# Réalisation d'un modèle d'architecture de réseau open source, simple et peu coûteux



Etudiants : Matteo BENNIS & Olivier PORCO

Tuteur : Faissal BAKALI

Année 2022-2023

# Remerciement

Nous voudrions sincèrement remercier M. Faissal Bakali, M. Philippe Egea et M Alexandre Guy pour leur soutien et leurs conseils pendant la réalisation du projet.

Grâce à eux, nous avons pu travailler au mieux des possibilités, avec des matériaux de qualité et avec un retour critique indispensable pour la réalisation de travaux de ce niveau.

Enfin, grâce aux connaissances qu’ils nous ont transmises, nous avons développé un produit également adapté au monde professionnel et académique.

Un remerciement spécial au Professeur Faissal Bakali pour son soutien jusqu’aux phases finales du projet et la rédaction de ce rapport.

Nous sommes convaincus que notre travail serait beaucoup plus difficile sans votre aide.

Sincèrement,

Matteo BENNIS & Olivier PORCO.

# Sommaire

Sommaire

[Sommaire 3](#_Toc129957084)

[Introduction 5](#_Toc129957085)

[Problématiques 6](#_Toc129957086)

[Cahier des charges 7](#_Toc129957087)

[Vocabulaire 8](#_Toc129957088)

[Solution propose 11](#_Toc129957089)

[Réflexion 11](#_Toc129957090)

[Schéma de la solution 12](#_Toc129957091)

[Répartition des tâches 13](#_Toc129957092)

[Routage et VLANs 14](#_Toc129957093)

[Matériels 14](#_Toc129957094)

[Commutateur et VLANs 14](#_Toc129957095)

[Difficultés rencontrées 14](#_Toc129957096)

[Nos VLAN 15](#_Toc129957097)

[Routage 16](#_Toc129957098)

[Difficultés rencontrées 16](#_Toc129957099)

[VLAN 99 19](#_Toc129957100)

[Serveur de fichier 20](#_Toc129957101)

[OpenMediaVault 20](#_Toc129957102)

[Configuration d’OpenMediaVault 21](#_Toc129957103)

[Serveur DHCP 22](#_Toc129957104)

[Présentation de DNSmasq 22](#_Toc129957105)

[Configuration 22](#_Toc129957106)

[Serveur DNS 23](#_Toc129957107)

[DNS 23](#_Toc129957108)

[Problèmes rencontrés 23](#_Toc129957109)

[Serveur VPN 24](#_Toc129957110)

[Problémes rencontrés 24](#_Toc129957111)

[Webographie 26](#_Toc129957112)

[Annexes 27](#_Toc129957113)

[Diagramme de Gantt 27](#_Toc129957115)

[Devis 28](#_Toc129957116)

[Documentation 29](#_Toc129957117)

# Introduction

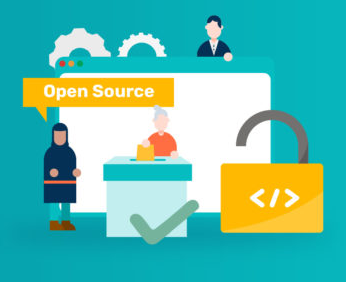
La mise en réseau est un élément clé pour les entreprises et les particuliers, car elle permet de connecter les personnes, les appareils et les objets connectés pour faciliter la communication, la collaboration et l'accès aux données.

Fig : [Source](https://opensourcepolitics.eu/wp-content/uploads/2021/08/prr-01-scaled.jpg)

Ce genre d’installations, peuvent s'avérer coûteuses, surtout pour les petites entreprises et les particuliers mais de plus en plus indispensable, car les cyberattaques sont devenues beaucoup plus fréquentes.

Ce projet tutoré vise à résoudre ces problèmes en proposant un modèle d'architecture de réseau open source1, simple et peu coûteux, qui peut s'adapter aux besoins spécifiques des petites entreprises et des particuliers.

Rapide à mettre en place, à configurée, évolutif et adaptable aux différents besoins en intégrant des fonctionnalités pour garantir la sécurité des données.

Dans ce rapport, nous allons détailler les besoins et les objectifs de ce projet, les caractéristiques techniques, les livrables attendus, les délais et le budget estimé.

1OpenSorce : L’Open Source est un modèle de développement logiciel et matériel qui permet à quiconque d’accéder, de modifier et de distribuer librement le produit sans restriction ou coûts de propriété intellectuelle. Ce modèle encourage la collaboration et la transparence pour aboutir à des solutions plus novatrices, efficaces et rentables.

# Problématiques

**Pour les petites entreprises :**

* Offrir un modèle d'architecture de réseau open source, simple et peu coûteux, qui peut s'adapter aux besoins spécifiques.
* Proposer des solutions pour améliorer la connectivité des employés en présentiel et en distanciel, faciliter la collaboration et la communication, et protéger les données sensibles.
* Intégration des fonctionnalités de cybersécurité pour protéger les données contre les attaques.

**Pour les particuliers :**

* Permette aux particuliers de mieux contrôler leur réseau domestique en leur offrant plus de possibilités de personnalisation dans le routage et de profiter d'outils normalement réservés aux entreprises comme un NAS et/ou un serveur VPN pour les sauvegardes sécurisées à distance.
* Intégration des fonctionnalités de domotique pour connecter et gérer les IoT (Internet des objets) à travers le réseau domestique.
* Intégration des fonctionnalités de cybersécurité pour protéger la confidentialité numérique.

# Cahier des charges

**Caractéristiques techniques :**

**Router et Serveurs :**

Nous avons utilisé des Raspberry Pi 4B, un ordinateur à carte unique, avec les spécifications suivantes :

* CPU : ARM Cortex-A72 Quad-core 1,5 GHz – 2GHz
* RAM : 4 Go ddr4
* Connectivité Ethernet
* Connectivité Wi-Fi/Bluetooth.

Il peut être utilisé pour différentes applications, notamment :

* Serveur
* Projets IoT
* Robotique
* Routage.

Il se distingue par son bas cout, sa polyvalence, sa flexibilité et la facilité d’utilisation.

**Switch :**

Nous avons utilisé un commutateur programmable DGS-1210-10 D-Link avec 8 ports RJ45 1Go/s et 2 ports SFP

**Livrables :**

* Modèle d'architecture de réseau open source complet
* Documentation technique
* Manuel d'utilisation pour les utilisateurs

**Budget :**

- Coût total estimé de 600 euros

**Responsabilités :**

Administrateur Réseau : responsables de la mise en place et de la configuration du modèle d'architecture de réseau,

Client : Responsable de la maintenance et de la mise à jour.

