

Rapport de projet

De Master Informatique

Sujet 52 :
Recherche de candidats/jobs sans contact



Etudiants : Maxime Gens Camille Riquier	Rapport de projet C.V sans contact	Tuteurs : Nabil Djarallah Nicolas Haderer Romain Rouvoy
--	---	---

Introduction

Android est un langage offrant une multitude de possibilité et ce sur tout les plans. Mais il reste encore beaucoup à faire pour qu'Android puisse être bénéfique à l'ensemble de la société : comme par exemple : à la recherche d'emploi. En effet, ce système pourrait peut être permettre à une personne lambda de trouver l'emploi de sa vie, et ce rien qu'en se promenant.

Dans ce rapport, nous aborderons un projet Android lié à une nouvelle façon de faire de la recherche de l'emploi et réalisé durant la première année de master informatique à Lille pour la matière de PJI.

Dans un premier temps nous présenterons le projet ainsi que son déroulement durant le semestre puis nous commenterons les analyses sur les besoins de même que les différentes façons de procéder pour le développement, puis nous finirons sur les apports qu'un tel projet a eu sur nous et enfin les évolutions possibles de notre librairie.

Sommaire

Introduction.....	3
I. Présentation	5
1. Présentation du sujet	5
2. Le Choix du sujet	6
3. Le Cadre de travail.....	6
II. L'Analyse du projet.....	7
1. Problématique.....	7
2. Protocole de Communication.....	8
3. Choix des technologies.....	9
4. Diagramme UML de l'application.....	10
III. Développement	11
1. Bluetooth.....	11
2. Wifi – Direct.....	12
3. Aperçu de l'application.....	13
IV. Apports et difficultés.....	17
1. Apports.....	17
2. Difficultés rencontrées	17
V. Evolution	18
VI. Remerciements	18
VII. Conclusion	19
VIII. Annexe	20
1. Références.....	20
2. Diagramme UML.....	21
3. Etat de l'art.....	22

I. Présentation

1. Présentation du sujet

L'application Android que nous avons développée est une application qui permet l'échange de données entre utilisateurs par le biais des technologies sans fil sans avoir recours à internet. Un utilisateur, A, échangera des données avec un utilisateur, B, simplement lorsque les deux terminaux Android se croiseront et sans intervention de la part de A et B.

Exemple concret :

Un employeur d'une entreprise recherche un développeur JAVA avec 5 ans d'expérience. Il va enregistrer l'offre d'emploi sur son téléphone et la sélectionner dans l'application, puis il va saisir divers informations et compétences requises pour le poste (ainsi qu'un numéro de téléphone ou mail pour le joindre). C'est tout, il valide et met son téléphone dans sa poche.

Un demandeur d'emploi va de son côté indiquer à l'application qu'il recherche du travail dans le développement JAVA et qu'il possède 6 ans d'expérience dans ce domaine, il va enregistrer son CV dans l'application et simplement valider. Puis il va mettre son téléphone dans son sac.

Chacun va se balader, prendre le métro, traverser un passage piéton, assister à des salons, conférences, bref vaquer à ses occupations, dès que les deux terminaux Android vont se croiser, et si les deux recherches correspondent, alors les deux téléphones vont échanger entre eux l'offre d'emploi et le CV ; ainsi l'employeur pourra avoir directement le CV d'une personne qu'il aura croisé dans la rue ou ailleurs et le demandeur d'emploi aura lui l'offre d'emploi d'un employeur qu'il aura croisé récemment.



2. Le Choix du sujet

Parmi la palette des projets proposés par le corps enseignant ainsi que l'INRIA et les entreprises extérieures, nous avons choisi le projet CV sans contact, mais pourquoi ce projet ?

Tout d'abord, nous voulions trouver un projet concernant un marché porteur. En effet le marché des applications mobiles est plus qu'important à l'heure actuel. C'est là qu'Android est devenu pour nous une évidence.

Le projet CV sans contact nous a paru le choix le plus approprié car il nous semblait plus novateur que les autres sur le plan utilisation dans la vie de tous les jours.

3. Le Cadre de travail

Travail INRIA bureau, badge, logiciel utilisé, recherche faites.

Afin de mener à bien notre projet, nous avons besoin de matériels. Notre application fonctionnant sous terminaux Android, ils nous faillaient donc des terminaux de tests afin de vérifier le bon fonctionnement de notre application. Nous sommes donc venus travailler plusieurs fois dans les locaux de l'INRIA (cf. annexe 1) afin d'avoir accès à des terminaux Android.



D'un point de vue plus technique, pour développer notre application nous avons utilisé l'IDE (cf. annexe 1) **Eclipse**, qui grâce à son plugin ADT (cf. annexe 1), nous a permis d'implémenter notre projet.

Nous avons aussi utilisé un S.V.N (cf. annexe 1) en particulier celui de Google (Google Code). Ceci pour pouvoir travailler ensemble sur le même projet, de récupérer directement et facilement les modifications de chacun.

De plus nous travaillons sur une version d'Android étant au minimum 4.0 à cause des technologies que nous détaillerons plus tard.

II. L'Analyse du projet

Nous verrons dans cette partie toutes les conclusions que nos analyses nous ont amenées à faire. Dans un premier temps nous verrons la problématique globale qui se dégage du projet, puis nous verrons des fragments de réponse à cette problématique.

1. Problématique

Pour commencer correctement ce projet nous avons dû faire un tour des technologies que l'on pourrait exploiter pour notre futur application (cf. annexe 3), afin de choisir la ou les meilleures technologies utilisable pour notre projet.

Puis avant d'implémenter notre application, nous avons réfléchi sur les fonctionnalités, les besoins et les habitudes de l'utilisateur concernant les applications mobiles mais également sur les problèmes survenant lors de transfert de fichier.

