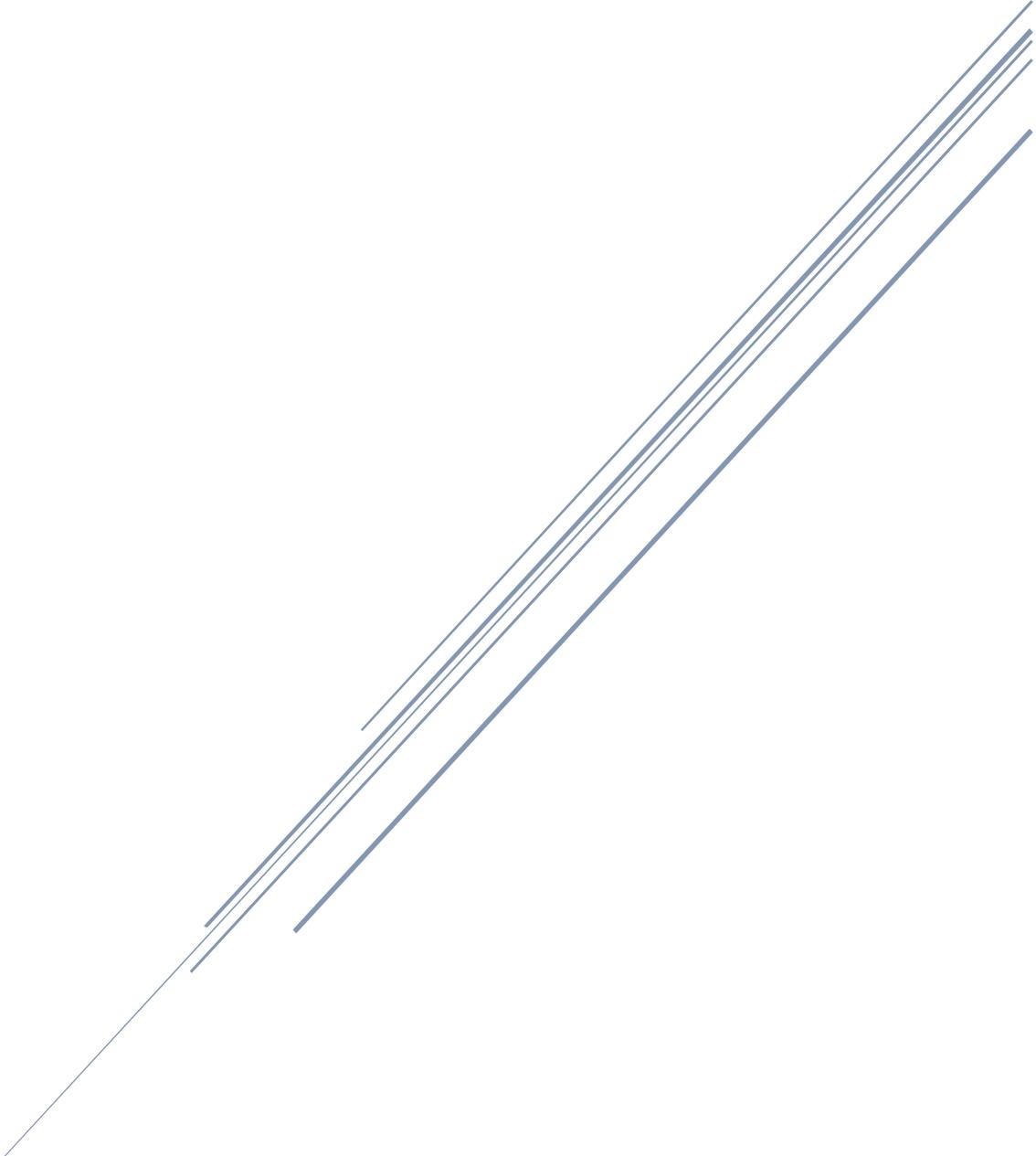


# PROJET SAE : INSTALLATION & CONFIGURATION D'UN SERVEUR DHCP

Compte-rendu



IUT de Champs-sur-Marne | BUT INF 1 2023-2024  
TELLE - SANTOS - ELANKEETHAN

## Table des matières

1. Description et explication des différentes étapes d'installation et de configuration du protocole ....	2
2. Description et explication des différentes étapes d'installation et de configuration du protocole ....	3
3. Description des fichiers utilisés lors de la configuration du protocole .....	10
4. Description de l'utilisation du protocole installé .....	11
5. Les utilisations potentielles du protocole .....	12

## 1. Description et explication des différentes étapes d'installation et de configuration du protocole

Le protocole que nous avons étudié est le *Dynamic Host Configuration Protocol* (DHCP), qui est un protocole réseau utilisé afin d'attribuer automatiquement des adresses IP et des paramètres de configuration réseau à des appareils. L'un des avantages de ce protocole est la simplicité de la gestion des adresses IP assignées dynamiquement, ce qui évite les conflits d'adresses. Un des autres avantages majeurs d'avoir un serveur DHCP est que l'on peut avoir plus de machines qu'il n'y a d'adresses IP disponibles, pour la simple et bonne raison que toutes les machines ne sont pas connectées simultanément, et que donc le service DHCP enlève et attribue les adresses IP lors d'une déconnexion et d'une connexion de machine. La sécurité pourrait aussi être un argument majeur en faveur de l'utilité du protocole DHCP, en effet, lorsqu'on navigue sur internet, on laisse des traces, et bien souvent, ces dernières sont liées à notre adresse IP, ce qui peut donc être problématique avec les sites modernes de géolocalisation d'adresses IP. Mais grâce au serveur DHCP, si l'adresse IP n'est pas la même à chaque fois qu'une machine se connecte, on diminue les chances de se faire tracer sur internet. Cependant, un inconvénient du service DHCP est qu'il dépend d'un serveur centralisé, ce qui peut entraîner