# Vocabulaire

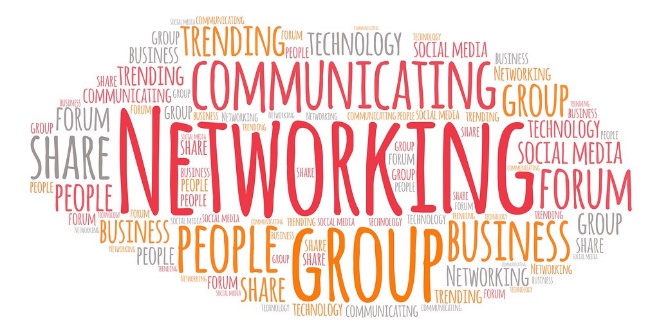


Fig ii: [Source](https://live.staticflickr.com/65535/48282676622_6629a3c0be_b.jpg)

Avant de présenter le projet, il y a quelques termes importants à connaître :

1. Interface : Chaque port qui permet l’interconnexion entre des périphériques est appelé interface.
   1. Sous-interfaces : permettent de diviser une interface physique en plusieurs sous-interfaces logiques, chacune ayant ses propres paramètres de configuration réseau. (Généralement utilisées pour le système router-on-a-stick)
2. SFP : Les ports SFP (Small Form-factor Pluggable) sont un type d’interface réseau utilisé pour connecter des périphériques réseau tels que des commutateurs, des routeurs, des serveurs et des périphériques de stockage.
3. Paquet : Chaque fois qu’une périphérique A, envoie une donnée à une autre B, il encapsule les données et toutes les informations utiles pour le transporter.
4. Switch/Commutateur : un dispositif qui permet l’interconnexion sécurisée entre plusieurs périphériques via une interface généralement Ethernet ou fibre.
5. Virtual Lans (VLANs) : Chaque commutateur paramétrable dispose d’un système de gestion de LAN virtuel par lequel, à l’aide de tag et untag des paquets, permet de le diviser en réseaux séparés et distincts ; brièvement, les LAN virtuels se trouvent physiquement dans le même switch, mais sont complètement séparés, ce qui implique qu’ils ont besoin d’un routeur pour communiquer entre eux.
   1. Tag / Untag : Assignation/suppression de numéro d’identification du vlan sur le paquet
6. DHCP : DHCP signifie "Dynamic Host Configuration Protocol", ce qui se traduit par protocole de configuration dynamique des hôtes. C'est un protocole réseau qui permet aux ordinateurs et autres périphériques de se connecter à un réseau et d'obtenir automatiquement une adresse IP et d'autres paramètres réseau tels que la passerelle par défaut, le masque de sous-réseau et les serveurs DNS.
7. DNS : DNS signifie "Domain Name Server", ce qui se traduit par serveur de noms de domaine. C'est un serveur qui permet de traduire les noms de domaine (FQDN) en adresses IP, qui sont utilisées pour identifier les ordinateurs et les ressources sur un réseau. Plutôt que de mémoriser des adresses IP numériques, les utilisateurs peuvent accéder à des sites web en utilisant des noms de domaine plus faciles à retenir, tels que www.example.com.
8. VPN : VPN signifie "Virtual Private Network", ce qui se traduit par réseau privé virtuel. C'est un type de réseau sécurisé qui permet à des ordinateurs ou des réseaux distants de se connecter entre eux via Internet de manière sécurisée.
9. Cryptage asymétrique à 256 bits : Le cryptage asymétrique (ou cryptage à clé publique) est un type de cryptage qui utilise une paire de clés différente pour chiffrer et déchiffrer des données. La paire de clés est composée d'une clé publique, qui peut être partagée avec tout le monde, et d'une clé privée, qui doit être gardée secrète. Le cryptage à 256 bits signifie que la clé utilisée pour chiffrer et déchiffrer les données est de 256 bits de longueur. Cette longueur de clé est considérée comme suffisamment sécurisée pour la plupart des applications.
10. Trunk : Une interface en mode Trunk permet au switch d’envoyer des paquets aux autres appareils avec les tags VLANs.
11. Routeur : Un routeur est un dispositif qui permet à deux ou plusieurs réseaux différents de communiquer entre eux, c’est ce qu’on appelle le routage.
12. Linux : Linux est un système d'exploitation open-source et gratuit, c'est-à-dire un logiciel qui gère les ressources d'un ordinateur et permet aux utilisateurs d'interagir avec celui-ci.
13. Fichier hosts : C’est un fichier texte qui associe des noms d'hôtes à des adresses IP local, normalement consulté par la machine avant de contacter le serveur DNS.
14. Passerelle (Gateway) : C’est un élément de réseau qui agit comme une interface entre deux réseaux distincts, permettant à des données de passer de l’un à l’autre.
15. Routes statiques : Une route statique signifie que certains paquets de données qui sont destinés à un réseau spécifique seront envoyés à une adresse de passerelle prédéfinie, quelle que soit la topologie du réseau.
16. Kernel : Le noyau central du système d’exploitation Linux, gère les ressources matérielles de l’ordinateur et fournit une interface avec l’OS.
17. Pare-feu (Firewall) : Un système de sécurité qui contrôle et limite le trafic réseau entre les périphériques et les réseaux ou entre deux réseaux différents.
18. Pilotes (Driver) : Logiciel qui agit comme un intermédiaire entre le système d’exploitation et le périphérique matériel, leur permettant d’échanger des informations et de travailler ensemble efficacement.
19. Composants :
    1. CPU signifie "Central Processing Unit", ce qui se traduit par Unité Centrale de Traitement. C'est le composant principal d'un ordinateur qui est responsable de l'exécution des instructions et des opérations de calcul.
    2. GPU signifie "Graphics Processing Unit", ce qui se traduit par Unité de traitement graphique. C'est un processeur spécialisé conçu pour accélérer les calculs liés aux graphismes et à l'affichage.
    3. RAM, ou mémoire vive en français, est un type de mémoire informatique volatile qui stocke temporairement les données et les instructions que l'ordinateur utilise lors de son fonctionnement.
20. Overclocking : c’est une technique est utilisée pour améliorer les performances des composants comme CPU et GPU.

# Solution proposée

## Réflexion

L'infrastructure proposée utilise un Raspberry Pi 4 comme routeur, pare-feu, DNS et VPN, ainsi que deux autres Raspberry Pi 4 comme serveur NAS et serveur DHCP.

L'architecture réseau proposée utilise un commutateur 8 ports configurables pour diviser le réseau en plusieurs VLAN :