Après cette période de réflexion, une problématique s'est dégagée :

« Comment communiquer et échanger des informations entre terminaux Android de façon transparente pour l'utilisateur ? »

2. Protocole de Communication

Une des principales problématiques de ce projet est la communication entre deux systèmes Android. En effet, il fallait que celle-ci soit brève, complète et optimisée. Pour ce faire, nous avons créé un protocole de communication qui assure que seuls les systèmes concernés par une recherche y répondent (filtrage), en cas d'interruption de connexion un minimum d'information aura été transmis pour permettre un futur contact (présentation) et un envoi de fichier si le temps de transfert de la technologie utilisé pour la communication et le temps de croisement des deux systèmes le permet.

Schéma lorsque les deux devices ont des échanges à faire.

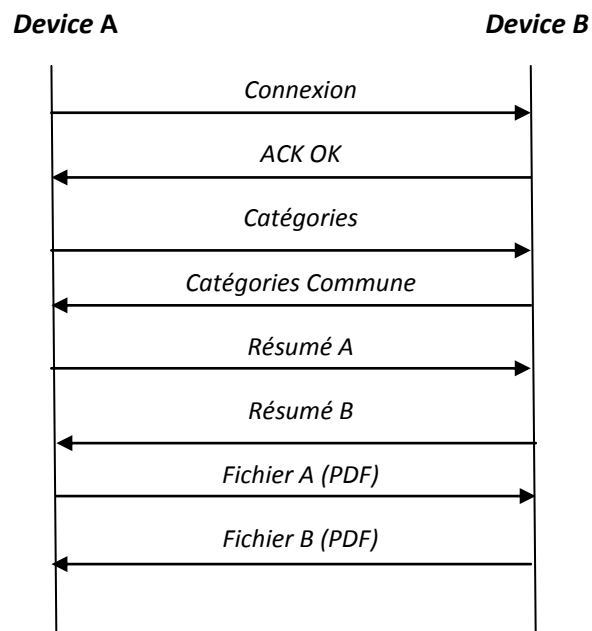
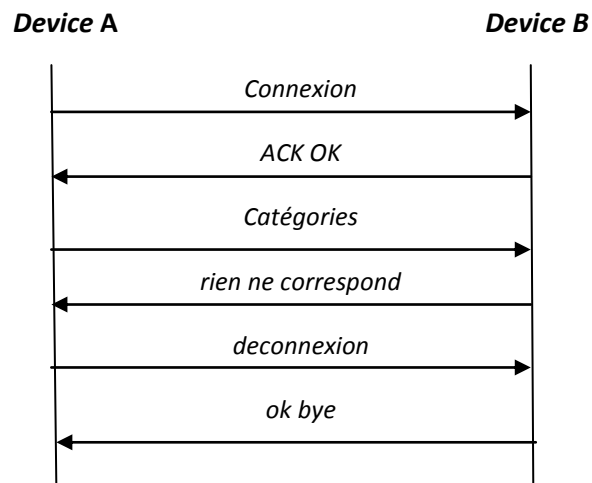


Schéma lorsqu'il n'y a pas d'échange possible(pas de concordance dans les catégories rechercher)



Le Filtrage

Nous voulions éviter un maximum de traitement suite aux différentes communications qui seront créés au fil des journées d'utilisation de notre application. C'est pourquoi, nous avons décidé de mettre en place un filtre le plus tôt possible dans l'échange de données. Ce filtrage se base sur les recherches du téléphone demandant la communication, que ce soit en offres ou en demandes.

La réponse a ce filtrage peut être soit désolé je n'ai rien qui corresponds à ce que tu recherches et qui met fin à l'échange, soit l'ensemble des réponses correspondantes.

La Présentation

Il est fort probable que l'échange n'arrive pas à son terme à cause principalement des systèmes qui disparaissent de la zone couverte par la technologie utilisée du terminal qui recherche des réponses. Pour empêcher que la communication ne soit pas fructueuse, un résumé sera envoyé en pré-réponse à la recherche. Celui-ci comportera les principales informations de la réponse.

3. Choix des technologies

Pour réaliser notre projet, il fallait dans un premier temps choisir les technologies qui seront utilisées par notre application lors de l'échange de donnée. Pour ce faire nous avons réalisé un état de l'art(cf : annexe 2) sur les technologies de communication qu'utilise un terminal Android. Les technologies devaient correspondre aux critères suivants : pas d'utilisation d'internet, consommation minimum d'énergie et débit de transfert assez important.

En fonction des renseignements que nous avons trouvés et mis en page. Trois technologies nous ont semblé correspondre : Le Bluetooth, le Wifi direct et le NFC.