des problèmes en cas d'une panne de ce serveur, les machines seraient dans l'incapacité d'avoir une adresse IP, sauf si l'utilisateur sait faire une configuration manuelle. Un autre problème lié au service DHCP est qu'il est dit "sans intervention", ce qui signifie que si une personne tierce parvient à implémenter un serveur DHCP non autorisé, ce qui permettrait d'envahir le réseau à des fins illégales ou d'accéder au réseau de manière aléatoire sans autorisation explicite.

## 2. Description et explication des différentes étapes d'installation et de configuration du protocole

Afin de créer les machines virtuelles nous avons utilisé l'outil Netkit.

```
galaktik@Debian11:~/Netkit/netkit$ export NETKIT_HOME=/home/galaktik/Netkit/netkit
galaktik@Debian11:~/Netkit/netkit$ export MANPATH=:$NETKIT_HOME/man
galaktik@Debian11:~/Netkit/netkit$ export PATH=$NETKIT_HOME/bin:$PATH
```

Figure 1: Création des chemins pour Netkit

```
galaktik@Debian11:~/Netkit/netkit$ sudo apt-get install xterm
```

Figure 2: Installation de la console Xterm, nécessaire à Netkit

```
galaktik@Debian11:~/Netkit/netkit$ ./check_configuration.sh
> Checking path correctness... passed.
> Checking environment... passed.
> Checking for availability of man pages... passed.
> Checking for proper directories in the PATH... passed.
> Checking for availability of auxiliary tools:
awk           : ok
basename     : ok
date         : ok
dirname      : ok
find         : ok
getopt       : ok
grep         : ok
head         : ok
id           : ok
kill         : ok
ls           : ok
lsof         : ok
ps           : ok
readlink     : ok
wc           : ok
port-helper  : ok
tunctl       : ok
uml_mconsole : ok
uml_switch   : ok
passed.
> Checking for availability of terminal emulator applications:
xterm         : found
konsole       : not found
gnome-terminal : found
passed.
> Checking filesystem type... passed.
> Checking whether 32-bit executables can run... passed.
[ READY ] Congratulations! Your Netkit setup is now complete!
          Enjoy Netkit!
```