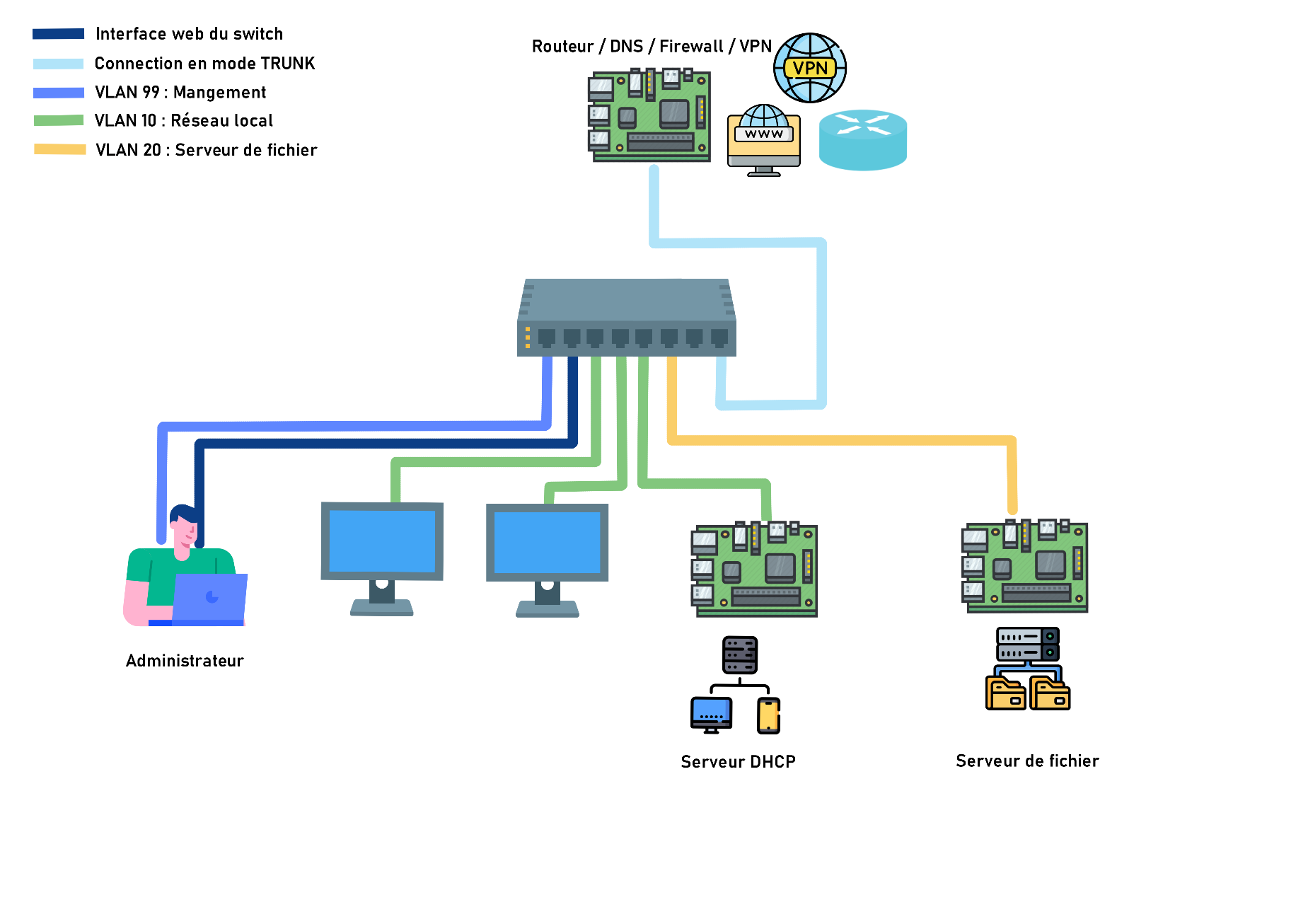
* Le VLAN 1 est utilisé pour configurer le commutateur ;
* VLAN 10 est le réseau principal qui contient les postes de travail et les serveurs ;
* VLAN 20 est utilisé pour le serveur de fichiers et d'application ;
* VLAN 99 est réservé à l'administration du réseau.

**Choix des Raspberry Pi 4**

Le choix d'utiliser des Raspberry Pi 4 comme routeur, serveurs d'application et serveur DHCP, est basé sur plusieurs raisons :

1. Les Raspberry Pi 4 sont abordables, ce qui en fait une solution idéale pour les petites entreprises ayant un budget limité, ils sont faciles dans l’installation et configuration grâce à leur system basé sur Debian et, où chaque problème est rapidement traité et résolu par la grande communauté du Raspberry.
2. Etant un système de carte unique, en cas de dysfonctionnement matériel il suffit de retirer le Microsd qui sert de mémoire à l’appareil, et le déplacer sur une nouvelle carte, sans avoir à faire aucune manipulation au niveau du logiciel.
3. Comme nous l’avons dit, le système d’exploitation est basé sur Debian, ce qui nous permet de profiter d’une liste presque illimitée et en constante évolution de services sous licence libre et souvent gratuits.

# Schéma de la solution



# Répartition des tâches

Une fois le projet est défini dans son intégralité, nous avons divisé les tâches à exécuter, en trois parties :

1. Configuration du routage et du switch
   1. Routage entre VLANs
   2. Création des VLANs
   3. Création et configuration du service DNS
2. Création et configuration de serveurs intranet
   1. Serveurs DHCP
   2. Serveurs NAS (OpenMediaVault)
3. Création et configuration du VPN

Bennis Matteo

Porco Olivier

Porco Olivier et Bennis Matteo

# Routage et VLANs

Afin de pouvoir travailler avec les équipements à notre disposition, nous avons opté pour une configuration routeur-on-a-stick, , ainsi dénommée, parce qu’on dirait que le routeur est posé sur un bâton, voir schéma au-dessous (figure 1).

Figure 1

Switch

## Matériels

Router : Raspberry pi 4B 4Gb di RAM + Adaptateur USB Ethernet

Commutateurs : D-Link DGS-1210-10

Câbles : Ethernet cat 6

## Commutateur et VLANs

Le commutateur D-Link DGS-1210-10, 8 ports avec interface de configuration web, adapté pour les petits réseaux, avec un prix abordable, environ 100€ et une consommation de courant relativement faible d’environ 13Wh.

## Difficultés rencontrées

Le commutateur joue un rôle important dans ce projet, la division du réseau en sous-réseaux virtuels (VLANs).

Dans le passé, j’ai travaillé exclusivement avec des équipements Cisco, qui ont une procédure de configuration totalement différente du DLink, donc la première difficulté, et aussi la plus importante, était de comprendre comment configurer les VLANs.

Malheureusement, nous n’avons pas trouvé un manuel spécifique de notre DLink ni dans sa boîte ni en ligne.

Après environ deux jours de recherche, grâce aux informations fragmentées que j’ai trouvées en ligne et à quelques conseils reçus d’un de nos collègues, j’ai compris deux choses qui m’ont permis de terminer la configuration du switch :

1. Il n’y a pas de mode Trunk dans les commutateurs DLink, on l’obtient en ajoutant manuellement le tag sur la porte en question pour toutes les VLANs
2. Il y a deux pages dans l’interface web importantes pour la création et la gestion des VLANs :
   1. Une s’occupe de la gestion des tags sur les interfaces.
   2. L’autre gère le comportement de l’interface avec les paquets en transit.

## Nos VLAN

Ip réseau interne 172.16.0.0/16

* VLAN 1

Est utilisé pour se connecter au switch et le configure

Subnet Address: 172.16.0.0/24

Host Address Range: 172.16.0.1 - 172.16.0.254

Broadcast Address: 172.16.0.255

* LAN 10 (LAN)

Contient les postes utilisateurs et le serveur DHCP

Subnet Address: 172.16.1.0/24

Host Address Range: 172.16.1.1 - 172.16.1.254

Broadcast Address: 172.16.1.255

* VLAN 20 (serveur de fichier et d’application)

Subnet Address: 172.16.2.0/24

Host Address Range: 172.16.2.1 - 172.16.2.254

Broadcast Address: 172.16.2.255

* VLAN 99 : (Administration Réseaux)

Vidéo [YouTube](https://youtu.be/5CENnfgniGI)

## Routage

Pour la configuration router-on-a-stick, il y a besoin de deux interfaces RJ45, mais le Raspberry pi n’en a qu’une, nous avons donc acheté une carte réseau externe, compatible avec Linux.

