




2020-2021

Epreuve-E6 BTS SIO



Xavier Tofili
IRIS-STRASBOURG
2020-2021

	BTS SIO		Orange WF
	Services Informatiques aux Organisations		
	Option	SISR	
	Session	2021	

TOFILI Xavier	Activité professionnelle N°	6
----------------------	------------------------------------	---

NATURE DE L'ACTIVITE	Mise en place d'un DHCP via Raspberry Pi
Contexte	<p>Dans le cadre de mon stage, j'ai eu la mission de configurer un serveur DHCP via une Raspberry Pi. L'infrastructure disposait de 3 réseaux à servir :</p> <ul style="list-style-type: none"> • La direction • La production • La R&D
Objectifs	Avoir un serveur DHCP
Lieu de réalisation	Orange WF – Wallis-et-Futuna

DESCRIPTION DE LA SOLUTION RETENUE	
Conditions initiales	Infrastructure sans serveur DHCP.
Conditions finales	Infrastructure avec serveur DHCP.
Outils utilisés	Raspberry Pi

CONDITIONS DE REALISATION	
Matériels	Disposer d'un clavier, une souris et un Raspberry Pi.
Logiciels	Balenaetcher
Contraintes	Avoir des bases Linux, savoir configurer un DHCP.

COMPETENCES MISES EN OEUVRE POUR CETTE ACTIVITE PROFESSIONNELLE	
A1.1.1	Analyse du cahier des charges d'un service à produire
A1.2.4	Détermination des tests nécessaires à la validation d'un service
A1.4.1	Participation à un projet
A3.1.1	Proposition d'une solution d'infrastructure
A3.1.2	Maquettage et prototypage d'une solution d'infrastructure
A3.2.1	Installation et configuration d'éléments d'infrastructure
A3.3.1	Administration sur site ou à distance des éléments d'un réseau, de serveurs...
A4.1.8	Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés
A4.1.9	Rédaction d'une documentation technique.

Sommaire

Cahier des charges.....	3
Description de l'existant.....	3
Expression du besoin	3
Analyse et proposition de réponse.....	3
Plan de travail.....	3
Mise en œuvre.....	4
Etape 1	5
Etape 2	5
Etape 3	8

Cahier des Charges

Description de l'existant

L'entreprise dispose donc de 3 réseaux qui jusqu'à présent communiquer via des IPs en statique sur leurs différents postes de travaux.

Les réseaux :

- La direction : VLAN10 -> 192.168.10.0/24
- La production : VLAN20 -> 192.168.20.0/24
- La R&D (recherche et développement) : VLAN30 -> 192.168.30.0/24

Expression du besoin

A vu de l'arrivée d'un nouveau secteur d'activité ainsi que de nouveau employé, la société cliente a décidé de faire évoluer leur infrastructure réseau en faisant l'acquisition d'un serveur DHCP. L'entreprise cliente étant nouvelle, elle ne dispose donc pas de beaucoup de moyen financier, nous demande de leur proposer une solution fiable tout en respectant une fourchette de prix allant de 300 à 500€. Elle fait donc appel à notre entreprise pour la mise en place et le paramétrage de leur équipement dans leur infrastructure.

Analyse et proposition :

Après une analyse de leur infrastructure est de leur demande. Nous avons estimé que pour répondre à leur besoin, tout en respectant la fourchette de prix demander nous mettrons on place leur serveur DHCP sur un Raspberry Pi avec comme OS (operating system) Debian 10.

Plan de travail

Pour la réalisation de cette mission, nous avons suivis le plan de travail suivant :

- **Etape 1** : Préparation du serveur
- **Etape 2** : Installation du serveur DHCP
- **Etape 3** : Test du serveur