Figure 3: La configuration de Netkit est réalisée

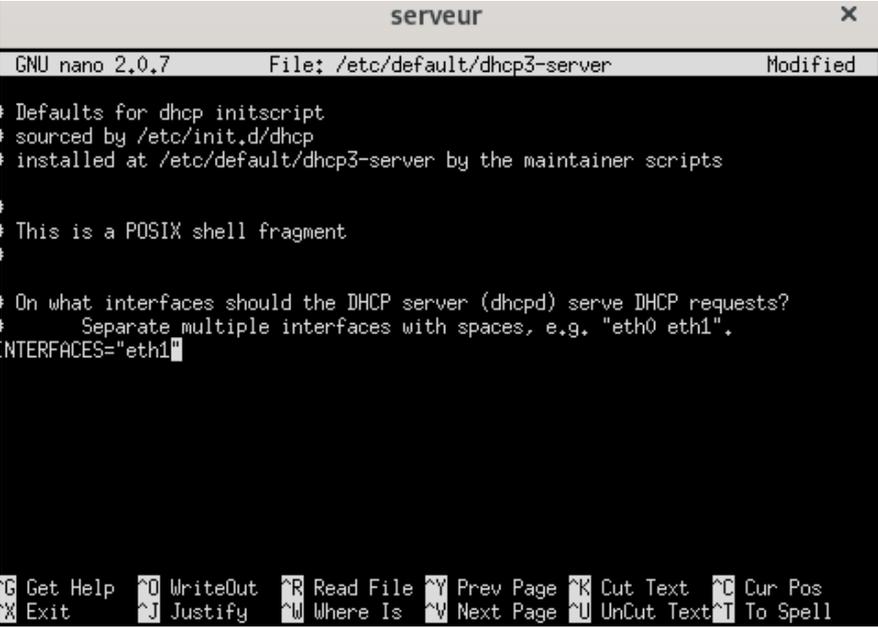
Après avoir installé Netkit il fallait choisir une adresse de réseau afin de réaliser cette configuration. Nous avons choisi comme adresse : 192.168.1.0/24 car nous avons déduit que nous n'aurions pas besoin d'un grand nombre d'adresses pour les ordinateurs. Le choix d'une adresse réseau de classe C, donc le masque est 255.255.255.0, s'est alors vite imposé.

Ensuite nous avons créé le serveur DHCP et nous l'avons configuré en renseignant la première adresse IP du réseau disponible.

```
serveur login: root (automatic login)
serveur:~# ifconfig eth1 192.168.10.1 netmask 255.255.255.0
serveur:~#
```

Figure 4: Configuration de l'adresse IP du serveur DHCP

Par la suite, nous avons modifié le fichier /etc/default/dhcp3-server afin de configurer l'interface d'écoute du serveur. C'est l'interface qui permet la connexion entre les appareils en convertissant les données, en gérant le flux d'information, en prenant en charge différents protocoles, et en assurant la sécurité des échanges. Il a donc fallu renseigner l'interface d'écoute comme étant « eth1 ».



