

GSW

Alan Menet, Azad Courtial, Corentin Brasseur
et Clément Boulmont





Sommaire

- Contexte
- Objectif du projet
- Description du matériel
- Partie Raspberry Pi
- Partie Python
- Expérimentations
 - Comparatif entre plusieurs configurations
- Conclusion



Contexte

- Projet Thématique 1
- Master 2 Informatique
 - ISRI
 - E-santé



Objectif du projet

L'objectif de ce projet est de créer une solution de détection de téléphone portable.

Cela doit se faire sans manipulation préalable sur le dit téléphone.

On se propose de passer par les prob request émis par le téléphone lors de la recherche de réseau.

On peut imaginer une utilisation pour restreindre l'accès à une zone sans avoir à installer des caméras de surveillance.

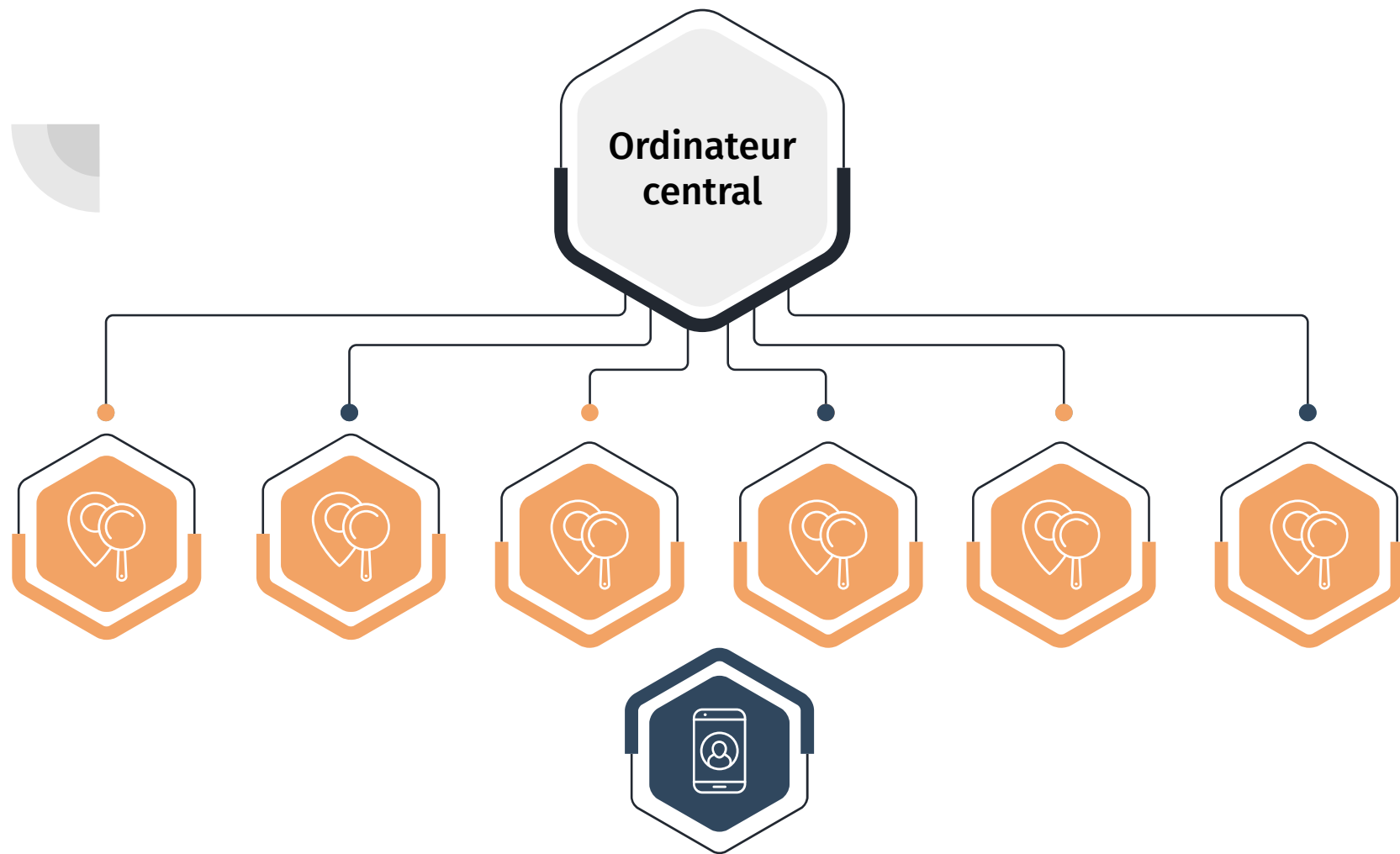


Description matériel

On dispose de raspberry avec aircrack qui vont nous permettre de scanner les demandes de connections.