Mise en œuvre

Introduction

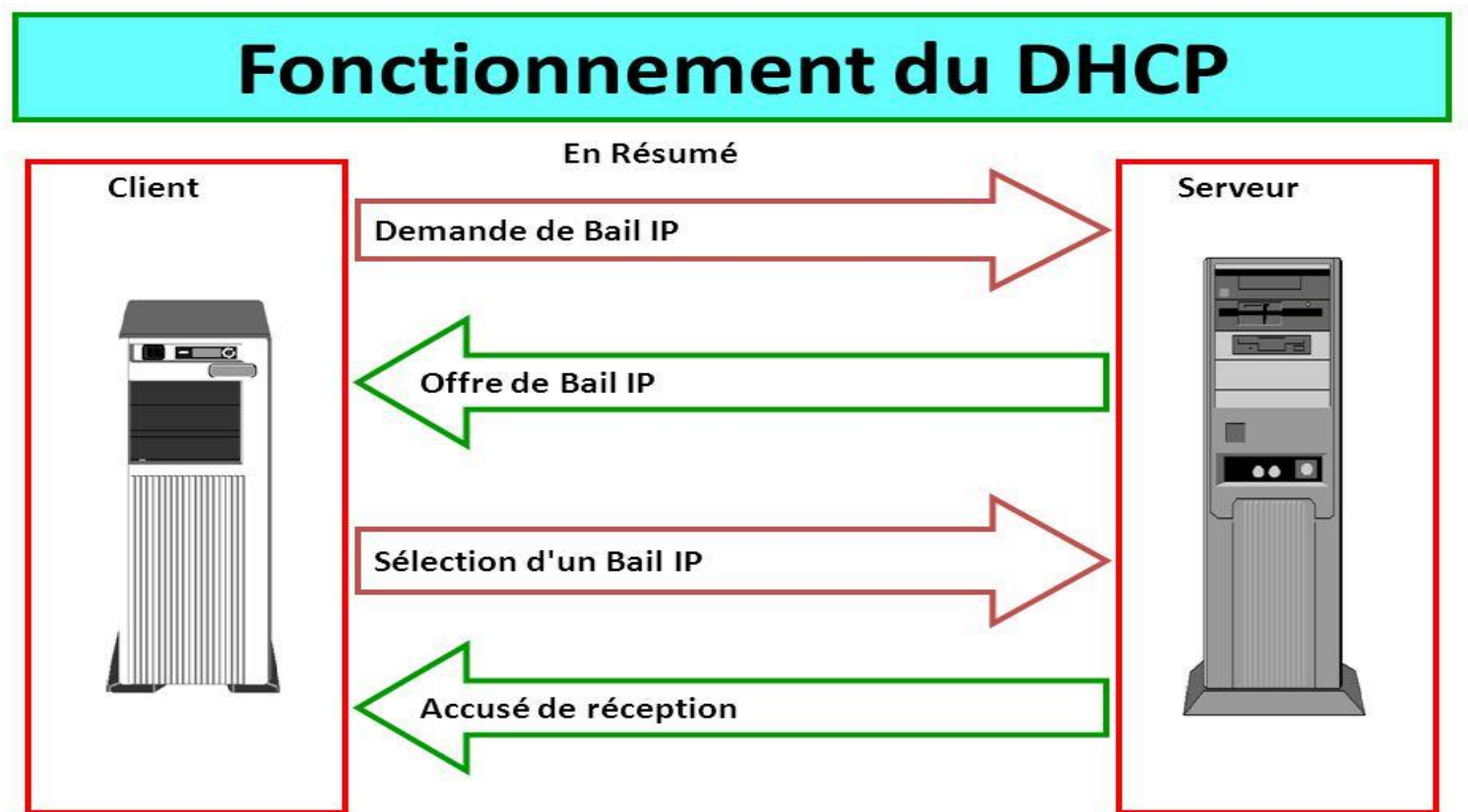
Il existe donc deux méthodes pour obtenir une adresse IP. Soit c'est vous qui la configurez (méthode manuelle), soit c'est un serveur qui vous la donne (méthode dynamique).

La méthode manuelle pose quelques problèmes de prime abord. En effet, on sait que pour qu'une machine puisse communiquer avec ses voisines, son adresse IP devait se trouver dans le même réseau que les autres machines. Pour sortir du réseau local, il faut que notre machine connaisse l'adresse de la passerelle. Cela fait déjà quelques informations dont il faut avoir connaissance lorsque l'on branche un ordinateur à un réseau local. De plus il faut savoir et être sûr qu'aucune autre machine sur le réseau utilise la même IP.

On se rend donc vite compte qu'il serait avantageux et bénéfique de disposer d'un système qui fasse tout cela automatiquement. C'est là que le protocole DHCP entre en jeu.

La première fonction d'un serveur DHCP est de fournir une adresse IP aux machines qui en font la demande.

