

# Saé21 : Construire un réseau informatique pour une petite structure

## Présentation

**SAé** est une **Situation d'Apprentissage** et d'évaluation.

Cette SAé a débuté le lundi 2 mai 2022 vers 9h10 à l'IUT Réseaux et Télécommunications de Montbéliard et s'est terminée le mercredi 4 mai 2022 vers 12h30.

Le projet consistait en la **construction d'un réseau informatique pour une petite entreprise**. Monsieur Givron et Monsieur Bouillet étaient à notre disposition afin de nous aiguiller et nous guider lorsque l'on rencontrait un problème.

Au total nous avons une vingtaine d'heure afin de réaliser cette SAé ainsi qu'un créneau horaire d'un peu moins d'une heure et demie le lundi matin afin de nous expliquer comment allait se passer ce début de semaine.

Nous disposons de **matériel spécifique** afin de réaliser notre réseau :

- Plusieurs **ordinateurs** sous Debian équipés d'écrans, claviers, souris et connectés avec des câbles Ethernet au réseau de l'IUT afin de se connecter à notre session et de configurer les différents équipements réseaux et serveurs (sauf un ordinateur qui est connecté au switch) ;
- D'un **routeur** et d'un **switch** de la marque Cisco avec leurs alimentations et des câbles Ethernet ;
- D'un **Raspberry Pi 4** de 2GB de RAM avec son câble d'alimentation, un câble HDMI, une carte SD, un clavier et une souris. Un écran pour le Raspberry Pi et son alimentation ;
- Un serveur Proxmox par groupe.

Les **objectifs** de cette Saé sont de nous former concrètement aux déploiements et aux configurations réseau tel que nous serions amenés à le faire dans un **contexte professionnel**.

Nous avons manipulé dans un **environnement PXE** qui est un environnement dans lequel un ordinateur dit client PXE va chercher son chargeur de démarrage, non plus sur l'un de ses périphériques matériels mais sur le réseau, dans notre cas chez le serveur PXE.

Cet environnement nécessite la configuration préalable de plusieurs parties (serveurs DHCP, TFTP et NFS).

La **problématique** principale est la mise en réseau des différents périphériques et services tout en gardant une topologie logique pour une petite entreprise. La bonne configuration des équipements actifs tel que routeur et switch sera alors primordiale tandis que nous nous occuperons de la configuration des serveurs énoncés précédemment et des différents clients présents sur le réseau.

Dans ce rapport, nous allons vous présenter ce que nous avons retenu de ce projet en suivant le plan suivant :

## Sommaire

<b>Préparation et organisation .....</b>	<b>1</b>
<b>Adressage et Topologie .....</b>	<b>5</b>
<b>Serveur DHCP/BootP .....</b>	Erreur ! Signet non défini.
<b>Configuration routeur et relais DHCP .....</b>	Erreur ! Signet non défini.
<b>Serveur TFTP .....</b>	Erreur ! Signet non défini.
<b>Partage des fichiers NFS.....</b>	Erreur ! Signet non défini.
<b>Serveur SSH .....</b>	Erreur ! Signet non défini.
<b>Sécurisation des équipements .....</b>	Erreur ! Signet non défini.

# 1 Préparation et organisation

Notre groupe s’est formé lundi matin, nous avons donc commencé à parler à ce moment de notre organisation au courant de cette SAé. Nous en avons donc déduit qu’un moyen simple, partagé entre nous, dans lequel l’état des jalons ainsi que le travail qui resterait à faire y seraient mentionnés :

- L’outil appelé **Notion** nous est alors parût une évidence, avec son interface simple et complète. L’un de nous s’est alors porté volontaire pour prendre une partie de la matinée afin de finaliser l’outil et partager le lien entre les membres du groupe. Notre planning ressemble à :

