

Rapport du projet

Réseau

BUT:

- ▶ Développer une application
 - ▶ connecter un réseau Wi-Fi ad hoc (adressage via DHCP)
 - ▶ Se connecter à un point d'accès Wi-Fi
 - ▶ Approximer une distance grâce à la puissance du signal reçu
 - ▶ Communiquer via deux interfaces différentes sur un même nœud (WiFi+bluetooth, ou Wi-Fi + Ethernet)
 - ▶ Communiquer via client/serveur
 - ▶ Récupérer une adresse GPS depuis une webapp sur téléphone
- ▶ Exemples d'applications:
 - ▶ véhicules autonomes, contacts entre individus sur Tousanticovid, surveillance de patients dans un établissement médicalisé, jeu collaboratif, ...

BOUNEGTA Mohamed Rami

MABROUK Sermed

BOUGHADOU Kenzi

GUINDOU Seydou

Plan :

Configuration des RaspberryPi	Tâches	Statut	Personnes	Semaine
Mise en oeuvre d'une solution informatique pour le projet : un cahier de charges peut être ?	2	Done	5	5 weeks ago
Collecte d'informations		Done	5	5 weeks ago
recherche d'exemple, de tutoriels, d'idées...		Done	5	5 weeks ago
+ Ajouter Élément				
Ecriture sur les cartes SD (Raspberry Pi Imager)	Tâches	Statut	Personnes	Semaine
Mohamed		Done	1	4 weeks ago
Sermed		Done		3 weeks ago
Kenzy		Done		4 weeks ago
Seydou	1	Stuck		4 weeks ago
+ Ajouter Élément				
Création du Réseau	Tâches	Statut	Personnes	Semaine
Ad-hoc		Working on it	5	3 weeks ago
Point d'accès		Done	5	3 weeks ago
+ Ajouter Élément				
Développement de l'application	Tâches	Statut	Personnes	Semaine
Coder l'application		Done	5	1 week ago
Interface graphique de l'appli		Working on it	5	1 week ago
Site web qui donne les informations client serveur		Done	5	1 week ago
+ Ajouter Élément				

Nous allons vous présenter dans ce rapport l'approche que nous avons entreprise afin de mettre en place et installer le réseau entre nos Raspberry et développer une application interactive entre eux.

Ce projet nous a appris à manier le matériel donné et acquérir les notions de base du réseau.

C'était notre premier projet en réseau en utilisant un Raspberry, nous avons connecté nos Raspberrys sur un réseau WiFi ad hoc avec un adressage via DHCP. On a connecté nos Raspberry sur un même nœud WiFi+bluetooth.

Vu que c'était notre première fois on s'est renseigné sur internet et on a essayé plusieurs guides.

Afin de commencer on s'est mis tout d'abord d'accord sur le langage à utiliser et l'application à implémenter. On a choisi de coder en python et implémenter une application de chat.

Les grandes étapes :

Notre travail est partagé en 3 grandes étapes

1. Configuration des Raspberry
2. Création du réseau
3. Développement de l'application

1. Configuration des Raspberry pi :

Nous avons installé le logiciel Raspberry Pi Imager afin de télécharger l'OS Rasbian sur la carte SD de chaque Raspberry pi nous avons réglé les paramètres de chaque Raspberry (heure, date, région mais

surtout le wifi de la région et activer le SSH les deux paramètres les plus importants pour la suite de notre projet après tous ces réglages nous pouvons passer à la prochaine étape.

2.Création du réseau :

C'est l'une des parties qui nous a pris le plus de temps et dans laquelle on a beaucoup appris tout d'abord on a suivi des guides pour la création d'un réseau ad hoc wifi et un adressage via DHCP nous avons appliqués les instructions à la lettre mais ça ne semblait pas fonctionner