Mais également de Dongles Wifi branchés sur les raspberry.

Et un moniteur central pour la centralisation des données





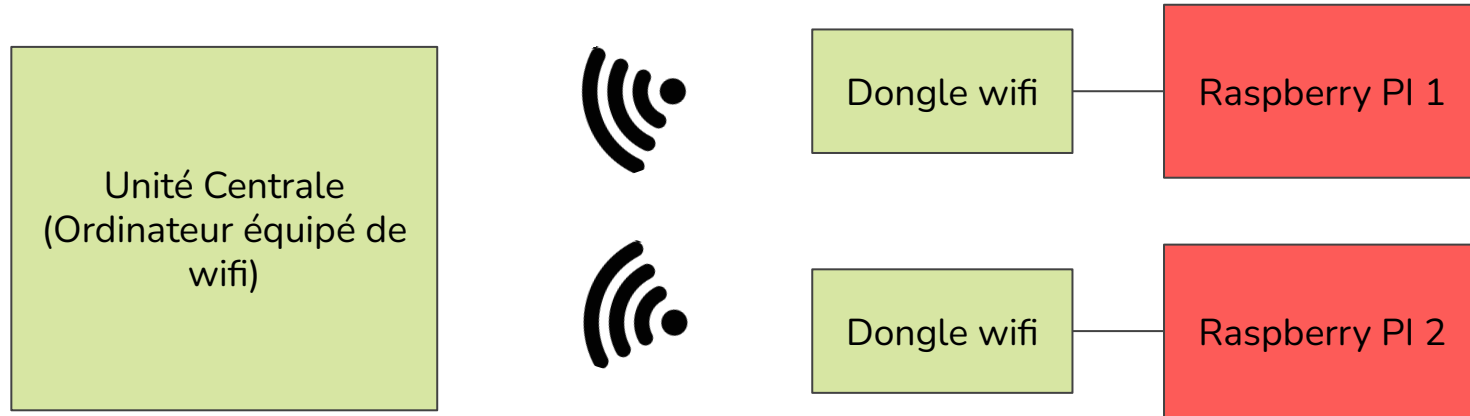
Partie Raspberry Pi

Récupération des données

Quelques contraintes :

- Puissance du signal dépend de plusieurs facteurs :
 - Position
 - marque du téléphone
 - Environnementaux (poche / main ; pluie, soleil, humidité ; mur en plâtre, béton, sous sol)
 - Autre source d'énergie tel que des ordinateurs, le micro-onde
- Absence du mode moniteur sur raspberry
 - Conséquence : blocage au niveau de nos expérimentations
 - Utilisation de 3 dongle wifi avec les raspberry
- Besoin de faire de l'étalonnage pour placer nos capteurs

Comment récupère t-on les données ?





Prérequis :

- Mode moniteur, non disponible sur les interfaces wifi par défaut de raspberry pi modèle 4
- Les commandes nc (netcat) et tcpdump
- Aircrack, plus précisément airmon pour activer le mode moniteur