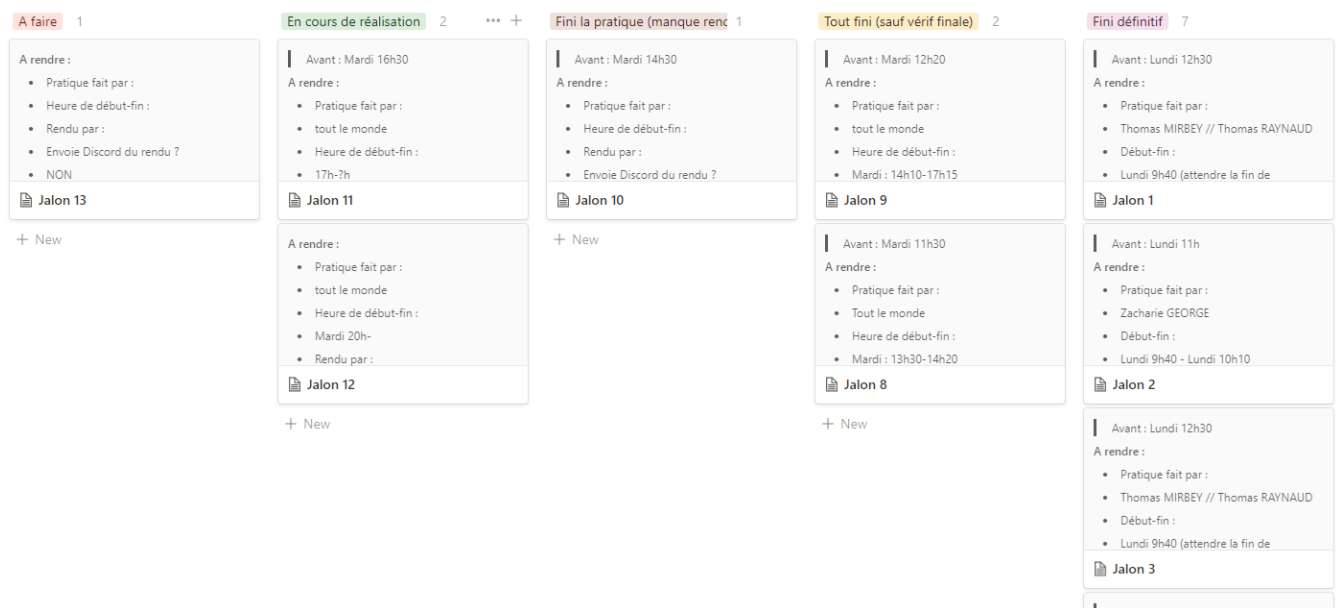


Figure 1 : Interface du document Notion

## Jalon 1

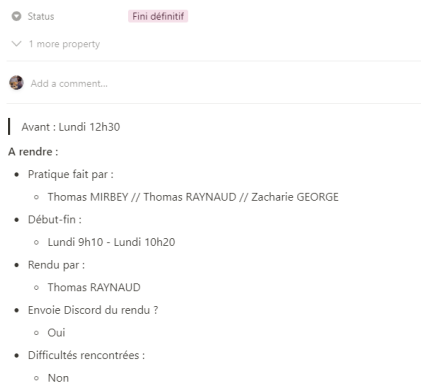


Figure 2 : Exemple de contenu d'un jalon



### Lien de la page Notion :

<https://spicy-hibiscus-e4d.notion.site/713e378a710043e38e0f3e1b99ec29ea?v=91d11b1d7e5d4c70be6a99e4292257da>

- Nous avons aussi créé un serveur discord dans lequel nous déposons nos documents relatifs à chaque jalon pour ne pas se perdre dans des documents qui ne sont pas forcément nommés par exemple, voici les différents salons présents :