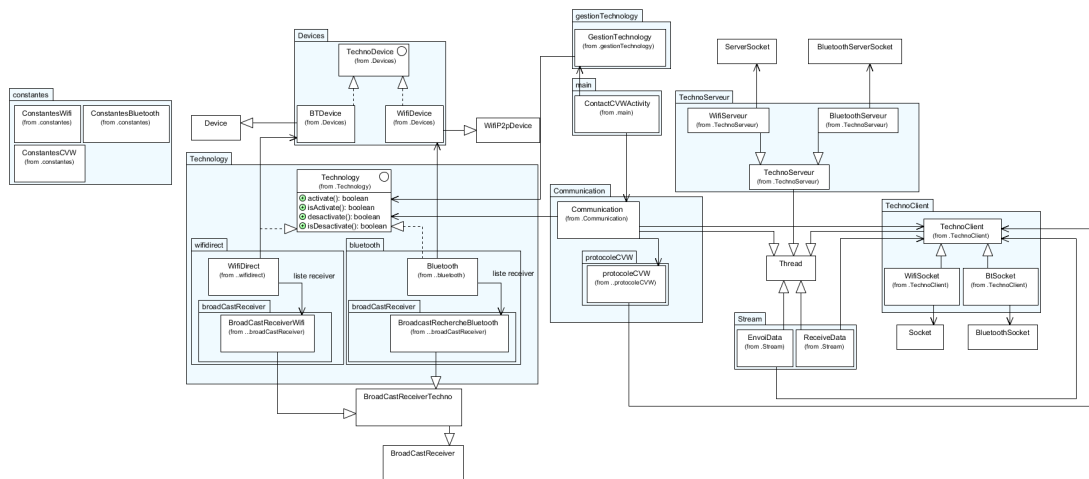
La sélection finale a été faite sur la portée de chaque technologie. Plus la portée est importante plus la masse de terminal ayant une réponse susceptible de correspondre à une recherche effectuée est importante. Par ce fait, le bluetooth et le wifi direct qui ont une portée largement supérieur au NFC ont été sélectionnées pour notre application.

Après la lecture de plusieurs études concernant les outils d'Android, il a été dit que plusieurs millions de systèmes Android posséderont d'ici 3 ans le Wifi Direct et le bluetooth « *low Energy* » (version 4.0 de bluetooth) dit « *BLE* », cela nous a conforté dans l'idée que ces technologies seront plus qu'utilisées dans le futur.

4. Diagramme UML de l'application

Deux digrammes globaux

➤ Gestion des technologies



Voir annexe 2 pour un affichage plus grand

Les principaux problèmes de la généralisation furent la gestion des éléments différents des technologies.

Chaque technologie possède un type de device, de broadcastReceiver, de serverSocket, et de socket différent. Généralisé les technologies revenait à généralisé l'ensemble des éléments leur étant associé. Pour cela la création d'un technoServer, technoClient, device et broadCastReceiver ont été nécessaire pour abstraire les classes lié aux technologies.

Un manager de technologie est là pour permettre l'utilisation de n'importe quelle technologie voulue.

Pour pouvoir communiquer entre deux périphériques, il est préférable de passer par la classe communication (package « communication ») qui une fois lancé, permet l'exécution du protocole de communication. Celui permet l'envoi et la réception de message qui sont ensuite traduit par l'objet communication.

Il y a également des constantes associées à chaque type de technologies, ainsi que des constantes générale symboliser par le package « constantes »

Pourquoi généralisation ?

Une abstraction a été faite pour permettre l'ajout de technologie plus innovante, ou moderne, et ce sans modifier l'algorithme global de l'application.

III. Développement

Dans cette partie nous avons détaillé principalement le fonctionnement des différentes technologies que nous avons implémenté (Bluetooth et WIFI-Direct) ainsi le fonctionnement général de l'application.

Pour le développement nous avons évidemment commenté notre code afin de faciliter la reprise ultérieure de notre code par d'autres développeurs.

1. Bluetooth

Le Bluetooth est une technologie très répandue aujourd'hui, pratiquement tous les terminaux mobiles le possèdent et il s'agit surtout d'une technologie devenue familière aux utilisateurs, de cette manière l'utilisation de cette technologie permettait de ne pas déstabiliser le futur utilisateur de notre application.



Bien que l'emploi dans la pratique du Bluetooth soit relativement simple, l'implémentation en est autre chose, en tout cas pour notre part, la prise en main à été difficile au début du projet, et il n'y a encore quelque « zones d'ombres » dans le code, principalement pour l'appareillage des appareils.

La librairie Bluetooth est fonctionnelle mais à utiliser en ayant des bases. En effet l'utilisation de cette librairie implique avoir des notions dans le domaine des réseaux informatique, l'emploi d'objet *Socket* et *SocketServer* y est prédominante (« *BluetoothSocket* » et « *BluetoothSocketServer* »). Ces sockets permettent en outre la création d'un canal de liaison entre deux terminaux mobiles afin d'échanger des informations, à conditions évidemment que les deux téléphones soient préalablement appariée. Ce dernier point à poser plusieurs problèmes car le fait que les deux téléphones doivent être appareillé restreint beaucoup l'utilisation de l'application, pour le moment aucun solution concrète n'a était trouvé, si ce n'est de contourner les normes de sécurité du Bluetooth pour « forcer » automatique l'appareillage.

Autres domaines abordé dans l'implémentation du Bluetooth, les « *BroadcastReceiver* », permettent « d'écouter » les modifications effectuées sur le terminal Android, ceux-ci sont principalement utilisés afin de surveiller l'activation ou la désactivation du Bluetooth par l'utilisateur. Ils sont aussi utilisés afin de rechercher des périphériques Bluetooth aux alentours (c'est la fonction de la classe « *BroadcastRechercheBluetooth* » qui recherche s'il y a de nouveaux terminaux Bluetooth disponible aux alentours.

D'un point de vue plus technique voici la description de chaque classe de la librairie Bluetooth :

Bluetooth : Il s'agit de la classe « principale » du Bluetooth, celle qu'il faudra instancier afin de pouvoir contrôler le Bluetooth et d'interagir avec

BroadcastRechercheBluetooth : Cette classe, qui extends la classe *Broadcast*, permet de lancer une recherche des périphériques Bluetooth aux alentours.

2. Wifi – Direct

Définition Wi-Fi-Direct

Le Wifi direct est une technologie qui est sur les terminaux Android à partir de la version 4.0. Celle-ci permet l'échange de données d'un système Android à un autre sans passer par internet. Elle a un fonctionnement similaire au Bluetooth mais en moins sécurisé. (pour plus de détails voir annexe 1)

Fonctionnement

L'utilisation du wifi direct est assez simple comme : la recherche, la connexion... et se fait par le biais d'un manager et de fonction ayant en paramètre des listeners.