## Difficultés rencontrées

Dans la gestion du routage via Raspberry pi la première étape est le choix du système d’exploitation, dans notre cas Openwrt, un système d’exploitation (OS) basé sur une version lite du kernel linux, qui permet, via une interface web, la gestion du routage et du pare-feu.

Après plusieurs tests, nous avons remarqué l’absence de compatibilité entre l’OS et l’adaptateur USB-Ethernet, le premier instinct a été de consulter la [documentation](https://openwrt.org/toh/raspberry_pi_foundation/raspberry_pi) d’Openwrt ; en fait elle indique que les seules cartes supportées sont Realteck ou Intel.

Pour résoudre le problème, nous avons envisagé trois solutions possibles :

* Ajouter manuellement le pilote
* Acheter un nouvel adaptateur
* Changer d’OS

La première option est impraticable, car le kernel modifié pour ce système empêche l’installation de tout paquet supplémentaire, ce qui est compréhensible du point de vue de la sécurité.

La deuxième option, vus les prix élevés des adaptateurs USB-Ethernet Realteck/Intel compatibles avec Linux, elle a été immédiatement écartée.

Donc, nous avons opté pour un changement d’OS en installant un Raspbian x64 Lite (Sans interface graphique).

En se basant sur le TP fait avec le professeur Alexandre GUY, j’ai écrit un script adapté pour la gestion du routage et du pare-feu que j’ai mis en exécution au démarrage.