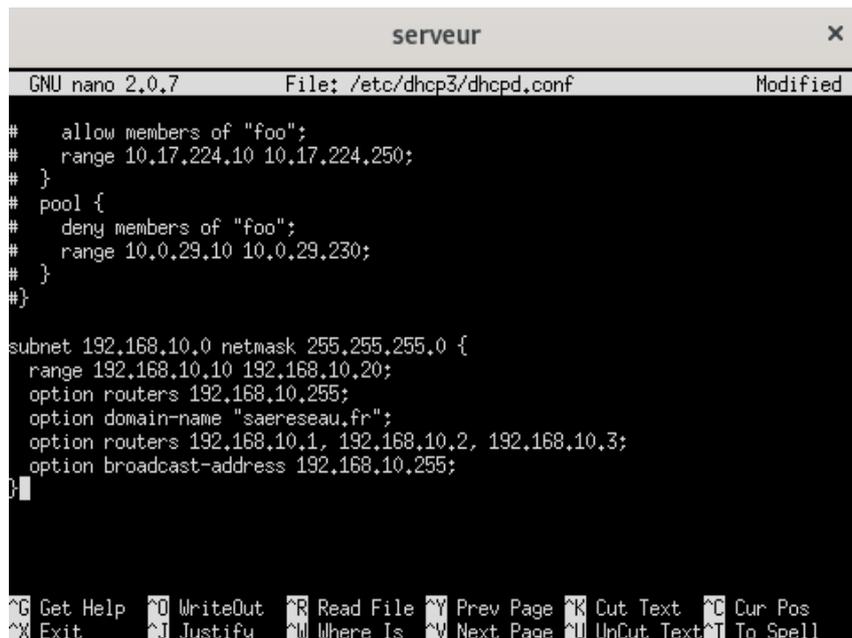
```
serveur
GNU nano 2.0.7 File: /etc/default/dhcp3-server Modified
Defaults for dhcp initscript
sourced by /etc/init.d/dhcp
installed at /etc/default/dhcp3-server by the maintainer scripts

This is a POSIX shell fragment

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

Figure 5: Fichier de configuration des interfaces d'écoutes

Puis le moment vient où l'on doit configurer un sous-réseau pour le serveur DHCP. Il va alors falloir rentrer plusieurs paramètres, à commencer par l'adresse du réseau, ici, 192.168.10.0, et son masque de sous-réseau, 255.255.255.0. Ensuite il faut rentrer un paramètre obligatoire qui est la plage d'adresse IP attribuable, dans notre cas, nous avons décidé que cette plage serait entre 192.168.10.10 et 192.168.10.20, afin de respecter la contrainte de 10 adresses IP attribuables. Comme l'indique le début de ligne des autres paramètres, les options suivantes sont optionnelles. Nous avons rajouté au total 4 options, la passerelle par défaut, qui est 192.168.10.254, le nom de domaine, dont le nom a été choisi arbitrairement et est « saeresseau.fr », ainsi que les adresses IP de ses 3 serveurs DNS fictifs, 192.168.10.2, 192.168.10.3 et 192.168.10.4. Pour finir, nous avons rajouté une ligne indiquant que l'adresse de broadcast est 192.168.10.255.



```
serveur x
GNU nano 2.0.7 File: /etc/dhcp3/dhcpd.conf Modified
# allow members of "foo";
# range 10,17,224,10 10,17,224,250;
# }
# pool {
# deny members of "foo";
# range 10,0,29,10 10,0,29,230;
# }
#}

subnet 192,168,10,0 netmask 255,255,255,0 {
range 192,168,10,10 192,168,10,20;
option routers 192,168,10,255;
option domain-name "saeresseau.fr";
option routers 192,168,10,1, 192,168,10,2, 192,168,10,3;
option broadcast-address 192,168,10,255;
}
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figure 6: Fichier de configuration du serveur DHCP

Nous avons ajouté les lignes « auto eth1 » et « iface eth1 inet dhcp » dans les fichiers /etc/network/interfaces des machines du réseau pour que quand les interfaces des machines s'activent, elles puissent automatiquement faire une requête au serveur DHCP pour obtenir une configuration réseau.