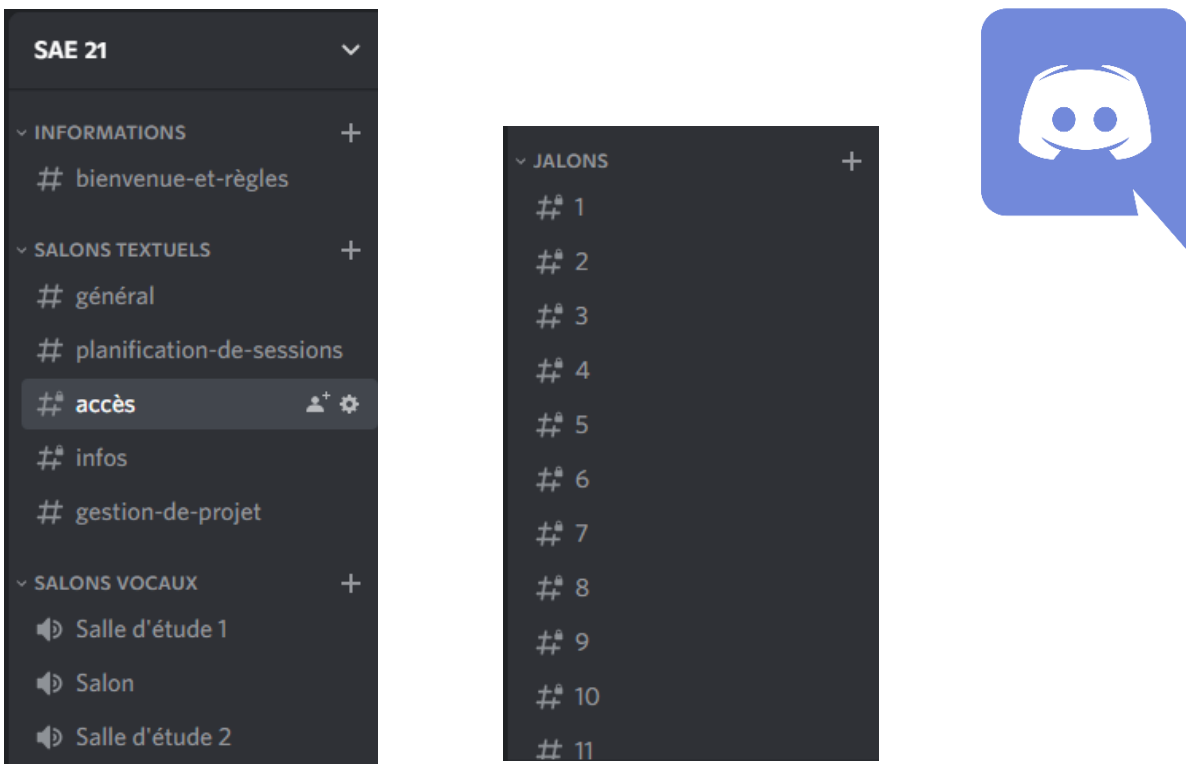


Figure 3 : Interface des différents salons de notre serveur Discord

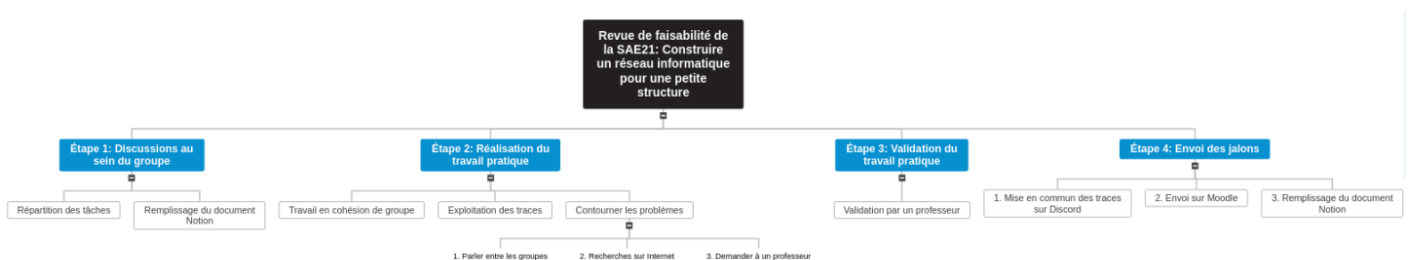


Figure 4 : revue de faisabilité de notre projet

## 2 Adressage et Topologie

Les premiers jalons consistaient en la mise en place d'une première approche de topologie sur logiciel « Cisco Paquet Tracer », pour cela nous devons :

- Découper le réseau 172.16.3.0/26 en quatre sous réseau. Nous avons donc obtenu ce plan d'adressage :

Nom du réseau	Sous réseau	Masque de sous réseau	1ère adresse disponible	Dernière adresse disponible	Adresse de diffusion	Nombre d'hôtes	IP statiques (réservées serveur routeur,...)
Réseau 1	172.16.3.0	255.255.255.240	172.16.3.1	172.16.3.14	172.16.3.15	14	- Serveur PXE : servpxe-3 => 172.16.3.1
							- Routeur : Routeur-3 => 172.16.3.14
Réseau 2	172.16.3.16	255.255.255.240	172.16.3.17	172.16.3.30	172.16.3.31	14	Serveur PXE : servpxe-3 => 172.16.3.17
							Client PXE : clipxe-3 => 172.16.3.26 (en DHCP)
Réseau 3	172.16.3.32	255.255.255.240	172.16.3.33	172.16.3.46	172.16.3.47	14	Routeur : router-3 => 172.16.3.33
							Commutateur - switch-3 => 172.16.3.34
							Client Raspberry : raspi-3 => 172.16.3.42 (en DHCP)
							Client PC : pcsalletp-3 : 172.16.3.43 (en DHCP)
Réseau 4	172.16.3.48	255.255.255.240	172.16.3.49	172.16.3.62	172.16.3.63	14	/ pas utilisé

Figure 5 : Plan d'adressage IP (jalon 1)

- Puis nous avons fait cette première topologie logique sur Lucidchart afin d’avoir un premier aperçu du réseau et de faciliter l’approche avec Cisco Paquet Tracer :