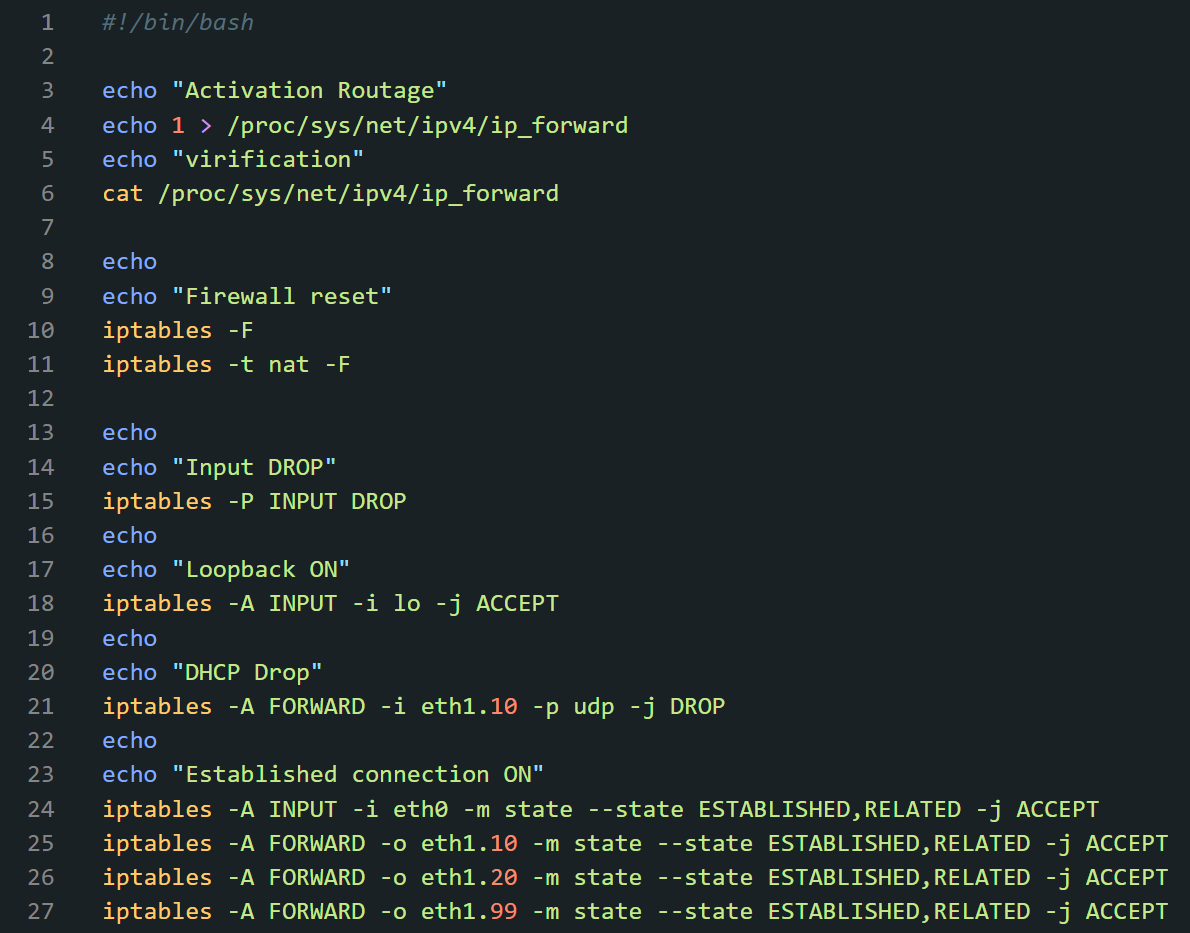


Fig iii Script Routage Page 1

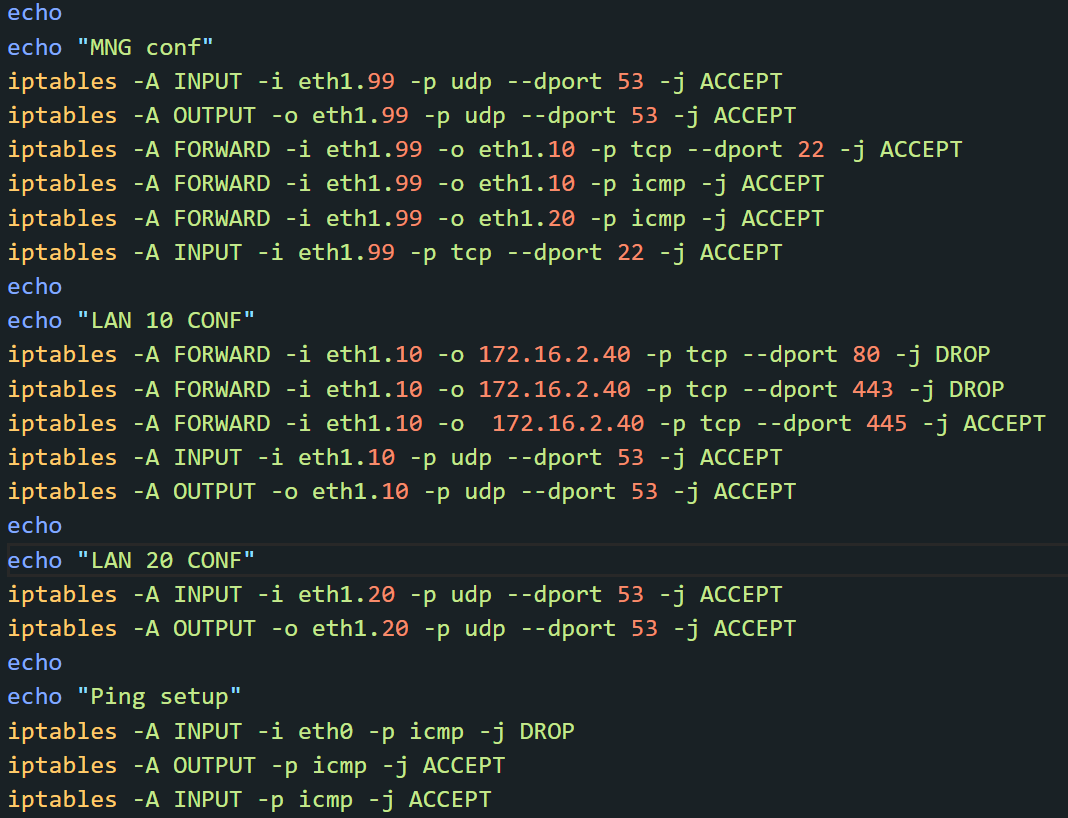


Fig iv Script Routage Page 2

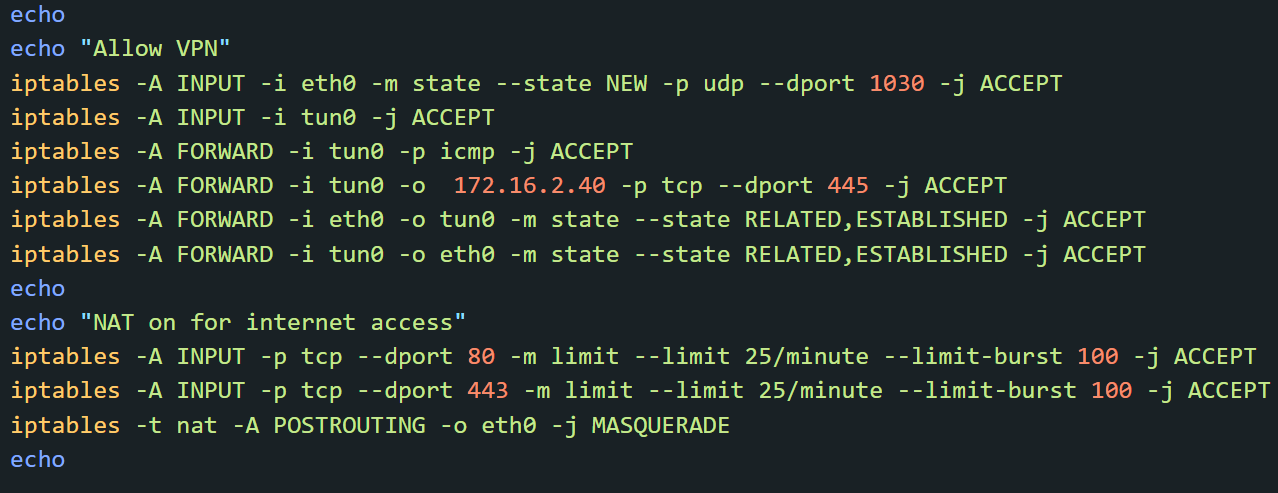


Fig v : Script Routage Page 3

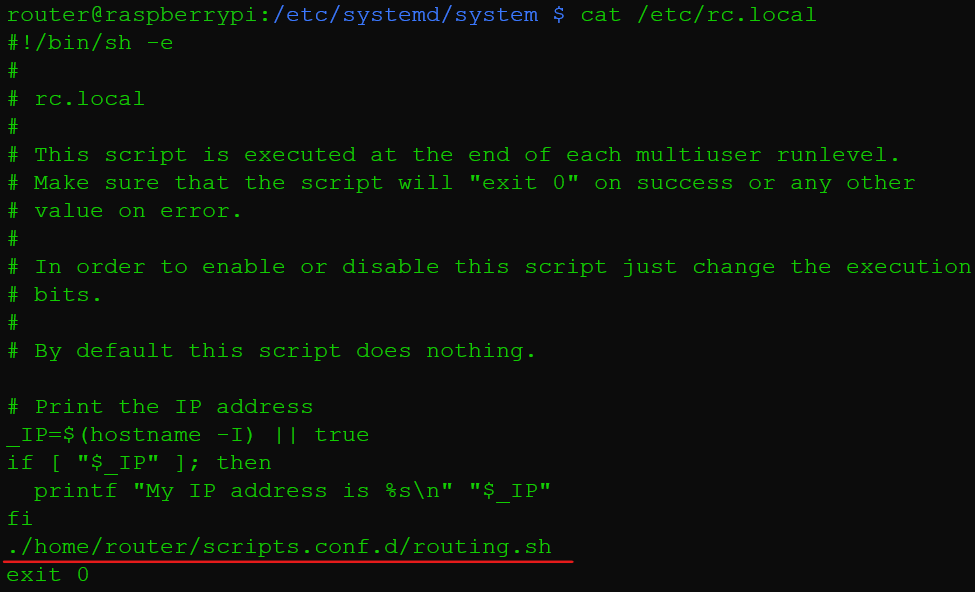


Fig vi fichier modifié pour l'exécution du script au démarrage

Nous avons ensuite rencontré un deuxième problème dans la gestion du routage entre vlans : le système de contrôle des interfaces de base dans Raspbian, avait du mal à gouverner les sous-interfaces.

Nous avons donc décidé de le configurer avec le système Netplan, qui permet une gestion concentrée des interfaces et sous-interfaces, à travers le même fichier. [[Source](https://askubuntu.com/questions/992428/netplan-with-multiple-vlans-on-single-interface-help-needed)]

