répondra.

Intéressons-nous maintenant aux ports UDP utilisés par ces messages :

Le port client (ici source) est le 68.

Le port serveur (ici destination) est le 67.

Ces ports sont inhabituels car ils ne sont jamais utilisés dans d'autres circonstances que les messages DHCP, ils sont réservés à ce type de transactions.

Analyse de trame : Renouvellement d'une adresse IP

Afin d'observer les messages liés à un renouvellement d'adresse IP, on modifie les valeurs du default-lease-time et du max-lease-time à 10 dans le fichier dhcpd.conf afin que le bail expire en 10 secondes.

On redémarre ensuite le serveur et les interfaces des différentes machines et on fait un nouvel enregistrement tcpdump.

DHCP	342	DHCP Request	- Transaction ID 0xd0098901
DHCP	342	DHCP ACK	- Transaction ID 0xd0098901
DHCP	342	DHCP Request	- Transaction ID 0x2c7daa44
DHCP	342	DHCP ACK	- Transaction ID 0x2c7daa44
DHCP	342	DHCP Request	- Transaction ID 0xd0098901
DHCP	342	DHCP ACK	- Transaction ID 0xd0098901
DHCP	342	DHCP Request	- Transaction ID 0x2c7daa44
DHCP	342	DHCP ACK	- Transaction ID 0x2c7daa44
DHCP	342	DHCP Request	- Transaction ID 0xd0098901
DHCP	342	DHCP ACK	- Transaction ID 0xd0098901

On peut voir qu'il nous manque certains messages pour chacune de ces transactions, le Discover et le Offer.

C'est en réalité tout à fait normal, les machines qui envoient ces Request n'ont pas besoin de « découvrir » le serveur DHCP sur le réseau, et n'ont pas besoin qu'on leur « offre » une adresse IP, ils en ont déjà une attribuée, ils font simplement une nouvelle requête une fois le bail arrivé à expiration afin de demander si ils peuvent continuer à utiliser cette adresse IP, ce à quoi le serveur va répondre par un ACK.

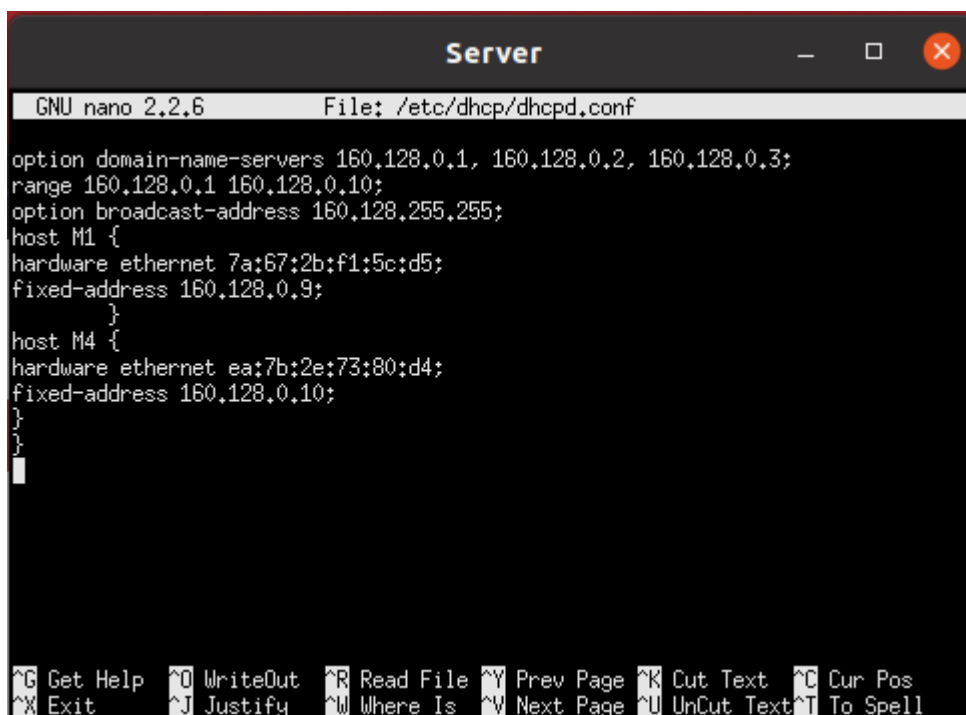
Mise en place d'adresses IP fixes

Afin de rendre l'adresse d'une machine fixe (ici M1 et M4), il faut modifier le fichier `dhcpd.conf` afin de reconnaître une machine qui ferait un Discover par son adresse MAC, car celle-ci est liée à la machine et ne changera pas.