Comme le Bluetooth, il faut passer par plusieurs étapes pour permettre l'échange de données : la recherche de périphériques, la connexion a ces périphériques.

Pour réaliser la communication entre terminaux Android grâce au Wifi Direct, il s'agit juste d'un serveur TCP/IP et d'un socket TCP/IP.

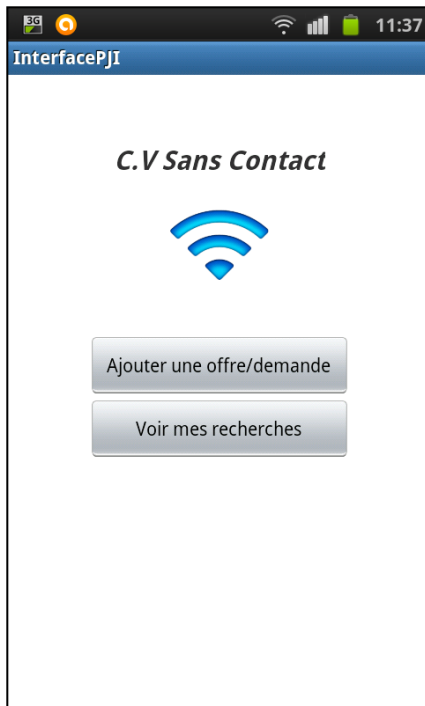
Les listeners utilisés permettent de récupérer des données du téléphone pour notre application : exemple : celui de la recherche permet de récupérer la liste de terminaux détecté par le téléphone.

Et bien évidemment, l'utilisation d'un broadcast receiver a été nécessaire pour intercepter les messages concernant le wifi Direct du système pour exécuter un code spécifique pour chaque message.

3. Aperçu de l'application

Voici plusieurs capture d'écran permettant d'avoir une idée générale de l'apparence de l'application, bien évidemment le design est très sommaire, ces capture d'écran permettent surtout d'avoir une plateforme de test pour le développement, avec quand même une première ligne directrice de l'apparence finale du projet.

Accueil



Il s'agit de la page de lancement de l'application, c'est page d'apparence simple visuellement est beaucoup plus lourde que ce qu'elle n'y paraît car au démarrage de l'application (si il s'agit du tout premier lancement de l'application sur le téléphone) va créer une base de données sur le terminal Android afin de pouvoir y stocker toutes les informations nécessaire (comme mémoriser les différentes recherches d'employé ou de travail de l'utilisateur.

Elle comporte deux boutons relativement clairs pour l'utilisateur. La possibilité d'ajouter une nouvelle recherche à l'application et un bouton permettant de visualiser les recherches en cours.



Le menu de cette page d'accueil contient pour le moment deux items, un item permettant simplement de quitter l'application et un autre item permettant d'arrêter immédiatement la recherche sans fils en cours et donc d'économiser de la batterie ou simplement si l'utilisateur ne souhaite pas échanger avec les autres terminaux Android.

Ajouter une offre/demande

InterfacePJI

Ajouter une recherche

Type de recherche

☒ Offre ☐ Demande

Choisissez une categories

Informatique

Indiquez le poste recherché

Chef de projet

PDF : Le PDF doit être placé à la racine de la carte SD pour le moment (TEST) nommé monCV.pdf ou monOffre.pdf

Résumé

☒ Automatique ☐ Manuel

InterfacePJI

Type de recherche

☒ Offre ☐ Demande

Choisissez une categories

Informatique

Indiquez le poste recherché

Chef de projet

PDF : Le PDF doit être placé à la racine de la carte SD pour le moment (TEST) nommé monCV.pdf ou monOffre.pdf

Résumé

☒ Automatique ☐ Manuel

Valider la recherche

InterfacePJI

Ajouter une recherche

Type de recherche

☒ Offre ☐ Demande

Choisissez une categories

Informatique

Indiquez le poste recherché

Chef de projet

PDF : Le PDF doit être placé à la racine de la carte SD pour le moment (TEST) nommé monCV.pdf ou monOffre.pdf

Résumé

☒ Automatique ☐ Manuel

Voir mes recherches

La page permettant d'ajouter une offre ou une demande est une simple page d'ajout, elle permet d'enregistrer dans la base de données une nouvelle offre ou demande que l'application prendra en compte lors des échanges avec les autres téléphones

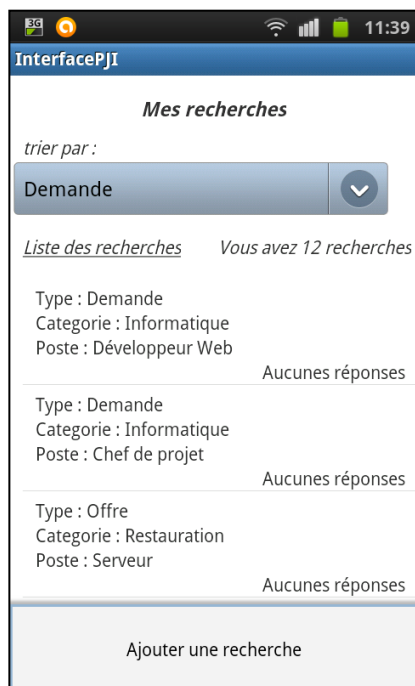
L'ajout d'un menu permettant de se rediriger directement vers la page de visualisation permet une meilleure ergonomie en évitant ainsi de devoir repasser par la page d'accueil.