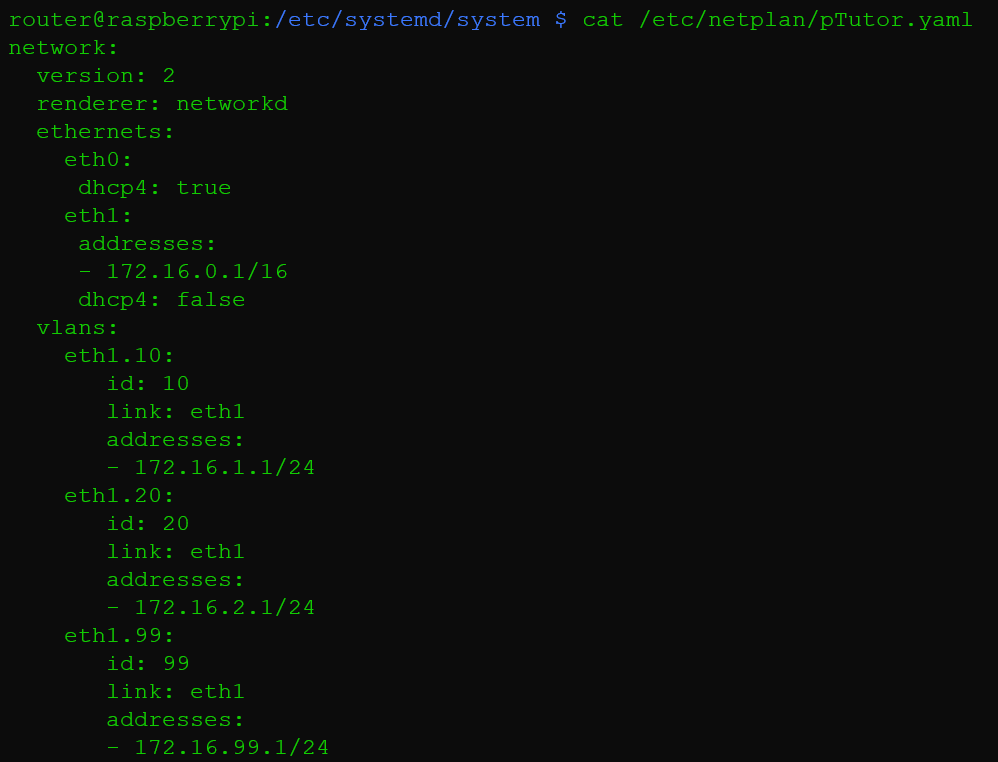


Fig vii fichier Netplan du router

Le troisième et dernier problème rencontré dans le routage, c’est la vitesse, même si nous avions un port RJ45 1Gb/s le processeur avait du mal à gérer cette quantité de données, donc nous avons décidé de faire un overclocking partiel du processeur, en suivant un [tutoriel](https://www.youtube.com/watch?v=TwDbQ26_Mp8&t=16s), trouvé en ligne.

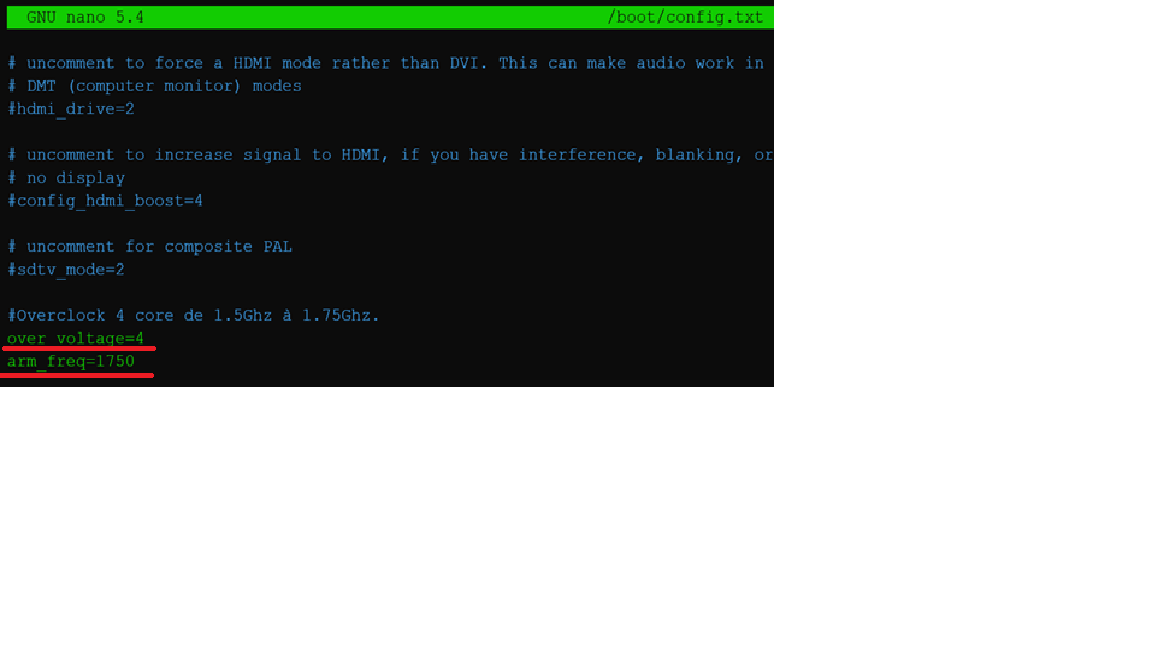


Fig viii Lignes à ajouter pour overclocker

## VLAN 99

Présentation Routage [[YouTube](https://youtu.be/5CENnfgniGI)]

# Serveur de fichier

## https://cdn4.iconfinder.com/data/icons/computer-network-4/64/data-management-file-folder-data-base-server-512.pngOpenMediaVault

OpenMediaVault est un logiciel qui peut être installé sur presque toutes les distributions Linux. Il peut être utilisé pour héberger et configurer un serveur de fichiers en quelques clics via une interface web intuitive.

Pourquoi avons-nous choisi OpenMediaVault ?

Compatibilité : Grâce à la large communauté, une version de ce service parfaitement compatible avec le système d’exploitation Raspbian OS a été créée.

Gestion de stockage : OpenMediaVault offre une gestion complète des disques durs, de la sauvegarde et de la restauration des données, et permet de créer facilement des espaces de stockage partagés pour les utilisateurs.

Fig ix: [Source](https://static1.squarespace.com/static/504eb270e4b07d0f5698991c/50e37755e4b0f771660d0d30/5f2030a7bfb88f2f3a1d8b74/1595949797548/file_server_migration_windows_cloud.png?format=1500w)

Interface d’administration : l'interface Web d'OpenMediaVault est simple et intuitive, ce qui facilite la configuration et la gestion de la plateforme.

Plugins : OpenMediaVault prend en charge de nombreux plugins pour ajouter des fonctionnalités supplémentaires, telles que la surveillance du système, le partage de fichiers via FTP, la configuration de RAID, etc.

Sécurité : OpenMediaVault propose des options de sécurité avancées, telles que la gestion des utilisateurs et des permissions, le chiffrement des disques.

#### Configuration :

**Configuration du Raspberry**

* username: admisys
* password: Azerty66
* static ip\_adresse=172.16.2.40

**Configuration du réseau avec netplan**

Netplan est un outil de configuration de réseau pour les systèmes d'exploitation basés sur Linux, utilisé pour définir et configurer les interfaces réseau sur des ordinateurs avec des systèmes d'exploitation tels que Debian. Il permet de définir les configurations réseau dans un fichier YAML, qui est ensuite converti en configurations système standard.

## Configuration d’OpenMediaVault

La mise en place du serveur de fichier est présentée dans la vidéo

[OpenMediaVault: Presentation et configuration du serveur de fichier](https://www.youtube.com/watch?v=19SP7Zv-1g8&list=PLUC0qhau2T7z9tG4cqSggsDdqcwn7r9i5&index=1)

Il est possible d'ajouter des utilisateurs manuellement ou avec un csv, cela offre la possibilité de créer des scripts pour automatiser la création d’utilisateur Lors de l’ajout manuel d’un utilisateur, OpenMediaVault utilise la commande [useradd].