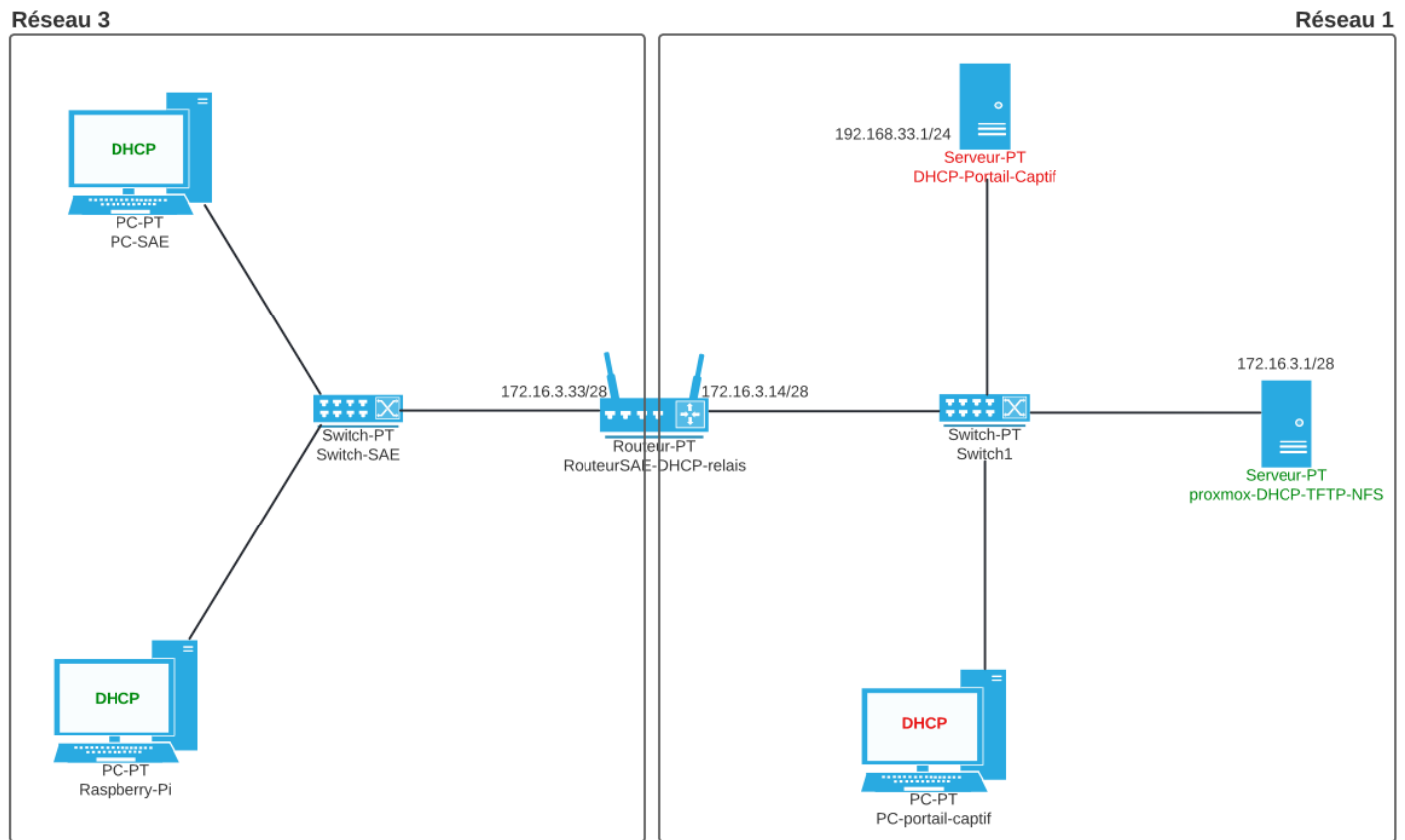
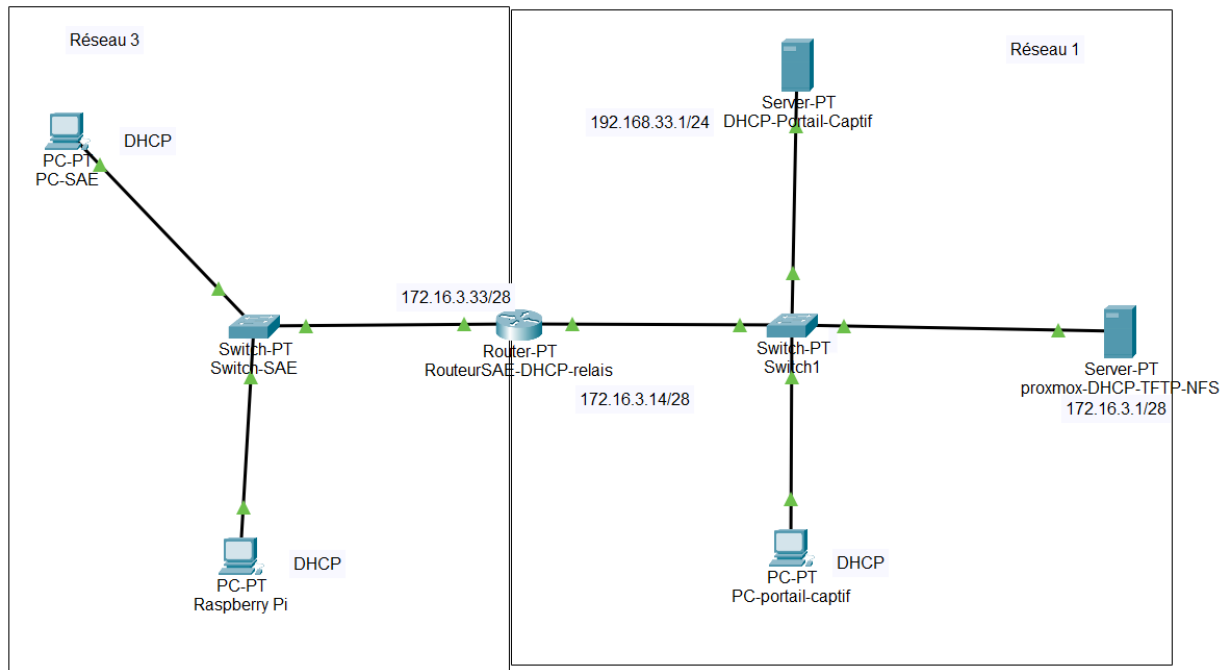