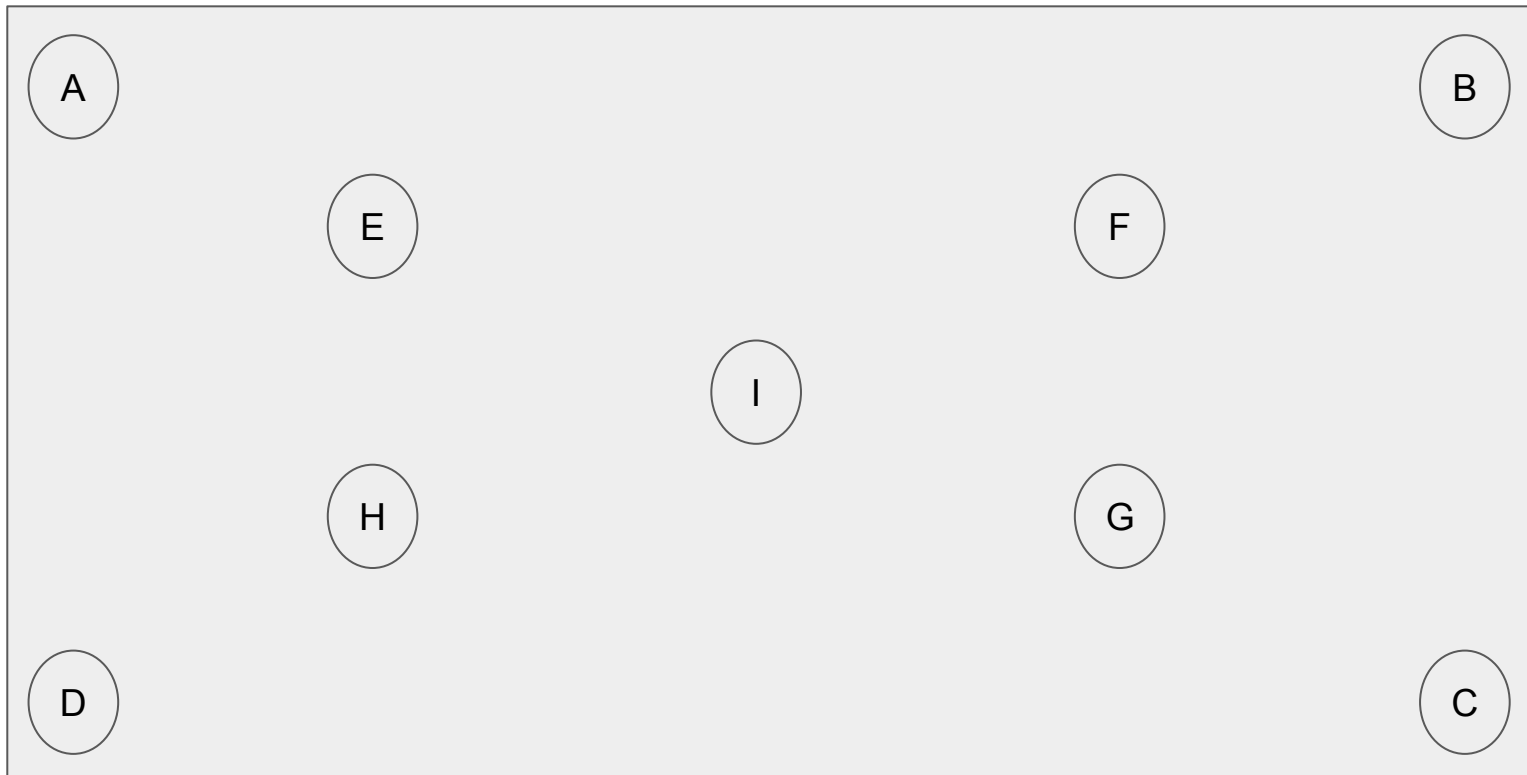
Scripts

- Les raspberry envoient les données de tcpdump via netcat à l'unité central.
- L'unité central parse les données dans un csv où chaque ligne contient le numéro du capteur, la puissance reçue, l'heure de réception et l'adresse mac.
- création d'une whitelist et d'une new list.
- Csv auquel accède le code de la partie analyse des données.

Numéro de capteur	Heure	Puissance	Adresse MAC
1	00:00:00.00000	-1dBm	00:00:00:00:00:00



Étalonnage

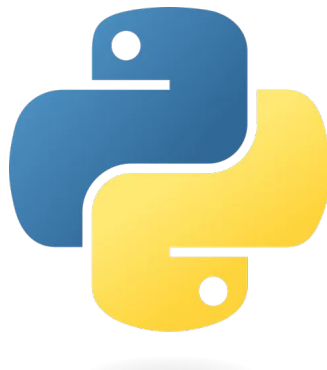




Partie Python

Traitement des données

- Outils de développement
 - Python
 - Bibliothèques Python
 - Numpy
 - Pandas
 - Script Shell





