

---

---

# INFO-F308 : PROJET D'ANNÉE

## OWL

### OPENWIFILocalizator

---

---

25 octobre 2016

Rémy Detobel - 000408013  
Denis Hoornaert - 000413326  
Nathan Licardo - 000408278  
Robin Petit - 000408282

## 1 Introduction

Avant d'entrer dans les détails concernant l'implémentation de notre projet, il est intéressant d'introduire, dans les grandes lignes, l'objectif de réalisation du projet mais également le but final de ce dernier.

Nous avons donc décidé, dans le cadre du cours Info-f308, de réaliser une application permettant à tout étudiant (ou visiteur) se trouvant sur l'un des campus de l'ULB de pouvoir se diriger au sein de ce dernier. L'idée principale de notre projet est de pouvoir rediriger un nouvel étudiant (ou un individu ne connaissant pas le campus sur lequel il se trouve) vers un auditoire cible. Afin de pouvoir localiser avec précision l'individu en question, nous sommes parti de l'idée de la triangulation. Cette technique est notamment utilisée dans les GPS. Les détails concernant l'implémentation de notre système seront donnés dans la suite de ce rapport.

L'objectif ultime de ce projet sera de présenter notre travail à un large public. Il sera donc intéressant de « vulgariser » notre programme et de montrer une application concrète. Pour ce faire nous développerons une application mobile (cf partie Implémentation).

Afin de réaliser notre projet nous avons eu besoin d'informations précises concernant l'ULB. Nous tenons donc à remercier M. Bruno Delcroix qui nous a permis d'accéder aux plans des campus concernés et donc de mettre en place notre application.

## 2 Implémentation

Dans le cadre de ce projet nous avons donc décidé de développer une application permettant aux étudiants i"perdus" de se géolocaliser au sein d'un campus directement via une application mobile. Il nous était donc nécessaire de choisir une plateforme intéressante pour porter notre application. Le système d'exploitation retenu a été, dans un premier temps, le système Android. La raison de notre choix repose principalement sur l'aspect « ouvert » de la plate-forme. Il nous est en effet possible d'accéder aux données liées aux réseaux Wi-Fi.

L'application sera donc développée en Java (langage Android). Afin de localiser l'utilisateur au sein du campus de manière précise nous avons décidé d'utiliser les bornes Wi-Fi comme expliqué précédemment. Il nous fallait donc avoir accès aux propriétés des bornes Wi-Fi au sein de l'application. Pour ce faire on utilisera l'API *WifiManager* qui nous permet d'obtenir les différentes informations concernant les réseaux Wi-Fi connectés et les différents réseaux accessibles.

Nous avons dans un premier temps choisi de développer sur Android. Ce n'est pas pour cela que nous nous fermons la possibilité de porter le programme sur iOS. Il est toutefois important de tenir compte du fait que l'accès aux données de l'utilisateur sont beaucoup plus complexes et limitées sur iOS. L'API *WifiManager* n'existe pas non plus, et nous n'en avons pas d'équivalent. Le développement sur la plateforme d'Apple fait donc partie des éléments que nous nous réservons de faire uniquement dans le cas où l'on aurait assez de temps pour le faire correctement.

## 3 Wi-Fi

Avant de décrire le fonctionnement et l'origine du Wi-Fi, voici comment le Larousse le définit : « Réseau local hertzien (sans fil) à haut débit destiné aux liaisons d'équipements informatiques dans un cadre domestique ou professionnel. » Le terme « Wi-Fi » vient de : « Wireless Fidelity » (qui est en opposition à « Hi-Fi » qui est généralement associé lui aux appareils sonores).

C'est en 1990 qu'une première connexion est établie entre un ordinateur et un serveur. Il ne s'agit là que d'une première étape dans le *World Wide Web*. C'est entre 1997 et 2000 que va réellement se développer le Wi-Fi. Un standard concernant la technologie sans fil est défini (par les laboratoires Bell et l'*Institute of Electrical and Electronics Engineers*). Depuis 2009, de plus en plus d'entreprise proposent un service de Wi-Fi gratuitement à leurs clients. On peut le retrouver dans les restaurants, les trains, les aéroports, etc.

### 3.1 Méthodes de localisation

Il existe plusieurs méthodes pour se repérer à partir des points d'accès Wi-Fi.

#### 3.1.1 Méthode dites par propagation

Cette méthode utilise la trilatération pour déterminer la position actuelle. On peut décomposer cette méthode en plusieurs étapes :

1. on collecte la qualité de signal d'au moins trois points d'accès wifi ;
2. pour chacun de ces points d'accès, on détermine la distance à celui-ci via une formule prenant en compte la qualité du signal ;

3. à partir de ces distances, on peut « générer » trois cercles et garder un point contenu dans l'intersection de ces cercles<sup>1</sup>.

**Détermination de la distance** La formule utilisée pour la détermination de la distance est la suivante :

$$d_f(s) = 10^{\alpha_f(s)},$$

où :

$$\alpha_f(s) = \frac{27,55 - 20 \log_{10}(f) + |s|}{20},$$

avec :

$f$  est la fréquence (généralement 2.4Ghz ou 5.0 GHz) ;

$s$  est la qualité du signal (mesuré en  $dBm$ ).