Figure 6 : Topologie logique réalisée sur Lucidchart

- Et enfin en suivant les instructions du document nommé « Simulation de la configuration DHCP » nous avons réaliser la mise en place du réseau sur Cisco Paquet Tracer :



- Lors de la première étape de cette Saé, nous avons donc eu une première approche de notre futur réseau tout en utilisant certaines commandes afin de configurer les équipements actifs. Contrairement à beaucoup de personnes, nous n'avons pas eu beaucoup de problème lors de la configuration paquet tracer, et ce grâce à une organisation contrôlée afin de ne pas aller trop vite et commettre des fautes de frappe. Seulement lors ce que l'on essayait de sauvegarder depuis le logiciel et faire CTRL + S, le PC s'éteignait. Nous avons donc repris à plusieurs reprise la configuration malgré que notre réseau machait.

### 3 Serveur DHCP/ BootP

Dans le réseau que nous voulons créer, trois clients sont présents :

- Le Raspberry-Pi (sur le réseau 3) ;
- Le clipxe-3 (sur proxmox, le réseau 2) ;
- Un ordinateur réel (sur le réseau 3).

Ces clients ne possèdent pas de configuration réseau fixe et doivent donc démarrer en prenant la configuration d'un serveur. Ce serveur est présent dans notre serveur proxmox et se nomme « servpxe-3 ».

Il faut savoir différencier DHCP et BootP. En effet DHCP permet de fournir les paramètres IP aux clients alors que BootP est utilisé pour configurer et démarrer des ordinateurs sans disque.

Nous avons donc commencé la configuration d'un serveur DHCP, pour cela nous avons modifié certains fichiers de configuration,

Un fichier de configuration nous avait particulièrement servi, il se trouve sous : /etc/dhcp/dhcp.conf

En effet ce dernier permet à notre serveur DHCP de distribuer des adresses IP sous une étendue dans les sous réseaux que nous avons créés au préalable :

```
#DECLARATION D'UN SOUS-RESEAU
subnet 172.16.3.32 netmask 255.255.255.240 {

#PLAGE d'IP à délivrer
range 172.16.3.35 172.16.3.44;

#GATEWAY par défaut
option routers 172.16.3.33;

#L'adresse de broadcast du sous-réseau
option broadcast-address 172.16.3.47;

}
```

Figure 7 : Exemple de déclaration pour le réseau 3



Nous pouvons également spécifier si des adresses doivent être réservées pour un hôte en particulier. Par exemple, pour la machine clipxe-3 qui doit avoir une adresse réservée de 172.16.3.26 :

```
host clipxe-3 {
hardware ethernet BE:C3:6E:49:5C:52;
fixed-address 172.16.3.26;
```

Si l'adresse MAC de la machine qui demande une adresse IP est « BE:C3:6E:49:5C:52 », le serveur lui donnera automatiquement celle qui se trouve après le « fixed-address ». Cette adresse est réservée et aucun autre hôte ne pourra la prendre.