Partie Python

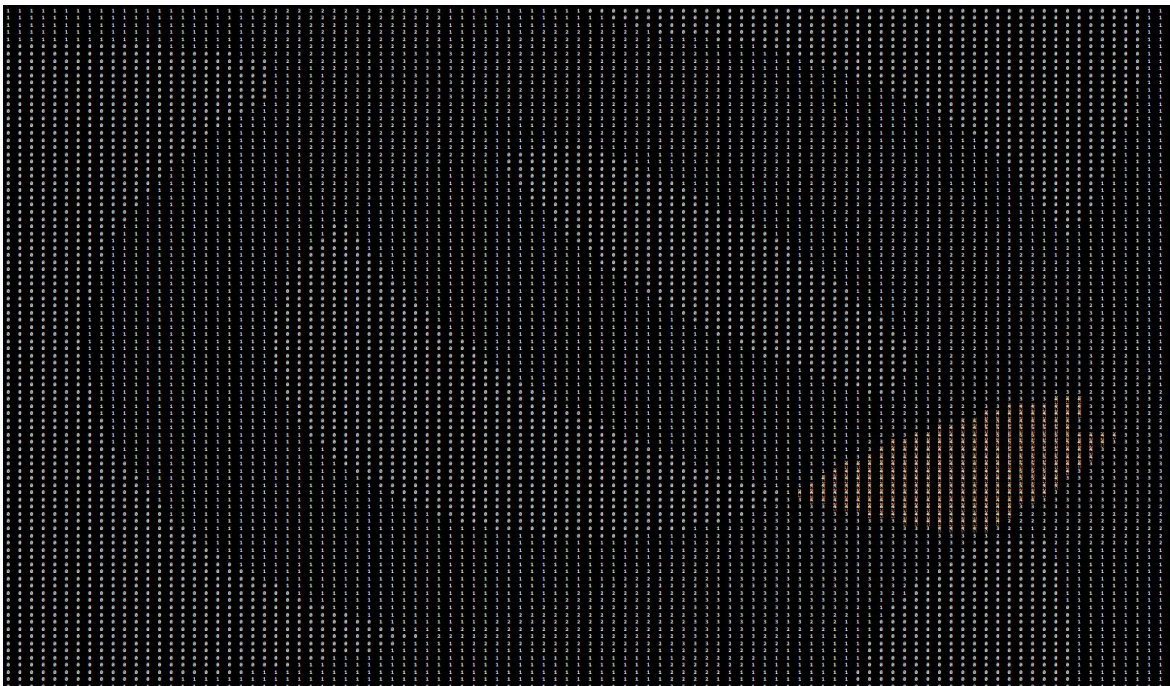
Localisation par Triangulation

- Comparer la distance
- Classer les positions d'étalonnage
- Mettre en place de poids selon le rang
- Création d'une carte
- Elaboration de la zone résultat

Avantage

Résultat précis

Aucune valeur de comparaison





Partie Python

Localisation par Calcul Vectoriel

- Récupère la puissance selon l'adresse et l'heure
- Opération sur les vecteurs
- Classification des positions Étalonnage

Résultat basé sur l'étalonnage

Étalonnage le plus proche

Valeur de précision

Avantage

Valeur permettant de comparer

Connaître comment améliorer le score



Expérimentations

- Expérience 1 : L'expérience simulait un étalonnage fictif avec cinq capteurs dans une salle, pour localiser un téléphone mobile. Neuf positions de téléphone ont été préétablies, puis deux emplacements aléatoires ont été simulés. Les données des capteurs ont été enregistrées et un programme Python a été utilisé pour estimer les distances entre chaque capteur et le téléphone. Ces estimations ont permis de déduire une zone géographique probable pour le téléphone.
- Expérience 2 : Nos expériences ont été menées dans une salle dans sa configuration par défaut. Un ordinateur a été utilisé à tour de rôle comme "capteur", tandis qu'un téléphone mobile devait être localisé. Les capteurs étaient positionnés aux quatre coins de la salle. En plus de cela, nous avons réalisé des mesures sans obstacle, en variant l'inclinaison à la fois du téléphone et du capteur, afin d'explorer différentes conditions de fonctionnement.