Voir mes recherches



La page des recherches est une liste de toutes les recherches en cours, sur cette page l'utilisateur peut consulter les recherches qu'il a lancées et voir s'il y a eu des « réponses » à ses recherches. Il peut aussi trier par « Demande » ou par « Offre » ses recherches.

Un appuie long sur une recherche envoie vers la page « modifier offre/recherche » et un appui simple ouvre la page « voir les réponses »



Bien évidemment un menu a été ajouté à cette page permettant de pouvoir ajouter directement une recherche sans avoir à repasser par la page d'accueil.

Modifier offre/demande

The screenshot shows the 'InterfacePJI' application on an Android device. The screen is titled 'Détails de la recherche' and contains the following elements:

- A status bar at the top showing the time as 11:38 and various icons.
- A header bar with the text 'InterfacePJI'.
- A sub-header 'Détails de la recherche' with a note: 'Sur cette page vous pouvez modifier ou supprimer la recherche'.
- A section 'Type de recherche' with two radio buttons: 'Offre' (selected) and 'Demande'.
- A section 'Choisissez une categories' with a dropdown menu showing 'Restauration'.
- A section 'Indiquez le poste recherché' with a dropdown menu showing 'Serveur'.
- A button labeled 'Voir le PDF'.
- A section 'Résumé' with two radio buttons: 'Automatique' and 'Manuel' (selected).
- At the bottom, two buttons: 'Valider modification' and 'Supprimer la recherche'.

Un appuie long sur une recherche permet d'ouvrir un nouvelle page avec toutes les caractéristiques de la recherche en cours afin de pouvoir la modifier si on le souhaite.

Voir les réponses

The screenshot shows the 'InterfacePJI' application on an Android device. The screen is titled 'Les réponses à votre recherche' and contains the following elements:

- A status bar at the top showing the time as 11:38 and various icons.
- A header bar with the text 'InterfacePJI'.
- A sub-header 'Les réponses à votre recherche' with a note: 'Vous avez 7 reponses'.
- A section 'Liste des réponses' with a list of responses. Each response is represented by a text label (e.g., 'Voir la réponse n°0') and a PDF icon.
- At the bottom, a button labeled 'PDF'.

Lorsqu' on appuie et donc sélection une recherche dans la liste de nos recherches, une nouvelle page s'ouvre avec la liste de toutes les réponses à cette recherche, c'est-à-dire le résultat de tous les terminaux Android rencontrés qui correspondent à notre recherche. Le fait de cliquer sur une réponse va ouvrir directement le PDF associé à cette réponse (ou le résumé si l'échange n'a pas correctement pu transmettre tout le PDF).

IV. Apports et difficultés

1. Apports

Ce projet nous permet de nous familiariser un peu plus avec le monde hors université, ceci afin de nous rendre autonome et de développer notre façon de travailler en équipe ainsi que dans un milieu plus professionnel.

Nous avons pu également approfondir nos connaissances sur Android ainsi que l'utilisation de technologies sans fils, en particulier le Bluetooth et le Wifi-Direct.

2. Difficultés rencontrées

Le wifi direct est une technologie assez récente donc peu de document traitant de celle-ci en globalité ou la documentation est éparpillé un peu partout et il y manquait également des documents sur certains éléments comme l'activation de cette technologie.

En ce qui concerne le Bluetooth, une des deux principales difficultés réside dans le fait que lorsqu'on effectue une recherche de périphériques aux alentours, nous sommes obligé d'attendre la fin de la recherche pour avoir accès à la liste des nouveaux périphériques. Cela est dû au *Broadcast Receiver* qu'utilise Android pour le Bluetooth, je n'ai pas réussi à contourner ce problème ou du moins je n'ai pas trouvé de solution optimale sans alourdir considérablement le code, il est donc préférable d'afficher manuellement (en cliquant sur un bouton par exemple) la liste des périphériques plutôt que de les faire s'afficher à la suite.

Un second problème assez embêtant concerne la sécurité Bluetooth, dans notre application nous devons envoyer des données de façon transparente à l'utilisateur, seulement la sécurité Bluetooth oblige l'utilisateur à autoriser une connexion sur son téléphone, c'est le problème de l'appareillage, il n'est pas possible d'appareiller automatique deux téléphones entre eux par Bluetooth sans l'accord du l'utilisateur. Les tests ont donc dû être effectué en ayant préalablement appareillé les téléphones de test.

V. Evolution

Afin d'améliorer l'application dans un avenir proche nous avons pensé à quelques améliorations qui selon nous apporteront plus d'ergonomie, facilité d'apprentissage et de reprise de code de l'application.

Coté utilisateur

Dans l'application même nous avons pensé à plusieurs améliorations :

- Possibilité d'ajouter un lien vers un site web ou vers son PDF stocké en ligne plutôt que de la récupérer directement sur la mémoire du téléphone.
- Amélioration de la gestion de la base de données (MVC et ajout des temps de chargement)

Coté développeur

Dans l'implémentation du code nous aimerions ajouter les fonctionnalités suivantes :

- Une meilleure gestion des erreurs
- Sélection automatique de la meilleure technologie disponible sur le téléphone.

VI. Remerciements

Pendant toute la durée de notre projet, plusieurs personnes nous ont été d'une aide précieuse.

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de *l'Université des Sciences et Technologies de Lille* pour avoir assuré la partie théorique de la formation du master Informatique.

Nous tenons à remercier aussi nos tuteurs *Nicolas Haderer* et *Romain Rouvroy* pour l'aide et les conseils qu'ils nous ont apporté durant notre projet.

Nous tenons à remercier aussi notre tuteur *Nabil Djaralla*, qui nous a apporté des conseils et de l'aide sur la réalisation du sujet, en particulier sur tout ce qui touche à la couche réseau de l'application, ainsi que sur le temps qu'il nous a consacré tout au long de cette période, sachant répondre à toutes nos interrogations.