Fichier Interfaces modifier pour Ad-Hoc

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp

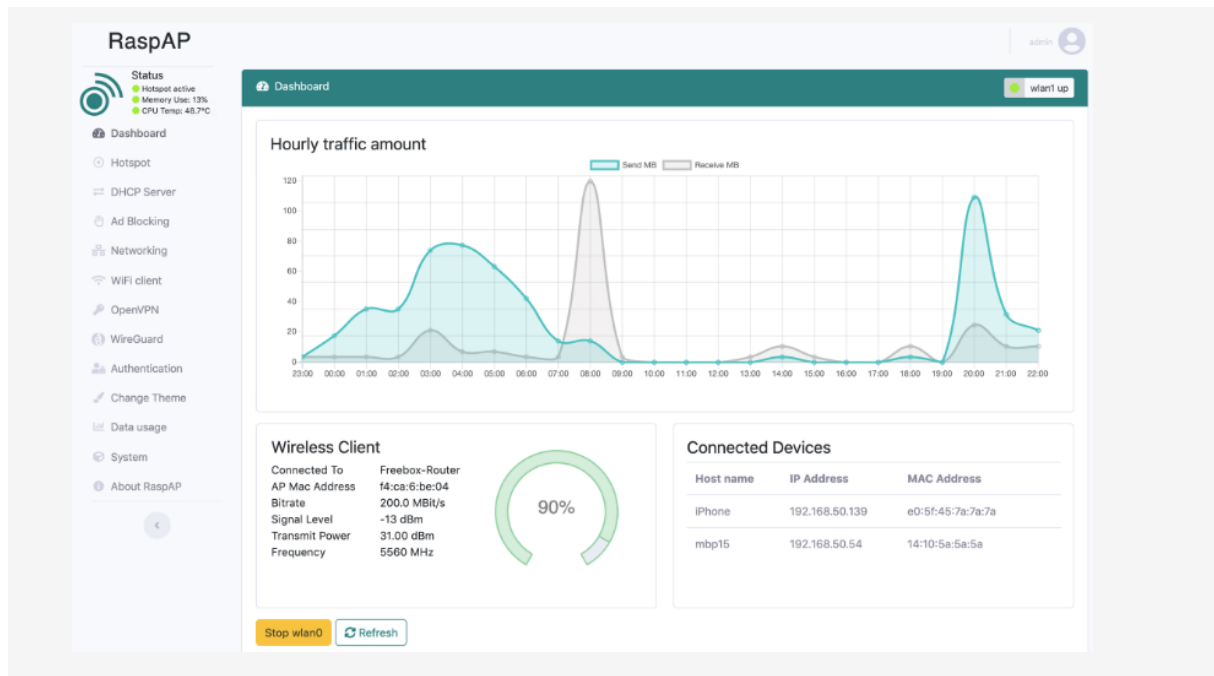
auto wlan0
iface wlan0 inet static
address 192.168.1.1
netmask 255.255.255.0
wireless-channel 4
wireless-essid RPitest
wireless-mode ad-hoc
```

Fichier dhcp.conf modifié pour l'adressage des adresses IP

```
ddns-update-style interim;
default-lease-time 600;
max-lease-time 7200;
authoritative;
log-facility local7;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.5 192.168.1.150;

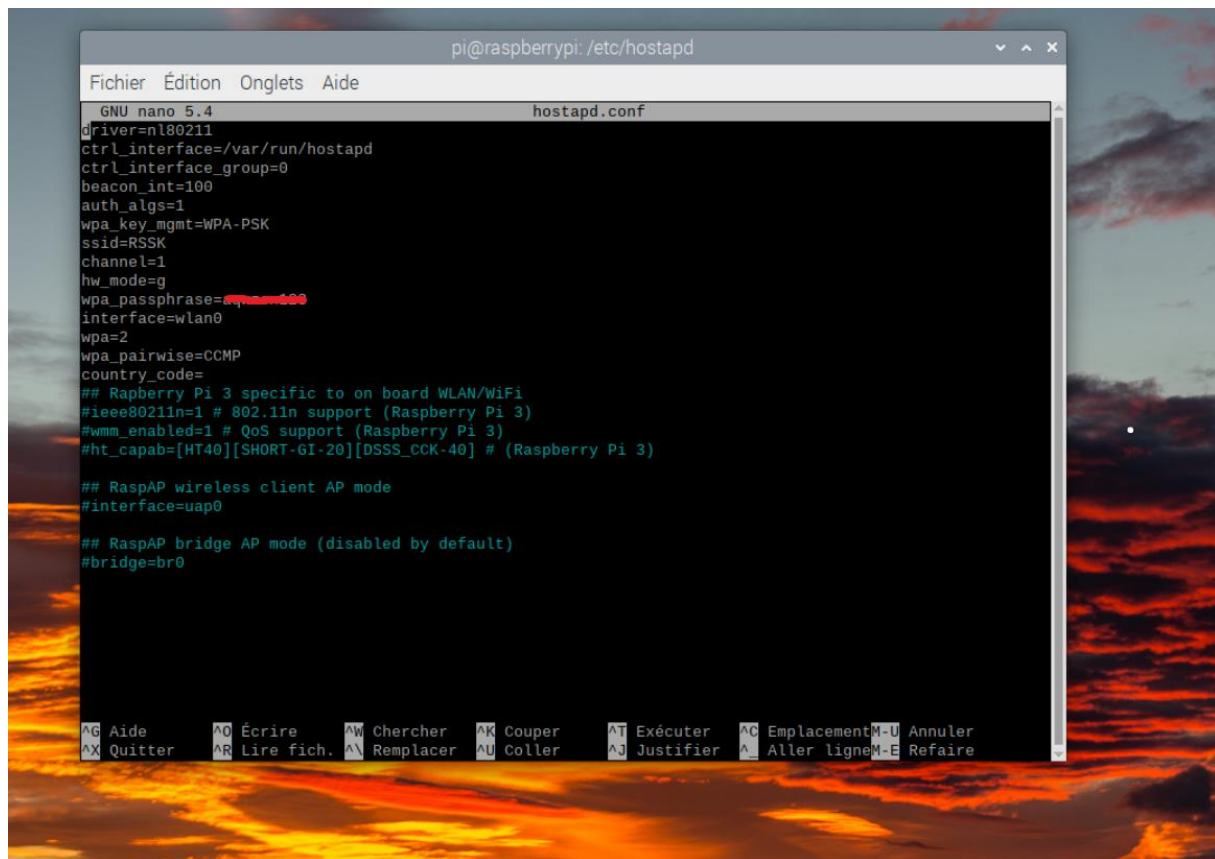
}
```

Après plusieurs essais cela n'a pas fonctionné on s'est alors retourné vers un réseau Point d'accès (Access Point) on a installé un logiciel RaspAP qui permet de configurer des routeurs basés sur Debian il permet aussi de créer aussi une interface personnalisable



Ce logiciel configure automatiquement les fichiers interfaces et dhcp.conf afin de passer en Ap mode et attribut une adresse ip par défaut aux raspberry dans le fichier wlan0 et qui est repris par le fichier interfaces qui est de 10.3.143.1

Ensuite pour personnaliser notre Raspberry AP (serveur) lui attribuer un SSID et un mot de passe on a dû changer hostapd.conf de cette façon