On ajoute donc deux « host » qu'il faut nommer, ici M1 et M4, et on écrit leur adresses MAC, ainsi que l'adresse IP que nous voulons leur assigner automatiquement, respectivement 160.128.0.9 et 160.128.0.10.

Cela aura pour effet que, lorsque la machine fera un Discover, si son adresse MAC est bien la même que celle qu'on a entrée, le serveur DHCP ne lui proposera pas l'adresse la plus petite disponible sur sa plage d'adresses mais lui proposera l'adresse fixe qui lui a été assignée.

Cela peut être très utile dans le cas, par exemple, d'un serveur web dont l'adresse IP ne doit pas changer même si le serveur se déconnecte du réseau.



```
GNU nano 2.2.6 File: /etc/dhcp/dhcpd.conf

option domain-name-servers 160.128.0.1, 160.128.0.2, 160.128.0.3;
range 160.128.0.1 160.128.0.10;
option broadcast-address 160.128.255.255;
host M1 {
    hardware ethernet 7a:67:2b:f1:5c:d5;
    fixed-address 160.128.0.9;
}
host M4 {
    hardware ethernet ea:7b:2e:73:80:d4;
    fixed-address 160.128.0.10;
}
}
}
```

Les Avantages et les Inconvénients du serveur DHCP

Le serveur DHCP, ou Dynamic Host Configuration Protocol, offre plusieurs avantages :

Il simplifie grandement la configuration des réseaux en automatisant l'attribution des adresses IP aux dispositifs connectés. Au lieu de devoir configurer manuellement chaque appareil avec une adresse IP statique, le serveur DHCP assigne dynamiquement des adresses IP. Cela réduit donc la charge administrative et les risques d'erreurs humaines lors de la configuration du réseau.

Les adresses IP sont attribuées de manière dynamique aux appareils au fur et à mesure de leur connexion au réseau. Cela permet une utilisation plus efficace des adresses IP disponibles en évitant les conflits d'adresses et en réallouant automatiquement les adresses qui ne sont plus utilisées. De plus, cette gestion dynamique facilite la mobilité des appareils sur le réseau, car ils peuvent obtenir une adresse IP différente à chaque connexion sans nécessiter d'intervention manuelle.

Le serveur offre une flexibilité essentielle pour les réseaux en évolution. Il permet d'ajouter facilement de nouveaux périphériques au réseau sans avoir à reconfigurer manuellement les paramètres réseau pour chaque appareil. De plus, le serveur DHCP peut être reconfiguré pour répondre aux nouveaux besoins sans perturber le fonctionnement des appareils existants. Cette capacité à s'adapter permet au serveur DHCP d'être utilisé pour les réseaux d'entreprise en constante évolution.

Mais il offre aussi quelques inconvénients :

Le serveur DHCP peut être une cible potentielle pour les attaques malveillantes. En effet, étant un élément centralisé dans le réseau, un serveur DHCP compromis peut avoir des conséquences graves. Par exemple, un attaquant pourrait exploiter des vulnérabilités dans le serveur DHCP pour distribuer des adresses IP incorrectes. De plus, une attaque réussie contre le serveur DHCP peut paralyser l'ensemble du réseau, entraînant une interruption de service et des pertes de données.

Le fonctionnement du réseau dépend du serveur DHCP. En cas de défaillance de celui-ci, les nouveaux appareils ne pourront pas obtenir d'adresse IP valide, ce qui peut entraîner une interruption de service pour les utilisateurs qui tentent de se connecter au réseau. De plus, la gestion dynamique des adresses IP signifie que les appareils déjà connectés au réseau peuvent également rencontrer des problèmes si le serveur DHCP devient paralysé. Cette dépendance au serveur DHCP peut rendre le réseau vulnérable aux pannes et aux interruptions de service, surtout si des mesures de sauvegarde ne sont pas présentes.