VII. Conclusion

Notre but a été de développer une librairie permettant d'utiliser les différentes technologies de communication que possède un terminal Android simplement, permettant l'ajout futur de technologies.

Au début du projet, une étude assez importante pour la suite du projet a été menée. Celle-ci concernait les différentes possibilités de communication sans fil entre terminaux mobiles (cf : annexe 2 - état de l'art).

Les apports qui découlent de ce projet, nous ont permis d'améliorer ou découvrir le développement sur plateforme Android et des connaissances de technologies sans fil, bien que finalement notre projet a beaucoup plus porté sur l'implémentation de ces technologies que sur le développement d'une application Android.

L'informatique évolue très rapidement, il est sans cesse en évolution, et donc notre librairie peut s'adapter aux nouvelles technologies qui apparaîtront sur terminaux Android.

VIII. Annexe

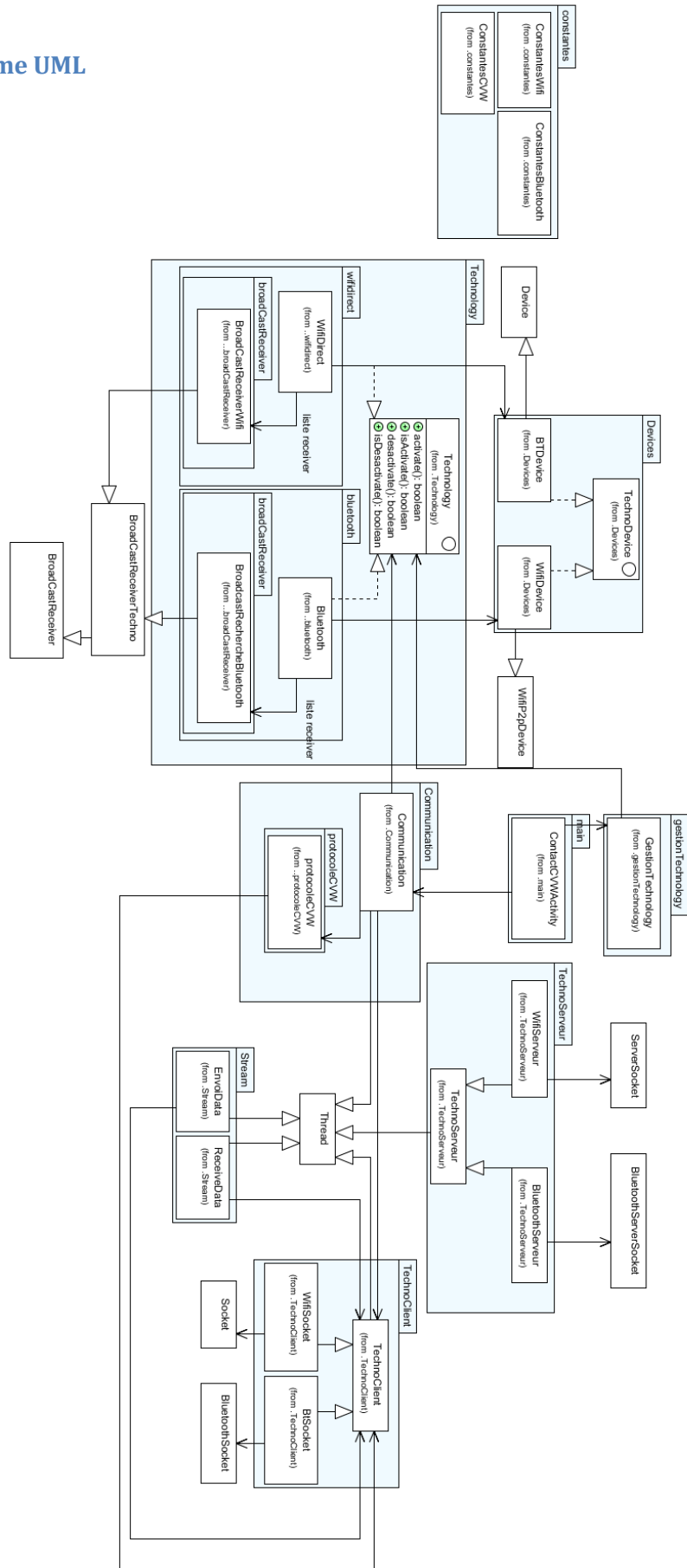
1. Références

ADT : Android Development Tools

IDE : Integrated development environment

INRIA : Institut national de recherche en informatique et en automatique

Rapport : CV sans contact



3. Etat de l'art

Projet individuel
Master Informatique
1ère année
2013



Etat de l'art

Les protocoles de communication sous Android



Etudiants : Maxime Gens Riquier Camille	Etat de l'art Protocoles de communication sous Android	Tuteurs : Nabil Djarallah Nicolas Haderer Romain Rouvoy
--	---	---

Sommaire

I. Introduction	24
II. Cadre	24
III. Résumé de l'application	25
IV. Analyse des versions d'Android	26
V. Les différents protocoles de communications.....	27
VI. Caractéristiques de chaque protocole de communication	27
1. Wi-Fi Direct.....	27
2. Wi-Fi DLNA :.....	28
3. Bluetooth.....	31
4. NFC	32
VII. Différence	33
VIII. Résultat et choix des technologies	33
IX. Protocoles et Version retenues	34
X. Sources	34

I. Introduction

La formation du master informatique de l'université des sciences et technologies de Lille, se décompose en quatre semestres répartis sur deux ans. Pendant le second semestre, il est demandé aux étudiants de réaliser un projet informatique.