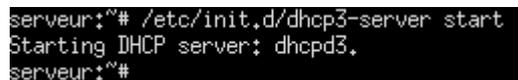
```
M1
GNU nano 2.0.7 File: /etc/network/interfaces
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.

# The loopback network interface
auto lo
iface lo inet loopback

auto eth1
iface eth1 inet dhcp
```

Figure 7: Fichier de configuration de l'interface de la machine M1

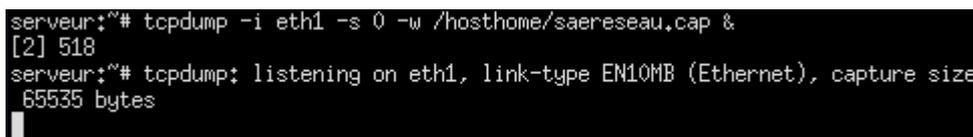
Une fois toutes les configurations finies, nous avons démarré le serveur DHCP grâce au fichier du dossier init.d nommé « dhcp3-server » et l'argument « start ».



```
serveur:~# /etc/init.d/dhcp3-server start
Starting DHCP server: dhcpd3.
serveur:~#
```

Figure 8: Commande de démarrage du serveur DHCP

Nous avons ensuite exécuté la capture « tcpdump » afin d'enregistrer toutes les actions que le serveur DHCP va effectuer, en particulier l'attribution d'adresses IP.



```
serveur:~# tcpdump -i eth1 -s 0 -w /hoshome/saereseau.cap &
[2] 518
serveur:~# tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size
65535 bytes
```

Figure 9: Commande de démarrage de la capture tcpdump

Puis nous avons activé les interfaces des machines, afin qu'elles puissent recevoir une adresse IP qui sera donnée dynamiquement par le serveur. Voici l'exemple sur la machine 1 qui reçoit l'adresse IP 192.168.10.10.

```
M1:~# ifup eth1
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth1/f6:8f:1b:56:49:23
Sending on   LPF/eth1/f6:8f:1b:56:49:23
Sending on   Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 7
DHCPOFFER from 192.168.10.1
DHCPREQUEST on eth1 to 255.255.255.255 port 67
DHCPACK from 192.168.10.1
bound to 192.168.10.10 -- renewal in 292 seconds.
M1:~#
```

Figure 10: La machine M1 obtient automatiquement son plan d'adressage réseau

On peut donc voir que la configuration des machines du réseau a bien fonctionné car la configuration des interfaces réseau correspond à ce qui a été défini dans le fichier de configuration du serveur DHCP. Le serveur DHCP aussi est correctement configuré, car il a été en mesure de répondre à une requête venant d'une machine, lui demandant l'attribution d'une adresse IP. La connexion entre la M1 et le serveur est bien établie.

```
M1:~# ifconfig
eth1      Link encap:Ethernet  HWaddr f6:8f:1b:56:49:23
          inet addr:192.168.10.10  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::f48f:1bff:fe56:4923/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:816 (816.0 B)  TX bytes:13898 (13.5 KiB)
          Interrupt:5

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1100 (1.0 KiB)  TX bytes:1100 (1.0 KiB)

M1:~#
```

Figure 11: Configuration réseau de M1

La connexion entre M2 et le serveur a bien été établie :

```
M2:~# ifconfig
eth2      Link encap:Ethernet  HWaddr 4e:26:c3:cc:46:91
          inet addr:192.168.10.11  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::4c26:c3ff:fecc:4691/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:96 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15440 (15.0 KiB)  TX bytes:2314 (2.2 KiB)
          Interrupt:5

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:200 (200.0 B)  TX bytes:200 (200.0 B)
```

Figure 12: Configuration réseau de M2

Nous avons répété la procédure pour les autres machines afin qu'elles puissent communiquer avec l'ensemble du réseau.