1	0.000000000	172.16.3.33	172.16.3.1	DHCP	364 DHCP Discover	- Transaction ID 0xbaea59ec
2	0.000194936	172.16.3.1	172.16.3.33	DHCP	352 DHCP Offer	- Transaction ID 0xbaea59ec [Malformed Packet]
3	0.001789901	172.16.3.33	172.16.3.1	DHCP	351 DHCP Request	- Transaction ID 0xbaea59ec
4	0.002000847	172.16.3.1	172.16.3.33	DHCP	358 DHCP ACK	- Transaction ID 0xbaea59ec [Malformed Packet]
5	5.040211214	172.16.3.33	172.16.3.1	DHCP	364 DHCP Discover	- Transaction ID 0xbb2771c2

Figure 8 : Capture Wireshark entre serveur et raspberry pi

- ⇒ Il faut noter que l'utilisation de la commande « tail -30 /var/log/syslog » nous a beaucoup aidé car de nombreuses erreurs sont apparues lors de la création de notre serveur. C'est pour cela que nous avons essayé d'isoler le problème, d'abord en essayant le routeur, puis le switch et enfin le Raspberry Pi. Ce dernier ne trouvait pas d'adresse IP alors que notre fichier de configuration était sans erreur. Après de nombreux essais nous avons donc découvert avec l'aide de Monsieur Bouillet et Monsieur Givron, que la carte SD du Raspberry Pi avait des problèmes de configuration.

## 4 Configuration routeur et relais DHCP

Afin d'assurer la transmission des paquets au sein du réseau, il fallait que le routeur soit bien configuré. Pour cela, il fallait donner les bonnes adresses IP aux bonnes interfaces.

```
routeur-3#show ip interface brief
Interface                IP-Address      OK? Method Status      Protocol
Embedded-Service-Engine0/0 unassigned      YES NVRAM   administratively down down
GigabitEthernet0/0       172.16.3.33     YES manual  up          up
GigabitEthernet0/1       172.16.3.14     YES manual  up          up
```

Figure 9 : Résultat de la commande "show ip interface brief"

On peut voir que la commande présente dans la figure ci-dessus répertorie comme dans notre topologie logique la bonne association interface – IP.

Cependant, on remarque d'après la figure 4, qu'un routeur sépare le réseau 3 qui contient les clients **demandant** une adresse IP (fonctionnent en DHCP) et le réseau 1 qui contient le serveur.

Les demandes DHCP des hôtes n'atteindront jamais le serveur si une configuration spécifique n'est pas appliquée au routeur, cette configuration est appelée **relais DHCP**. C'est pour cela qui ne fallait pas oublier d'ajouter dans l'interface client du routeur « Gi0/0 » la commande :

- **ip helper-address 172.16.3.1**

Nous avons pu vérifier cette configuration lors de l'application de deux commandes différentes :

- **copy running-config startup-config** -> permet d'enregistrer la configuration en cours dans la configuration initiale.

- show startup-config -> permet de vérifier les configurations de notre routeur en totalité. Cette commande dans des résultats assez approfondie et nous assure de la bonne configuration de notre routeur.
- Cette partie de configuration de routeur a été plus facile pour nous grâce à la formation CCNA1 et CCNA2 de Cisco dans lesquels nous avons appris à manipuler certains équipements actifs de la marque Cisco.

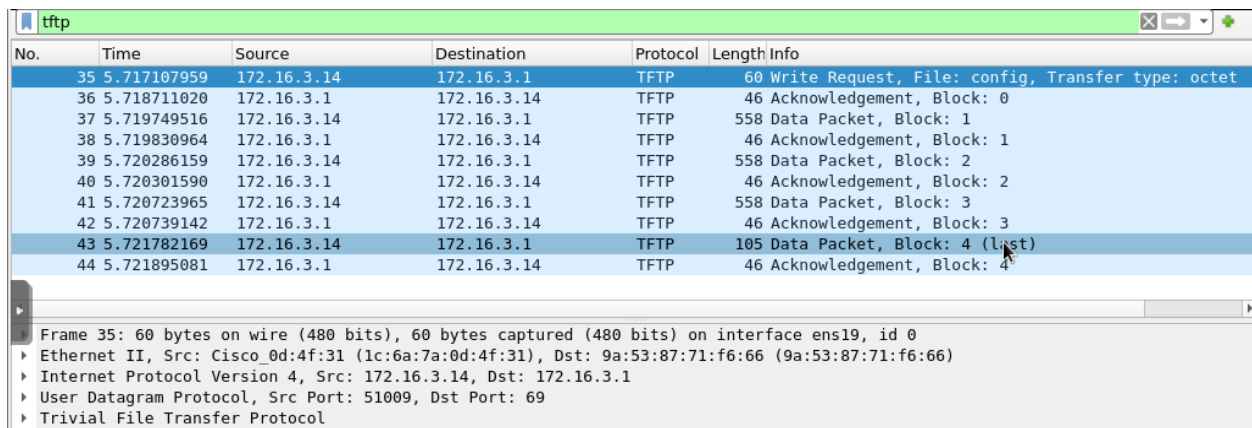
## 5 Serveur TFTP

Un serveur TFTP est un serveur qui est en mesure de distribuer des fichiers hébergés sur sa racine, sur une simple requête sans authentification. C'est un protocole qui passe par UDP et son port est le 69.