Ce projet est la mise en pratique des connaissances apprises lors du cursus universitaire, il se termine par une soutenance orale et la rédaction d'un rapport à rendre à la fin. Ce projet nous permet de nous familiariser un peu plus avec le monde hors université, ceci afin de nous rendre autonome et de développer notre façon de travailler en équipe ainsi que dans un milieu plus professionnel.

Dans un premier temps nous expliquerons le fonctionnement et l'utilité d'application que nous devons développer, nous analyseront par la suite la répartition des différentes versions d'Android afin de choisir notre plateforme de développement, puis nous étudierons les différences qui existent entre les protocoles de communications afin de pouvoir sélectionner ceux qui s'adapteront le mieux à notre projet.

II. Cadre

Dans le cadre de notre master Informatique 1ère année, nous devons choisir un projet parmi une liste d'une centaine de projet Informatique. Ces projets proviennent à la fois de nos professeurs, d'entreprise extérieure à la faculté, de sujet de recherche, de sujet proposé par le Centre de recherche INRIA, c'est des sujets de ce centre de recherche qui a attiré notre attention.

Tous les domaines de l'informatique sont abordés. Souhaitant nous orienter dans le monde du développement mobile, nous avons donc retenu certains projets en rapport avec du développement Android car nous pensons que le développement sur plateforme mobile ouvre beaucoup plus de possibilité à l'innovation et à la créativité d'applications et de programmes nouveaux que le développement de programmes sur ordinateurs.

Après avoir consulté les différents projets Android, notre choix c'est porté sur une application utilisant les protocoles de communications sans-fil, ceci afin de découvrir le fonctionnement de ces technologies et de participer à un projet qui nous semble vraiment novateur (voir [résumé de l'application](#)).

III. Résumé de l'application

L'application Android que nous allons développer est une application d'échanges de données entre utilisateurs par le biais des technologies sans fil sans avoir recours à une connexion 3G ou Internet. Un utilisateur A échangera une certaine donnée avec un utilisateur B, si toutes les caractéristiques/conditions d'échanges sont satisfaites, simplement lorsque les deux terminaux Android se croiseront.

Exemple concret :

Un employeur d'une entreprise recherche un développeur JAVA avec 5 ans d'expérience. Il va enregistrer l'offre d'emploi sur son téléphone et la sélectionner dans l'application, puis il va saisir divers informations et compétences requises pour le poste (ainsi qu'un numéro de téléphone ou mail pour le joindre). C'est tout, il valide et met son téléphone dans sa poche.

Un demandeur d'emploi va de son côté indiquer à l'application qu'il recherche du travail dans le développement JAVA et qu'il possède 6 ans d'expérience dans ce domaine, il va enregistrer son CV dans l'application et simplement valider. Puis il va mettre son téléphone dans son sac.

Chacun va se balader, prendre le métro, traverser un passage piéton, assister à des salons, conférences, bref vaquer à ses occupations, dès que les deux terminaux Android vont se croiser, et si les deux recherches correspondent, alors les deux téléphones vont échanger entre eux l'offre d'emploi et le CV ; ainsi l'employeur pourra avoir directement le CV d'une personne qu'il aura croisé dans la rue ou ailleurs et le demandeur d'emploi aura lui l'offre d'emploi d'un employeur qu'il aura croisé récemment.

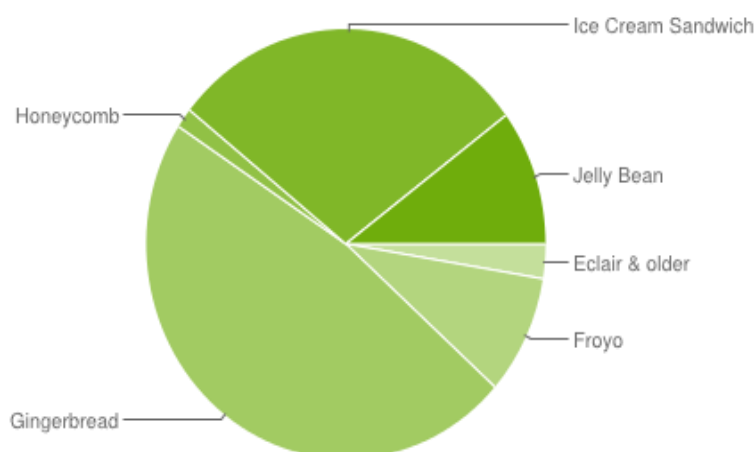


IV. Analyse des versions d'Android

Notre application doit permettre à deux terminal Android de pouvoir échanger en temps réel des données entre eux sans aucun intervention de l'utilisateur, les technologies sans fil sont donc nécessairement. De nombreuse solutions sans fil existent avec chacune ses avantages et inconvénients, cet état de l'art référencera et analysera toutes les technologies sans fil les plus appropriées pour notre application de communication.