# Serveur DHCP

## Présentation de DNSmasq

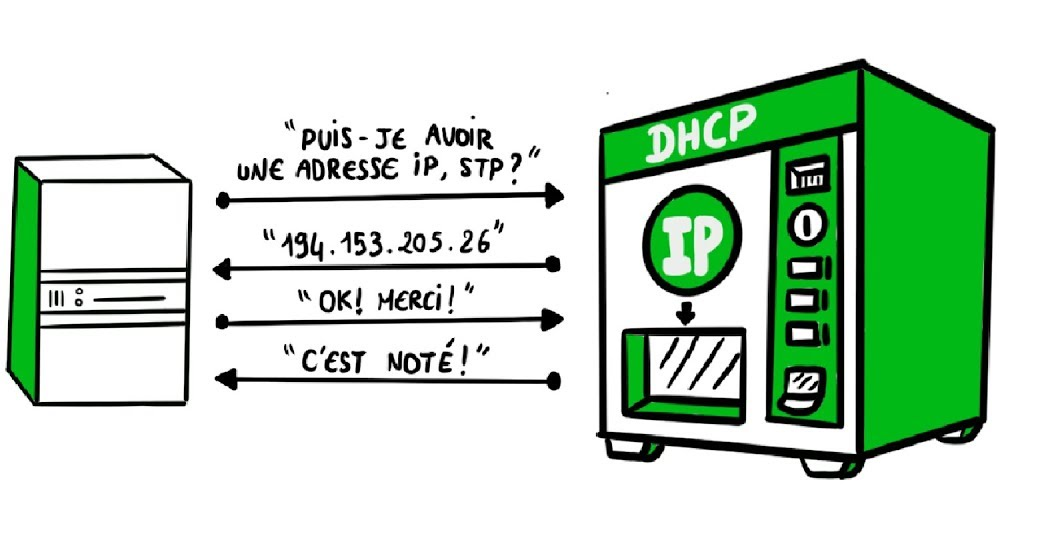
Dnsmasq est un service qui offre la possibilité de mettre en place un serveur DNS et DHCP sur les systèmes Linux. Il est capable de fournir des adresses IP et des informations de configuration du réseau aux postes connectés en locale.

Fig x: [Source](https://discordapp.com/channels/@me/1017713153821392960/1085591448851857509)

Pourquoi avons-nous choisi DNSmasq comme service DHCP ?

* En effet, Ce service a été conçu pour fournir une solution complète pour les réseaux locaux. Il est rapide et facile à configurer au moyen d’un seul fichier.
* De plus DNSmasq est compatible avec tous les systèmes d'exploitation et matériels réseau.

En somme, DNSmasq est une solution simple, légère et polyvalente pour un serveur DHCP, qui répond parfaitement aux besoins d'une petite entreprise.

## Configuration

Comme pour le serveur d’application l’ip statique est configuré avec **netplan.** La configuration du service **DNSmasq** est présentée dans la vidéo

[Dnsmasq: Configuration du serveur DHCP](https://www.youtube.com/watch?v=etOPF3clNdU)

**Configuration du Raspberry**

* username: admisys
* password: Azerty66
* static ip\_adresse=172.16.1.250/24

# Serveur DNS

## DNS

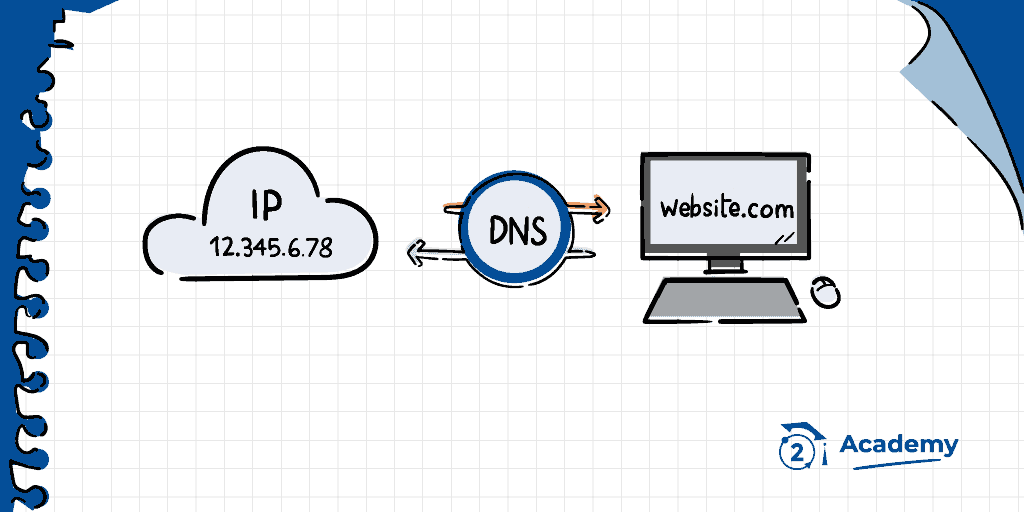
Un service DNS privé est indispensable pour la sécurité et l’ergonomie d’un réseau.   
Ergonomie : L’une des fonctions les plus utiles d’un serveur DNS est de pouvoir associer une adresse IP privée à un FQDN valide uniquement sur le réseau local, ce qui permet à tous les utilisateurs du réseau d’accéder au service sans mémoriser l’adresse IP.

Fig xi: [Source](https://academy.bit2me.com/wp-content/uploads/2019/05/49_DNS.png)

Sécurité : le serveur DNS sauvegarde une liste des associations d’adresses IP – FQDN, en cas de piratage du service DNS le pirate aurait accès à la liste de toutes les adresses IP des services de tous les réseaux qui en font usage, donc garder le serveur en local nous donnerait un niveau de sécurité supplémentaire.

Pour notre réseau, le service DNS est géré par le Raspberry pi qui s’occupe du routage.

Comme service, nous avons envisagé plusieurs options :

1. Bind9
2. PowerDNS
3. dnsMasq

## Problèmes rencontrés

Bind9: Service complet pour un usage professionnel, souvent mis à jour et disponible avec interface web pour la gestion des zones, très rapide dans l’installation, mais pas dans la configuration et, étant gourmand de ressources, le Raspberry pi ne pouvait pas gérer le service et le routage en même temps, donc nous avons dû l’écarter.

PowerDNS: Service rapide dans l’installation et la configuration, toujours à jour mais pas compatible avec les processeurs ARM, donc inapplicable à notre projet.

dnsMasq: Notre choix est tombé sur ce service, rapide dans l’installation et la configuration, le seul problème rencontré est la compréhension de la gestion des zones inverses (la gestion de l’attribution des adresses IP locales à FQDN).

Après de nombreuses recherches et tests, nous avons compris que la zone de recherche inversée utilise le fichier hosts du serveur.

# Serveur VPN

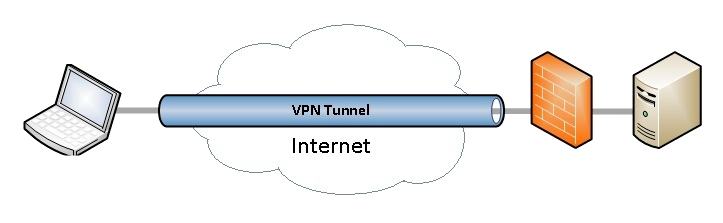


Fig xii: [Source](https://vpntunnel.com/wp-content/uploads/2020/08/imageedit_23_2459998402.png)

La première étape du montage d’un tunnel VPN consiste à choisir entre un protocole ipsec ou SSL.