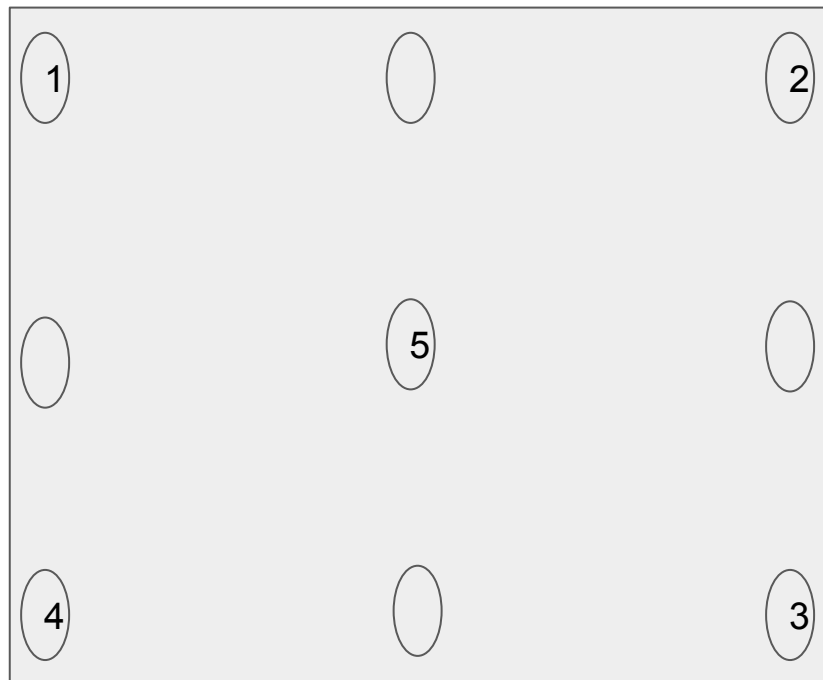
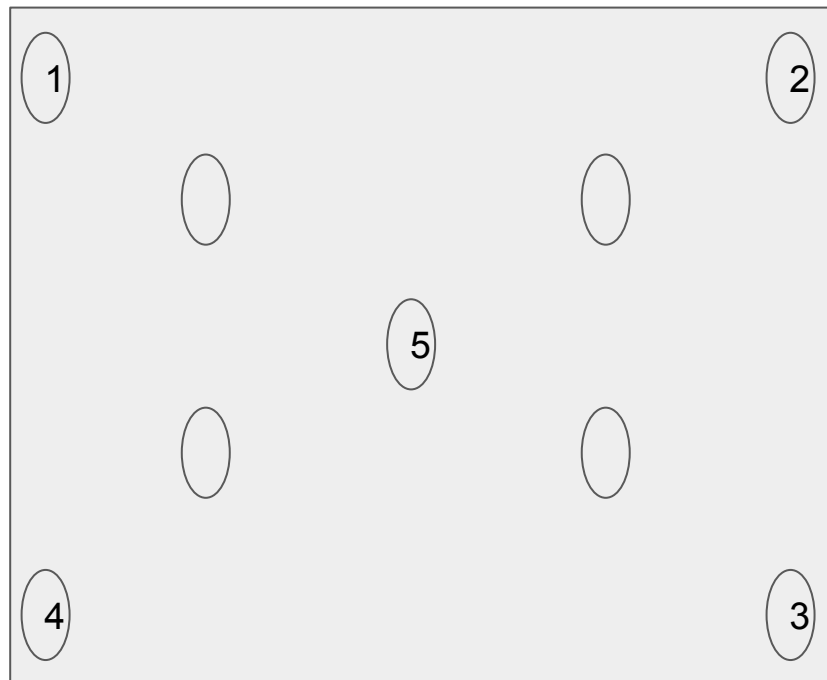


Expérimentations

- Expérience 3 : Nous avons équipé nos Raspberry Pi de dongles WiFi pour créer un réseau de capteurs. Après avoir installé les logiciels nécessaires, nous avons configuré les dongles en mode monitor. Les capteurs enregistrent les données et les envoient à une unité centrale. Les données sont traitées pour créer un fichier CSV indiquant le numéro du capteur, l'heure, la puissance du signal et les adresses MAC détectées. Cela nous permet de localiser approximativement un téléphone en fonction de sa proximité avec les capteurs.



croix vs carré





Conclusion

Malheureusement les conditions et la configuration de la DO2 nous ont pas permis de pouvoir faire nos expérimentations correctement. Nous nous sommes rabattus sur la salle CURI 306 où nous sommes souvent. Dans la salle CURI, nous avons pu faire nos expérimentations au mieux.

Au niveau des résultats de nos expériences nous pouvons en déduire :

- Plus on a de capteur, meilleur sera la précision
- Avec des capteurs plus performants, on aura une meilleure puissance de calcul
- Plus on a d'étalonnage, meilleur sera la précision

Cela demande plus de temps d'installation et donc plus de puissance de calcul (négligeable)

Merci de votre attention !

Démonstration