Maintenant que les machines, ainsi que le serveur, sont bien configurés et que les machines communiquent bien avec le serveur. On doit tester si elles peuvent bien communiquer avec l'ensemble du réseau. Nous avons donc utilisé la commande ping pour envoyer des paquets vers les autres machines du réseau. Voici un exemple de communication entre les machines M1 et M2 :

```
M1:~# ping 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data:
64 bytes from 192.168.10.11: icmp_seq=1 ttl=64 time=3.32 ms
64 bytes from 192.168.10.11: icmp_seq=2 ttl=64 time=0.964 ms
64 bytes from 192.168.10.11: icmp_seq=3 ttl=64 time=3.14 ms
64 bytes from 192.168.10.11: icmp_seq=4 ttl=64 time=0.854 ms
64 bytes from 192.168.10.11: icmp_seq=5 ttl=64 time=1.06 ms
64 bytes from 192.168.10.11: icmp_seq=6 ttl=64 time=0.812 ms
64 bytes from 192.168.10.11: icmp_seq=7 ttl=64 time=1.18 ms
64 bytes from 192.168.10.11: icmp_seq=8 ttl=64 time=0.678 ms
64 bytes from 192.168.10.11: icmp_seq=9 ttl=64 time=1.78 ms
64 bytes from 192.168.10.11: icmp_seq=10 ttl=64 time=0.656 ms
64 bytes from 192.168.10.11: icmp_seq=11 ttl=64 time=0.827 ms
^C
--- 192.168.10.11 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 1011ms
rtt min/avg/max/mdev = 0.656/1.390/3.326/0.920 ms
```

Figure 13: Envoi de paquet PING de la machine M1 vers M2, avec une réponse de M2

Les communications sont bel et bien possibles entre les machines. Maintenant, nous pouvons arrêter la capture « tcpdump » lancé précédemment et analyser, à l'aide du logiciel Wireshark, les différentes trames qui ont été enregistrées (Soutenance orale).