Sur notre réseau, le serveur TFTP se trouve sur notre serveur-PXE présent sous notre serveur proxmox. L'activation de ce serveur se fait à nouveau en la modification de plusieurs fichiers de configuration :

- /etc/dhcp/dhcpd.conf => Permet par l'activation de BootP, et permet en spécifiant l'IP du serveur quel fichier de configuration chercher. Il se nomme « pxelinux.0 ». Il faut noter que le Raspberry Pi a des spécifications vis-à-vis des lignes qu'il faut ajouter.
- /etc/default/tftpd-hpa => fichier qui spécifie sur quels IP doit écouter le serveur et certains autres paramètres.

Les connexions du client au serveur TFTP peut se vérifier en utilisant Wireshark. Le résultat que nous avons obtenu est le suivant :



No.	Time	Source	Destination	Protocol	Length	Info
35	5.717107959	172.16.3.14	172.16.3.1	TFTP	60	Write Request, File: config, Transfer type: octet
36	5.718711020	172.16.3.1	172.16.3.14	TFTP	46	Acknowledgement, Block: 0
37	5.719749516	172.16.3.14	172.16.3.1	TFTP	558	Data Packet, Block: 1
38	5.719830964	172.16.3.1	172.16.3.14	TFTP	46	Acknowledgement, Block: 1
39	5.720286159	172.16.3.14	172.16.3.1	TFTP	558	Data Packet, Block: 2
40	5.720301590	172.16.3.1	172.16.3.14	TFTP	46	Acknowledgement, Block: 2
41	5.720723965	172.16.3.14	172.16.3.1	TFTP	558	Data Packet, Block: 3
42	5.720739142	172.16.3.1	172.16.3.14	TFTP	46	Acknowledgement, Block: 3
43	5.721782169	172.16.3.14	172.16.3.1	TFTP	105	Data Packet, Block: 4 (last)
44	5.721895081	172.16.3.1	172.16.3.14	TFTP	46	Acknowledgement, Block: 4

Frame 35: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface ens19, id 0  
▶ Ethernet II, Src: Cisco\_0d:4f:31 (1c:6a:7a:0d:4f:31), Dst: 9a:53:87:71:f6:66 (9a:53:87:71:f6:66)  
▶ Internet Protocol Version 4, Src: 172.16.3.14, Dst: 172.16.3.1  
▶ User Datagram Protocol, Src Port: 51009, Dst Port: 69  
▶ Trivial File Transfer Protocol

Nous avons aussi réalisé une sauvegarde des configurations du routeur via TFTP. Le fichier s'appelle **config**. Il est très utile pour des administrateurs réseau, car si la configuration d'un routeur est perdue, ce dernier peut toujours redémarrer grâce à ces sauvegardes.

- Sur cette partie, les problèmes rencontrés ont été moindres. En effet elle était plutôt simple et nous avons déjà réalisé au préalable lors de la ressource **Technologie de l'Internet** des sauvegardes de routeurs via tftp.

## 6 Partages de fichiers NFS

Un **serveur NFS** est par définition un serveur qui permet à des utilisateurs de consulter, stocker et mettre à jour des fichiers sur un ordinateur distant.

Il fallait au début installer des paquets. C'est pour cela que l'utilisation d'un proxy était obligatoire. Nous nous sommes rendu compte d'un conflit entre la carte réseau NAT et la carte réseau « ens19 » nous avons donc décidé de désactiver cette dernière et de réaliser un dhclient sur la carte NAT afin de renouveler la configuration réseau.

La copie et la manipulation de différents fichiers et dossiers étant simple dans la première partie, nous n'avons pas éprouvé de problèmes en particulier. La Machine Client-PXE démarrait donc sur l'image **netinstall** de linux.

Une configuration spécifique est requise pour le Raspberry Pi. Le démarrage du Raspberry Pi sur le réseau ne se fait que sous certaines conditions, il faut que ses partitions boot et root sont situées dans un répertoire correspondant à son numéro de série.

Il fallait donc utiliser tout au long de la configuration utiliser le numéro de série de notre Raspberry Pi que nous avons trouvé dans le jalon 2, il fallait prendre les 8 derniers chiffres.

La seconde partie de ce travail nous a plus posé des problèmes. Il fallait se connecter en sftp sur rt-serv, et importer les deux partitions boot et root du raspbian lite avec la commande **get**. La décompression et la manipulation de ces différentes images nous ont portés problème.

- La première fois, nous avons laissé le « \$ » avant le numéro de série du Raspberry Pi mais à la fin de la configuration nous nous sommes rendu compte d'une erreur « panic ». Après avoir parlé avec différents groupes, nous avons pu déterminer notre erreur. Nous avons donc recommencé la manipulation.
- La deuxième fois, que nous avons essayé, nous avons fait des erreurs de frappes et d'inattention dû à la limite de temps. Des lignes et certains fichiers de configurations n'étaient soit pas bien copiés au bon endroit, soit des erreurs de frappes. Notamment dans le fichier **cmdline.txt** qui demande une syntaxe irréprochable.
- La troisième fois était la bonne. Cette fois-ci nous n'avons pas utilisé de commandes mais avons préférés passer par l'interface graphique. Les seuls éléments que nous avons fait par les lignes de commandes sont les modifications de fichiers de configuration.