Fonctionnement du serveur DHCP :



Le DHCP fonctionne en mode client serveur

26

Etape 1 : Préparation du serveur

Afin de disposer d'un environnement de travail propre, il est nécessaire de configurer notre machine serveur. Il est important que le serveur ait une adresse IP statique. Donc avant de commencer nous avons modifier les paramètres réseaux de notre serveur :

Pour cela nous avons éditer le fichier de configuration suivant :

```
/etc/network/interfaces
```

Et nous y avons renseigner les paramètres suivants :

```
#on précise l'interface
auto ens34
#on met l'interface en mode statique
iface ens34 inet static

#on définit l'adresse IP de notre serveur
    address 192.168.11.100
#le masque
    netmask 255.255.255.0
#le réseau
    network 192.168.11.0
#l'adresse broadcast
    broadcast 192.168.11.255
#la passerelle
    gateway 192.168.11.5
```

Et on redémarre le service :

```
systemctl restart networking.service
```

Etape 2 : Installation du serveur DHCP

Une fois que tout est prêt nous passons à l'installation des paquet DHCP.

Pour mettre en place le service DHCP dans notre réseau, nous allons utiliser le paquet isc-dhcp-server. Commençons par l'installer :

```
apt install isc-dhcp-server -y
```

Ensuite, nous devons préciser sur quelle interface du serveur, le “démon” (le “service”) va écouter et donc attendre les requêtes des clients. Modifiez le fichier nécessaire avec la commande suivante :

```
nano /etc/default/isc-dhcp-server
```

Nous n'utilisons pas d'IPv6, la ligne peut donc être commentée. En revanche, pour l'interface en IPv4, il vous faudra ajouter le nom de l'interface réseau de votre serveur entre les guillemets :

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP
requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens34"
#INTERFACESv6=""
```

Ensuite, il faut éditer le fichier **dhcpd.conf** pour configurer le service DHCP :

```
nano /etc/dhcp/dhcpd.conf
```

Il faut être vigilants à la syntaxe employée dans ce fichier, une erreur toute bête peut empêcher le service de démarrer et faire perdre du temps. Chaque ligne de paramètre doit se terminer par le symbole “;”. Veillez également à bien ouvrir/fermer les blocs de paramètres pour les étendues avec les symboles “{ }”.

Dans un premier temps :

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# option definitions common to all supported networks...
#option domain-name "example.org";
option domain-name-servers 8.8.8.8;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;
```

Le premier paramètre est le “domain-name”. Comme son nom l’indique, on déclare ici le nom de notre domaine. Nous n’en avons pas, ce paramètre est donc à commenter.

Ensuite nous avons l’option “domain-name-servers”. Si les serveurs DNS sont les mêmes pour toutes les étendues, on peut déclarer cette option à partir de ce point. Pour le DNS, j’ai choisi tout bêtement celui de Google, le 8.8.8.8. Peut importe les étendues, ce sera le même DNS partout, sinon, l’option peut être précisée dans chaque déclaration d’étendue. C’est donc ici une option dite “de serveur” et non pas “d’étendue”.

Les paramètres “default-lease-time” et “max-lease-time” sont les durées des baux pour les adresses attribuées avant une libération ou un renouvellement. Par défaut ces durées sont comprises entre 3600 secondes (1 heure) et 7200 secondes (2 heures).

L'option "ddns-update-style" sert à définir le type de mise à jour du DNS. Nous n'en n'avons pas besoin dans notre contexte. Par défaut, ce paramètre est défini sur "none".

Le paramètre "authoritative" peut être décommenté si ce serveur DHCP est le serveur officiel du réseau local, ce qui sera le cas ici.

L'option "log-facility" est le niveau de log à conserver. Par défaut, vous trouverez les logs du DHCP dans /var/log/syslog.