10.000000	::	ff02::16	ICMPv6	90 Multicast Listener Report Message v2
20.000001	::	ff02::1:ff56:4923	ICMPv6	78 Neighbor Solicitation for fe80::f48f:1bff:fe56:4923
31.221657	fe80::f48f:1bff:fe56:4923	ff02::2	ICMPv6	70 Router Solicitation from f6:8f:1b:56:49:23
4.4.942896	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x69cf813e
5.5.866241	f6:51:69:d6:0d:dd	Broadcast	ARP	42 Who has 192.168.10.10? Tell 192.168.10.1
6.5.224484	fe80::f48f:1bff:fe56:4923	ff02::2	ICMPv6	70 Router Solicitation from f6:8f:1b:56:49:23
7.5.649987	192.168.10.1	192.168.10.10	DHCP	342 DHCP Offer - Transaction ID 0x69cf813e
8.5.676515	0.0.0.0	255.255.255.255	DHCP	342 DHCP Request - Transaction ID 0x69cf813e
9.5.770666	192.168.10.1	192.168.10.10	DHCP	342 DHCP ACK - Transaction ID 0x69cf813e
10.6.964130	f6:51:69:d6:0d:dd	Broadcast	ARP	42 Who has 192.168.10.10? Tell 192.168.10.1
11.7.064186	f6:51:69:d6:0d:dd	Broadcast	ARP	42 Who has 192.168.10.10? Tell 192.168.10.1
12.7.064814	f6:8f:1b:56:49:23	f6:51:69:d6:0d:dd	ARP	42 192.168.10.10 is at f6:8f:1b:56:49:23
13.7.064621	192.168.10.1	192.168.10.10	ICMP	62 Echo (ping) request id=0xa0ca, seq=0/0, ttl=64 (reply in 14)
14.7.066306	192.168.10.10	192.168.10.1	ICMP	62 Echo (ping) reply id=0xa0ca, seq=0/0, ttl=64 (request in 13)
15.14.105000	fe80::f48f:1bff:fe56:4923	ff02::16	ICMPv6	90 Multicast Listener Report Message v2
16.14.105002	fe80::f48f:1bff:fe56:4923	ff02::2	ICMPv6	70 Router Solicitation from f6:8f:1b:56:49:23
17.14.105044	f6:8f:1b:56:49:23	f6:51:69:d6:0d:dd	ARP	42 Who has 192.168.10.1? Tell 192.168.10.10
18.14.105026	f6:51:69:d6:0d:dd	f6:8f:1b:56:49:23	ARP	42 192.168.10.1 is at f6:51:69:d6:0d:dd
19.106.672670	::	ff02::16	ICMPv6	90 Multicast Listener Report Message v2
20.106.672672	::	ff02::1:ffcc:4691	ICMPv6	78 Neighbor Solicitation for fe80::4c26:c3ff:fecc:4691
21.107.672651	fe80::4c26:c3ff:fecc:4691	ff02::2	ICMPv6	70 Router Solicitation from 4e:26:c3:cc:46:91
22.109.045780	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x6f678a4c
23.109.113364	f6:51:69:d6:0d:dd	Broadcast	ARP	42 Who has 192.168.10.11? Tell 192.168.10.1
24.109.496065	fe80::4c26:c3ff:fecc:4691	ff02::16	ICMPv6	90 Multicast Listener Report Message v2
25.109.506383	192.168.10.1	192.168.10.11	DHCP	342 DHCP Offer - Transaction ID 0x6f678a4c
26.109.644884	0.0.0.0	255.255.255.255	DHCP	342 DHCP Request - Transaction ID 0x6f678a4c
27.109.740520	192.168.10.1	192.168.10.11	DHCP	342 DHCP ACK - Transaction ID 0x6f678a4c
28.110.114104	f6:51:69:d6:0d:dd	Broadcast	ARP	42 Who has 192.168.10.11? Tell 192.168.10.1
29.111.114186	f6:51:69:d6:0d:dd	Broadcast	ARP	42 Who has 192.168.10.11? Tell 192.168.10.1
30.112.762381	fe80::4c26:c3ff:fecc:4691	ff02::2	ICMPv6	70 Router Solicitation from 4e:26:c3:cc:46:91
31.122.505575	fe80::4c26:c3ff:fecc:4691	ff02::2	ICMPv6	70 Router Solicitation from 4e:26:c3:cc:46:91
32.127.097867	f6:8f:1b:56:49:23	Broadcast	ARP	42 Who has 192.168.10.11? Tell 192.168.10.10
33.127.099229	4e:26:c3:cc:46:91	f6:8f:1b:56:49:23	ARP	42 192.168.10.11 is at 4e:26:c3:cc:46:91
34.127.099229	192.168.10.10	192.168.10.11	ICMP	62 Echo (ping) request id=0x5023, seq=0/0, ttl=64 (reply in 26)

Désormais, il ne nous reste plus qu'une étape à faire, celle de l'attribution d'adresses IP fixes. Ici, on part de l'hypothèse que l'on veut que certaines machines de notre réseau aient une adresse IP fixe. Mais alors, comment faire pour que les machines M1 à M4 aient une adresse IP fixe ?

Il faudrait, dans le fichier /etc/dhcp3/dhcpd.conf du serveur DHCP rajouter une entrée pour la réservation DHCP. Dans le cas de la machine M1, l'entrée serait la suivante :

```
host M1 {
    hardware ethernet f6:8f:1b:56:49:23;
    fixed-address 192.168.10.11;
}
```

Ici, on définit dans le serveur DHCP que l'hôte M1, avec pour adresse MAC (Media Access Control) f6:8f:1b:56:49:23, a une adresse IP qui lui est réservée, à savoir 192.168.10.11.

Malheureusement, si l'on veut que certaines machines aient une adresse IP fixe, il faut faire des entrées manuellement dans le fichier de configuration du serveur DHCP. À noter aussi que le rôle de l'adresse MAC est très important, c'est un identifiant unique pour l'interface réseau de la machine M1.