## 7 Serveur SSH

Le rôle du protocole SSH est d'établir une connexion sécurisée vers une machine distante.

Dans un premier temps, nous avons suivi les instructions afin de configurer le SSH pour le serveur, nous avons pu :

- Interdire la connexion de l'utilisateur root ;
- Mettre en place une bannière MOTD ;
- Autoriser uniquement les accès depuis le réseau 1.

Puis nous avons mis en place différents VLANs sur le switch. Les VLANs sont des réseaux locaux virtuels :

- VLAN 10 : destiné au réseau 3 ;
- VLAN 20 : destiné au réseau 4 ;
- VLAN 99 : vlan de gestion.

Ces VLANs permettent alors une meilleure sécurisation du réseau et avoir des réseaux indépendants à chaque service dans le réseau.

L'organisation des VLANs est réalisée et visible ci-dessous :

VLAN	Name	Status	Ports
1	default	active	Gil/0/1, Gil/0/2
10	Reseau3	active	Fal/0/1, Fal/0/2, Fal/0/3
20	Reseau4	active	Fal/0/4, Fal/0/5, Fal/0/6 Fal/0/7, Fal/0/8, Fal/0/9 Fal/0/10, Fal/0/11, Fal/0/12 Fal/0/13, Fal/0/14, Fal/0/15 Fal/0/16, Fal/0/17, Fal/0/18 Fal/0/19, Fal/0/20, Fal/0/21 Fal/0/22, Fal/0/23, Fal/0/24

Et enfin, la configuration du SSH.

- Le problème principal que nous avons rencontré était que la ligne « KeyAlgorithms +diffie-hellman-group1-sha1 » ne permettait pas de se connecter en SSH entre les équipements. Nous avons alors remplacé cette ligne par :

```
# Ciphers and keying
#RekeyLimit default none

KexAlgorithms +diffie-hellman-group1-sha1
Ciphers aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
```

- Elle est fonctionnelle et l'on peut voir la bannière :

```
switch-3#ssh -l admin 172.16.3.14
Password:
Password:
.....#####.....#####
##.....##.....##.....##.....
##.....##.....##.....##.....
.....#####.....#####
.....#####.....#####
##.....##.....##.....##.....
.....#####.....#####
.....#####.....#####
routeur-3#
```

Figure 10 : Connexion en SSH du switch vers le routeur

## 8 Sécurisation des équipements

Afin de sécuriser le switch, nous avons :

- Eteint les ports non utilisés ;
  - Activé la sécurité des ports par limitation MAC « switchport port-security mac-address sticky » et défini une action à réaliser en cas de violation « switchport port-security violation shutdown »;
  - Activé le DHCP Spoofing afin d'empêcher un serveur DHCP pirate de donner des IP sur le réseau.
- ⇒ Cette partie ne nous a pas particulièrement posée de problème en vu de la courte partie de TP à réaliser.

## Conclusion

Cette [Situation d'Apprentissage et d'évaluation](#) nous a apporté de nombreuses compétences dans la configuration de réseaux. La manipulation de matériel actif et de serveurs. Cette Saé permet de valider les acquis de cette année de travail.

Nos [succès](#) principaux étaient :

- Une **organisation au sein du groupe** avec le document notion et le serveur discord.
- Une bonne **répartition du travail équitable** au sein de notre groupe. Cela nous a permis de travailler efficacement en équipe et de partager nos connaissances, ce qui est primordial en entreprise.

Certains problèmes nous ont empêché d'avancer aussi vite que l'on le souhaitait, c'est pour cela qu'il a fallu faire preuve de flexibilité et de coordination dans le groupe.

Nous avons rencontré une [situation imprévue](#) lors du dernier jour et la création des diaporamas car nous avons oublié de garder certaines traces lors de notre pratique. Nous avons alors contourné le problème en remanipulant certains équipements afin de reprendre certaines photos de nos travaux.

Vis-à-vis du [respect de la méthodologie](#), nous pensons avoir réalisé du bon travail, organisé, **sans sauter d'étapes** en voulant aller trop vite.

Cette SAé nous aura permis de **travailler efficacement en équipe**. Nous avons également dû chercher comment **contourner des problèmes** en cherchant des informations par nous-même et en travaillant avec d'autres groupes. Ceci nous permet de capitaliser des connaissances qui pourrons être utilisées dans les [projets à venir ainsi que dans notre vie professionnelle](#).