Il faut d’abord comprendre la différence entre les deux :

* ipsec à protocole agissant au niveau 3 [OSI](https://upload.wikimedia.org/wikipedia/commons/thumb/8/8d/OSI_Model_v1.svg/langfr-800px-OSI_Model_v1.svg.png) et gourmand de ressources.
* SSL à protocole agissant sur le niveau 5 OSI, contenu dans l’utilisation des ressources et très sécurisé puisqu’il utilise une méthode de cryptage asymétrique de 256 bits.

#### Après plusieurs recherches et sur les conseils du professeur Alexandre GUY, nous avons opté pour un VPN de type SSL, dans notre cas OpenVPN.

#### OpenVPN : Service open source, gratuit et compatible avec tous les systèmes d’exploitation.

#### La communauté Raspberry pi et OpenVPN a mis à disposition plusieurs tutoriels pour l’installation et la configuration du service.

## Problémes rencontrés

Malheureusement, toutes les procédures adaptées à Raspbian exigent que l’utilisateur passe par les serveurs privés d’OpenVPN et créer un compte, ce qui peut être faisable dans certains cas, mais il ne convient pas aux les professionnels.

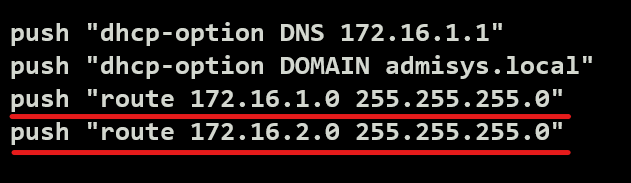
Notre objectif était de créer des serveurs VPN locaux afin d’avoir une connexion directe entre l’utilisateur et le serveur, nous avons donc été obligés de réadapter une procédure créée pour Debian à Raspbian, qui n’a pas nécessité de grandes modifications.

Les premiers problèmes rencontrés étaient la gestion du routage des utilisateurs connectés via VPN.

Une fois modifier le script de routage, pour la connexion au serveur et la redirection au Vlan10. Nous devrions avoir les mêmes droits d’accès qu’un utilisateur en local, qui ce n’était pas vérifié dans notre cas.

Pour résoudre ce problème, nous avons suivi la procédure suivante :

1. Après plusieurs recherches sur le forum OpenVPN, nous avons compris que la gestion du routage était effectuée par le serveur lui-même, nous avons donc été obligés de réadapter le fichier de configuration du serveur pour rediriger le trafic vers les deux VLAN.



1. Le dernier problème était de configurer automatiquement la connexion de l’utilisateur au serveur de fichiers, sans désactiver à chaque fois toutes les interfaces réseau, la seule solution que nous avons trouvée était d’ajouter routes statiques dans le script de configuration du chemin réseau, le tout en utilisant les commandes suivantes :

route -add à Windows (powershell)

ip route add à Linux / macos

# Webographie

#### Playlist du Projet sur YouTube

[Projet ADMISYS](https://youtube.com/playlist?list=PLUC0qhau2T7z9tG4cqSggsDdqcwn7r9i5)

#### Service d’application OpenMediaVault

[💻 CREEZ votre SERVEUR NAS avec un RASPBERRY PI sur OPENMEDIAVAULT ! Serveur de fichiers](https://www.youtube.com/watch?v=dKjfwzs5Myw&t)

<https://github.com/OpenMediaVault-Plugin-Developers/installScript>

#### VPN

<https://std.rocks/fr/vpn_openvpn_bullseye.html>

#### DNS

<https://www.section.io/engineering-education/setting-up-a-private-dns-server-with-raspberry-pi/>

Autres liens utiles:  
<https://www.tala-informatique.fr/wiki/index.php/Iproute2>

<https://linux.goffinet.org/administration/configuration-du-reseau/gestion-du-reseau-linux-avec-iproute2/>

<https://www.digitalocean.com/community/tutorials/how-to-list-and-delete-iptables-firewall-rules>

<https://www.daddaily.life/building-a-router-with-raspberry-pi-router-on-a-stick-en/>

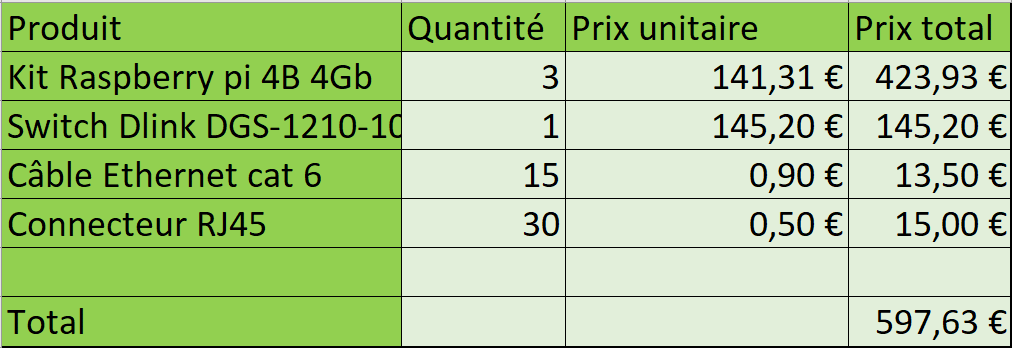
# Annexes

## 

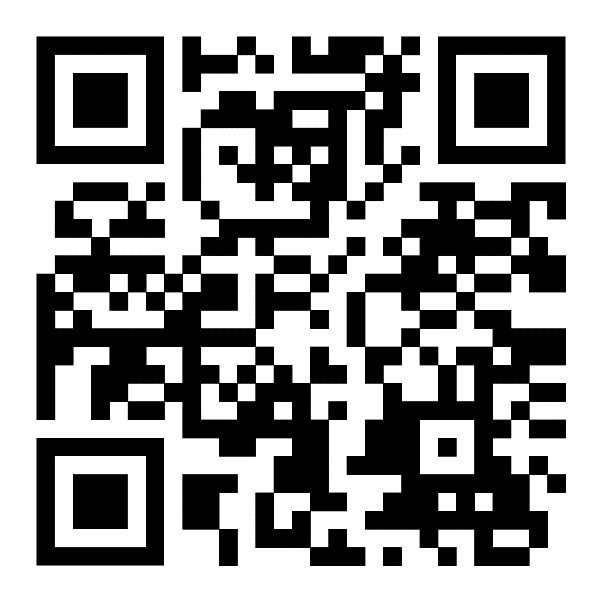
## [Diagramme de Gantt](https://drive.google.com/file/d/12Jxn9AKsRxTHV5xkqUc-EncfJ9Bp4fPx/view?usp=share_link)

## Devis

Le devis suivant indique les dépenses effectuées par nous, en ce qui concerne le devis pour un hypothétique client il faudrait ajouter la main d’œuvre et la marge bénéficiaire.



## Documentation



[GitHub](https://github.com/Th3DarkOn3/ProjetTuteur)