Ensuite nous déclarons les réseaux a servir, donc nous rajoutons les ligne suivante a la fin de notre fichier :

```
subnet 192.168.11.0 netmask 255.255.255.0{
}

#VLAN10
subnet 192.168.10.0 netmask 255.255.255.0 {

    #Broadcast address
    option broadcast-address 192.168.10.255;
    #Gateway
    option routers 192.168.10.5;
    #Range IP address to leased
    range 192.168.10.50 192.168.10.100;

}

#VLAN20
subnet 192.168.20.0 netmask 255.255.255.0 {

    #Broadcast address
    option broadcast-address 192.168.20.255;
    #Gateway
    option routers 192.168.20.5;
    #Range IP address to leased
    range 192.168.20.50 192.168.20.100;

}

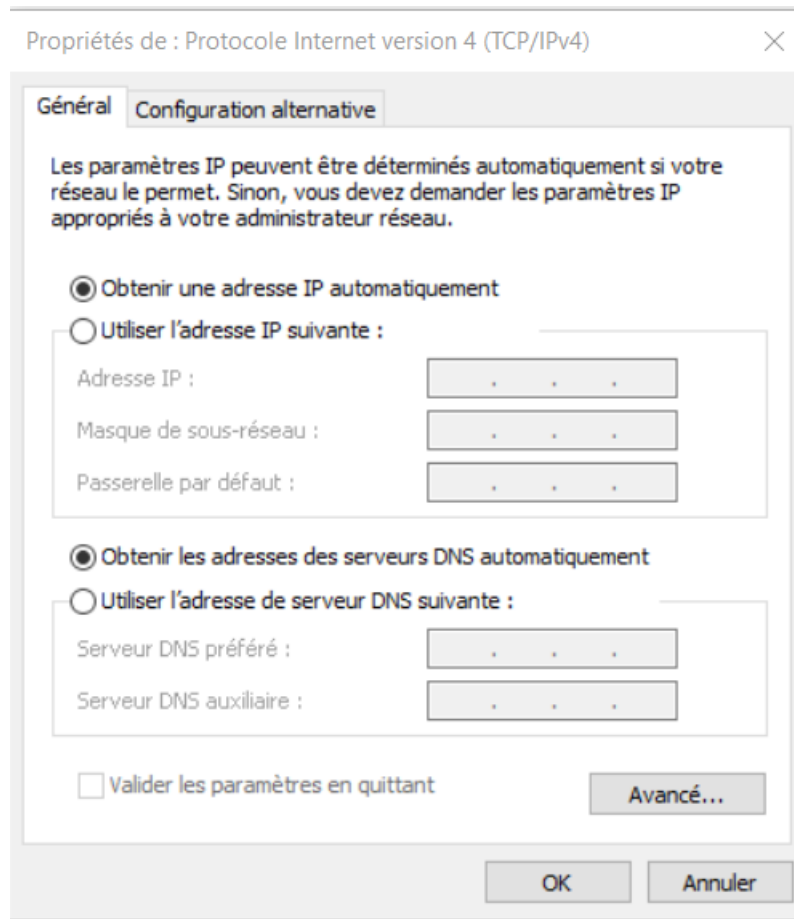
#VLAN30
subnet 192.168.30.0 netmask 255.255.255.0 {

    #Broadcast address
    option broadcast-address 192.168.30.255;
    #Gateway
    option routers 192.168.30.5;
    #Range IP address to leased
    range 192.168.30.50 192.168.30.100;

}
```