**Trilateration** Soient  $(x, y) \in \mathbb{R}^2$  la position actuelle de l'utilisateur,  $x_\lambda$ ,  $y_\lambda$  et  $r_\lambda$  (où  $1 \leq \lambda \leq 3$ ) est le position et la distance de chaque émetteur. Alors on peut écrire :

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2, \quad (1)$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2, \quad (2)$$

$$(x - x_3)^2 + (y - y_3)^2 = r_3^2. \quad (3)$$

On développe les produits remarquables :

$$x^2 - 2xx_1 + x_1^2 + y^2 - 2yy_1 + y_1^2 = r_1^2 \quad (4)$$

$$x^2 - 2xx_2 + x_2^2 + y^2 - 2yy_2 + y_2^2 = r_2^2 \quad (5)$$

$$x^2 - 2xx_3 + x_3^2 + y^2 - 2yy_3 + y_3^2 = r_3^2 \quad (6)$$

On soustrait (4) à (5) ainsi que (5) à (6) :

$$\begin{cases} -2xx_1 + 2xx_2 + x_1^2 - x_2^2 - 2yy_1 + 2yy_2 + y_1^2 - y_2^2 = r_1^2 - r_2^2 \\ -2xx_2 + 2xx_3 + x_2^2 - x_3^2 - 2yy_2 + 2yy_3 + y_2^2 - y_3^2 = r_2^2 - r_3^2 \end{cases}$$

$$\begin{cases} -2xx_1 + 2xx_2 - 2yy_1 + 2yy_2 = r_1^2 - r_2^2 - y_1^2 + y_2^2 - x_1^2 + x_2^2 \\ -2xx_2 + 2xx_3 - 2yy_2 + 2yy_3 = r_2^2 - r_3^2 - y_2^2 + y_3^2 - x_2^2 + x_3^2 \end{cases}$$

$$\begin{cases} x(2x_2 - 2x_1) + y(2y_2 - 2y_1) = r_1^2 - r_2^2 - y_1^2 + y_2^2 - x_1^2 + x_2^2 \\ x(2x_3 - 2x_2) + y(2y_3 - 2y_2) = r_2^2 - r_3^2 - y_2^2 + y_3^2 - x_2^2 + x_3^2 \end{cases}$$

On remplace ces valeurs constantes par des symboles plus *parlant*.

$$\begin{cases} Ax + By = C \\ Dx + Ey = F \end{cases}$$

---

1. Mathématiquement, trois cercles s'intersectent en un et un seul point. Cependant, toutes interférences prises en comptes, on peut visualiser les cercles comme ayant une *épaisseur* correspondant à l'incertitude. On peut alors parler d'un *point de l'intersection des trois cercles*.

On développe d'abord pour  $x$  :

$$x = \frac{C - B(\frac{F-Dx}{E})}{A}$$

$$x = \frac{CE - BF + BDx}{EA}$$

$$EAx = CE - BF + BDx$$

$$x = \frac{CE - BF}{EA - BD}$$

On développe ensuite pour  $y$  :

$$y = \frac{F - D(\frac{C-By}{A})}{E}$$

$$y = \frac{FA - DC + DBy}{AE}$$

$$AEy = FA - DC + DBy$$

$$y = \frac{CE - BF}{EA - BD}$$

#### Avantages :

- Rapidité de mise en place.

#### Désavantages :

- Difficulté de mise à jour de la base de donnée ;
- la qualité du signal peut fortement varier en fonction de :
  - la quantité de personnes présentes sur le point d'accès ;
  - divers éléments externes tels que des murs, etc.

## 4 Traitement du graphe

### 4.1 Utilisation de la triangulation

Le graphe manipulé sera représenté par  $\Gamma = (V, E)$ , avec  $V$  l'ensemble des locaux accessibles ainsi que l'ensemble des points additionnels dans les couloirs afin de relier les locaux entre eux, et  $E$  l'ensemble des liens permettant de joindre les nœuds.

L'idée initiale était de lier le graphe des nœuds avec un graphe des bornes Wi-Fi et de diminuer les poids des arêtes d'une constante  $K \geq 0$  pour les arêtes ayant une borne à une des deux extrémités. Ce cas aurait laissé la possibilité d'avoir des arêtes de poids négatif, et donc un algorithme classique tel que Dijkstra ou A\* n'aurait pas été possible.

Nous avons donc pensé qu'un algorithme plus général tel que *Bellman-Ford* serait nécessaire (afin de gérer les poids négatifs), et pour pouvoir diminuer les calculs, faire une passe de *Floyd-Warshall* au préalable afin de pouvoir élaguer tous les chemins en cours de calcul, de poids supérieur à la borne donnée par Floyd-Warshall.

## 4.2 Utilisation de la SSM

Cependant, le fait d'utiliser une SSM rend la localisation plus simple : il n'est plus nécessaire de maintenir un graphe des bornes, et donc il n'y a plus qu'un seul graphe, à savoir celui des locaux (dans lequel seront stockées les informations relatives aux réseaux). Cela retire la nécessité de retirer une constante  $K$  et donc, il n'y a plus de risque d'avoir des arêtes de poids négatif.

Dès lors, un algorithme standard tel que Dijkstra ou  $A^*$  peut être utilisé. C'est  $A^*$  qui sera implémenté ici car il est de complexité  $O(|E|)$  dans le pire des cas, alors que Dijkstra est de complexité  $O(|E| + |V| \log |V|)$  dans le pire des cas.

Le gain est permis par l'utilisation d'une heuristique qui nous suffira amplement ici.