```
pi@raspberrypi: /etc/hostapd
Fichier  Édition  Onglets  Aide
GNU nano 5.4                                hostapd.conf
driver=nl80211
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
beacon_int=100
auth_algs=1
wpa_key_mgmt=WPA-PSK
ssid=RSSK
channel=1
hw_mode=g
wpa_passphrase=a
interface=wlan0
wpa=2
wpa_pairwise=CCMP
country_code=
## Raspberry Pi 3 specific to on board WLAN/WiFi
#ieee80211n=1 # 802.11n support (Raspberry Pi 3)
#wmm_enabled=1 # QoS support (Raspberry Pi 3)
#ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40] # (Raspberry Pi 3)

## RaspAP wireless client AP mode
#interface=uap0

## RaspAP bridge AP mode (disabled by default)
#bridge=br0

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement ^U Annuler
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne ^E Refaire
```

Et maintenant on peut se connecter à notre RaspberryAP (serveur) et on peut maintenant développer notre application .

Développer l'application :

Pour le développement de l'application, on a été très ambitieux au début, on voulu créer une application qui permet de lancer un fichier mp3 depuis un autre Raspberry c'est-à-dire le fichier est stocké seulement dans l'autre Raspberry, grâce à un code en c++.

A cause de la complexité du code permettant la création de cette appli + les échanges de données assez importantes, du coup la limite des performances du Raspberry aussi,

nous avons abandonné et nous nous sommes rabattus sur une autre application, pour créer une messagerie pour échange de chat entre utilisateurs, grâce à la notion de socket qui nous a été très utile.

On a bien réussi à la développer et elle fonctionne parfaitement.

```
pi@raspberrypi:~/messagerie $ python3 client.py
Choose your nickname: Rami
Rami joined!
Connected to server!
Salut
Rami: Salut
█
```

```
pi@raspberrypi:~/messagerie $ python3 serv.py
Connected with ('10.3.141.235', 52754)
Nickname is Rami
█
```

On a aussi ensuite voulu lui rajouter une interface graphique en utilisant quelques packages, c'est les librairies kivy et kivymd qu'on a installé avec la commande pip.

```
Sudo apt-get install libav-dev libsdl2-dev libavformat-dev libavdevice-dev

Sudo pip3 install ffpypyplayer
Sudo pip3 install kivy
Sudo pip3 install kivy-examples

#to test if kivy is working run this command
python3
#then
import kivy
```

Bilan :

Cette UE est pour nous l'une des plus intéressantes découverte en apprenant un domaine , ou on a repris des connaissances en réseaux en plus avancés qu'on a transposer dans notre expérience en développement et de mise en œuvre de solution informatique.

Conclusion :

On a était confronté à un travail de type de recherche/développement, c'était surtout des recherches de tutoriels informatiques et recherches de bibliographies pour les fonctionnalités qu'on a voulu créer pour l'application, et bien évidemment vu que c'était un configuration puis un développement ; le travail consistait plus à un avancement collectif plus qu'à une répartition des tâches vu que la nature du travail, cela a demandé une forte communication au sein du groupe.

Avis, motivation et ressentis :

Le travail sur ce projet s'est très bien passé ; cette première expérience pratique en cette discipline était motivante et très intéressante, elle nous a vraiment plu et on sera encore plus déterminé pour une prochaine expérience.