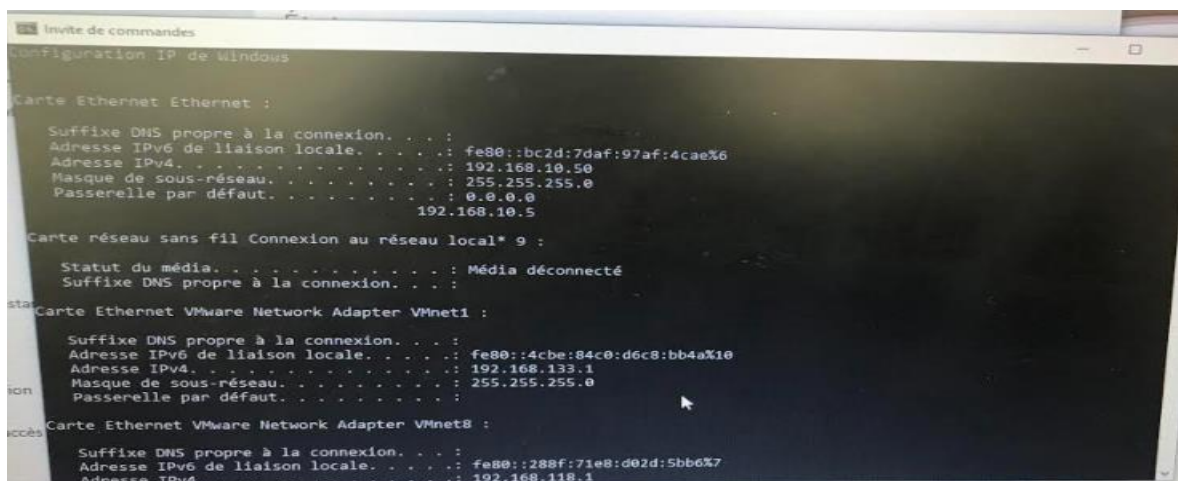

Etape 3 : Test du serveur

Pour tester la fonctionnalité de notre serveur, nous avons définis « Obtenir une adresse IP automatiquement » dans les paramètres réseaux des PC client :



Puis nous avons vérifié grâce à la commande « ipconfig » que les postes reçoivent bien les bonne IP selon le VLAN dans lequel ils sont :

Poste VLAN 10 :



Poste VLAN 20 :

```
Carte Ethernet Ethernet :
Suffixe DNS propre à la connexion. . . :
Adresse IPv6 de liaison locale. . . . : fe80::b115:4dc8:2a60:c233%12
Adresse IPv4. . . . . : 192.168.20.50
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.20.5
```

Poste VLAN 30 :

```
Carte Ethernet Ethernet :
Suffixe DNS propre à la connexion. . . :
Adresse IPv6 de liaison locale. . . . : fe80::148e:a7d1:a261:5b48%12
Adresse IPv4. . . . . : 192.168.30.50
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.30.5
```

On remarque donc que chaque poste reçoit bien une IP.

Nous avons ensuite vérifié que les postes communiquent entre eux via des tests de PING :

Test depuis le poste en VLAN10 vers les VLAN20 et VLAN30 :

```
C:\>ping 192.168.20.50

Pinging 192.168.20.50 with 32 bytes of data:

Reply from 192.168.20.50: bytes=32 time<1ms TTL=126
Reply from 192.168.20.50: bytes=32 time<1ms TTL=126
Reply from 192.168.20.50: bytes=32 time<1ms TTL=126
Reply from 192.168.20.50: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.20.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 192.168.30.50

Pinging 192.168.30.50 with 32 bytes of data:

Request timed out.
Reply from 192.168.30.50: bytes=32 time<1ms TTL=126
Reply from 192.168.30.50: bytes=32 time<1ms TTL=126
Reply from 192.168.30.50: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.30.50:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Test depuis le poste en VLAN20 vers les VLAN10 et VLAN30 :

```
C:\>ping 192.168.10.50

Pinging 192.168.10.50 with 32 bytes of data:

Reply from 192.168.10.50: bytes=32 time<1ms TTL=126
Reply from 192.168.10.50: bytes=32 time=1ms TTL=126
Reply from 192.168.10.50: bytes=32 time=3ms TTL=126
Reply from 192.168.10.50: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.10.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 3ms, Average = 1ms

C:\>ping 192.168.30.50

Pinging 192.168.30.50 with 32 bytes of data:

Reply from 192.168.30.50: bytes=32 time<1ms TTL=127
Reply from 192.168.30.50: bytes=32 time<1ms TTL=127
Reply from 192.168.30.50: bytes=32 time<1ms TTL=127
Reply from 192.168.30.50: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.30.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Test depuis le poste en VLAN30 vers les VLAN10 et VLAN20 :

```
C:\>ping 192.168.10.50

Pinging 192.168.10.50 with 32 bytes of data:

Reply from 192.168.10.50: bytes=32 time<1ms TTL=126
Reply from 192.168.10.50: bytes=32 time=1ms TTL=126
Reply from 192.168.10.50: bytes=32 time<1ms TTL=126
Reply from 192.168.10.50: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.10.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 192.168.20.50

Pinging 192.168.20.50 with 32 bytes of data:

Reply from 192.168.20.50: bytes=32 time<1ms TTL=127
Reply from 192.168.20.50: bytes=32 time=1ms TTL=127
Reply from 192.168.20.50: bytes=32 time<1ms TTL=127
Reply from 192.168.20.50: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.20.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```