### 3. Description des fichiers utilisés lors de la configuration du protocole

Durant cette mise en place du serveur DHCP, il a été nécessaire de modifier 3 différents fichiers, 1 côté client et 2 côté serveurs. Tout d'abord, le fichier de la machine du client doit seulement se voir rajouter deux lignes dans le fichier de configuration des interfaces `/etc/network/interfaces`, ainsi, il pourra faire une requête à un serveur DHCP qui lui fournira son adresse IP. Du côté du serveur, il faut modifier tout d'abord le fichier `/etc/default/dhcp3-server`, pour tout simplement configurer l'interface d'écoute des requêtes DHCP. Puis, nous devons modifier le fichier `/etc/dhcp3/dhcpd.conf`, où nous indiquerons toutes les entrées nécessaires au bon fonctionnement du serveur, telles que : l'adresse du réseau et son masque de réseau, le nom du serveur DNS et ses adresses IP, les adresses IP que l'on peut distribués, une adresse de broadcast, et bien d'autres.

## 4. Description de l'utilisation du protocole installé

Ici, le protocole a été utilisé dans un petit réseau composé de 4 machines clientes, qui, lorsqu'elles ont voulu configurer leur interface, ont fait appel au serveur DHCP. Cela permet donc d'avoir un plan d'adressage optimisé et surtout automatique. Dans le cadre d'un réseau de 4 machines, cela peut sembler ne pas être utile, en revanche, si l'on prend de plus grands exemples, comme une université, un hôpital ou même encore une entreprise, on n'éprouve pas l'envie de configurer manuellement chaque appareil qui se connecte sur le réseau.

## 5. Les utilisations potentielles du protocole

Le protocole DHCP (Dynamic Host Configuration Protocol) est largement utilisé pour attribuer automatiquement des adresses IP et d'autres paramètres réseau à des appareils clients dans un réseau. Parmi les utilisations potentielles du protocole DHCP, ainsi que les vérifications et les données qu'il peut gérer, on peut retrouver :

1. Attribution automatique d'adresses IP : DHCP permet de distribuer des adresses IP de manière dynamique aux appareils clients à chaque fois qu'ils se connectent au réseau. Une vérification importante est de s'assurer que les adresses IP attribuées ne sont pas en conflit avec celles déjà utilisées dans le réseau. Le protocole peut également fournir des informations sur la disponibilité des adresses IP dans la plage d'adresse gérée par le serveur DHCP.
2. Configuration des paramètres réseau : En plus des adresses IP, DHCP peut également fournir d'autres paramètres réseau tels que les adresses de passerelle, les serveurs DNS, l'adresse de diffusion réseau, etc. Il est important de vérifier que ces paramètres sont correctement configurés et qu'ils sont compatibles avec l'infrastructure réseau existante.
3. Gestion des baux DHCP : Le protocole DHCP utilise des baux pour attribuer temporairement des adresses IP aux clients, c'est-à-dire que les machines ont une adresse IP limitée dans le temps. Il est important de vérifier la durée des baux et de s'assurer que les renouvellements de bail se déroulent correctement pour éviter les interruptions de connectivité. Le protocole peut fournir des informations sur les baux actifs, leur durée restante, et permettre d'effectuer des opérations telles que la libération et le renouvellement des baux.
4. Sécurité : DHCP peut être sujet à des attaques telles que le détournement de bail (DHCP spoofing) ou les attaques de déni de service (DDoS). Il est donc important de mettre en place des mesures de sécurité telles que l'authentification des serveurs DHCP et la surveillance du trafic DHCP pour détecter toute activité suspecte.

En résumé, le protocole DHCP permet de gérer efficacement l'attribution des adresses IP et d'autres paramètres réseau dans un environnement informatique. En effectuant des vérifications appropriées et en surveillant son fonctionnement, on peut assurer le bon fonctionnement du réseau et détecter rapidement tout problème éventuel.