A prendre en compte : la répartition des versions d'Android sur le marché

Version	Codename	API	Distribution
1.6	Donut	4	0.2%
2.1	Eclair	7	2.4%
2.2	Froyo	8	9.0%
2.3 - 2.3.2	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	47.4%
3.1		12	0.4%
3.2	Honeycomb	13	1.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	29.1%
4.1	Jelly Bean	16	9.0%
4.2		17	1.2%



<http://developer.android.com/about/dashboards/index.html>

Data collected during a 14-day period ending on January 3, 2013

- **Android Eclair et versions antérieures** sont passées de 3.1 à **2.6%** de possession
- **Android FroYo (2.2)** est passée de 10.3 à **9%** de possession
- **Android Gingerbread (2.3)** est tombée de 50.8 à **47.6%** de possession
- **Android Honeycomb (3.x)** est passée de 1.6 à **1.5%** de possession
- **Android Ice Cream Sandwich (4.0.x)** a grimpé de 27.5 à **29.1%** de possession
- **Android Jelly Bean (4.1/4.2)** est passée de 6.7 à **10.2%** de possession

Les versions les plus répandues sont donc :

- Android 2.3 Gingerbread : **47.6%**
- Android 4.0 Ice Cream Sandwich (4.0) : **29.1%**
- Android Jelly Bean (4.1) : **10.2%**

Android **Gingerbread** étant la plus répandue nous travaillerons donc avec celle-ci. Les dernières versions proposent plus de fonctionnalité que nous pourrions peut être utilisé (Bluetooth 4.0) mais cela serait des fonctions non primordiales à l'utilisation standards de l'application.

La progression des dernières versions d'Android (à partir de 4.0) est relativement importante (39.3%), il pourrait être intéressant de réfléchir à une futur transition vers une des dernières versions d'Android. Afin de cibler le maximum de terminaux Android et de posséder les dernières technologies(voir IV – 3)nous avons choisi la version **Ice Cream Sandwich 4.0 API 15**.

V. Les différents protocoles de communications

Cette liste contient tous les protocoles de communication disponible sous Android sans connexion internet.

- WI-FI Direct
- WI-FI DLNA
- Bluetooth
- NFC

Il existe d'autres protocoles de communication tel que le Quadribande et le VPN mais ces deux technologies ont besoin d'un accès internet, qui n'est pas compatible avec notre application car celle-ci doit pouvoir communiquer partout (métro). L'infrarouge étant de moins en moins utilisé et son rayon d'action étant très restreint nous ne l'avons pas sélectionné. Les autres technologies possibles n'apportent pas autant de garantie et sont beaucoup moins répondant que les quatre protocoles de communications cité ci-dessus

VI. Caractéristiques de chaque protocole de communication

1. Wi-Fi Direct

Cette nouvelle technologie, officialisée par le consortium Wi-Fi Alliance depuis octobre 2009, permet à deux appareils équipés du Wifi, de communiquer entre eux, sans passer par un point d'accès ou un routeur, un peu à la manière du Bluetooth, mais avec un débit beaucoup plus important. Le Wi-Fi Direct garantit une qualité CD (on dit aussi Lossless), soit 44.1 kHz en 16 bits. Le débit est largement suffisant pour transférer des fichiers de type FLAC, par exemple. En outre, le Wi-Fi direct intègre le protocole de sécurisation des réseaux WPA2. Bien que lancée depuis trois ans, cette technologie prometteuse peine à se démocratiser. Néanmoins, les choses pourraient évoluer rapidement, puisque le Wifi Direct est intégré dans l'OS pour mobiles Android 4.0. Les premières stations d'accueil compatibles commencent à apparaître.



Consommation d'énergie 0.1 Watt

Avantage

- Sécurité WPA2
- Débit plus important que le Bluetooth
- Tends à se démocratiser
- Traverse les obstacles, murs

Inconvénient

- Très peu répandu dans les terminaux Android
- Besoin d'Android 4.0 => environ 40% des terminaux mobiles Android

Il s'agit du futur du Bluetooth, même utilité mais avec des distances et débits supérieur.

2. Wi-Fi DLNA :

Le DLNA (Digital Living Network Alliance) est un consortium regroupant plus de 250 sociétés du secteur des nouvelles technologies. L'appellation désigne, par extension, un standard de communication entre différents types d'appareil (TV, téléphones, enceintes...). Le DLNA s'appuie sur le protocole Wi-Fi pour la communication sans fil. Ainsi, tous les appareils compatibles DLNA peuvent communiquer entre eux, et notamment les Smartphones, tablettes et stations d'accueil. Contrairement au Wi-Fi Direct, le Wi-Fi DLNA s'appuie sur un réseau local.



Dans la pratique, les normes DLNA peuvent être utilisées de nombreuses manières : pour partager le contenu d'un disque dur externe en sans-fil entre plusieurs ordinateurs, pour voir vos photos stockées sur votre appareil ou votre téléphone portable depuis un téléviseur ou un moniteur DLNA compatible

Consommation d'énergie : 1 Watt

Avantage

- Avantage du Wifi
- Connexion a distance
- Principalement pour le multimédia et la lecture sur d'autre terminaux

Inconvénient

- Besoin d'un routeur
- Grosse consommation d'énergie

Le besoin du routeur pour la technologie Wi-Fi DLNA n'est pas compatible avec notre application, celle-ci doit pouvoir communiquer n'importe où.

Les différentes normes de Wi-Fi existant

802.11b : presque dépassée

Il s'agit de la première norme sans fil exploitée par le grand public et les professionnels depuis le début des années 2000. Fonctionnant sur la bande de fréquences radio de 2,4 GHz, le 802.11b offre une portée assez importante. Elle peut varier de 30 à 100 m, voire 400 m suivant le matériel utilisé et l'environnement. En revanche, à 11 Mbps, son débit théorique demeure. En pratique, c'est pire : il dépasse rarement 3 à 4 Mbps. En effet, de nombreuses pertes de données dues aux transmissions radio entraînent des corrections d'erreurs permanentes, gourmandes en bande passante. Pour partager une connexion à l'internet, les 11 Mbps du 802.11b suffisent. En revanche, pour des applications plus gourmandes en bande passante comme le partage de gros fichiers (plusieurs dizaines de méga-octets), le streaming audio/vidéo ou le jeu en réseau, le 802.11b montre ses limites. Il vaut donc mieux éviter d'utiliser le matériel à cette norme pour constituer un réseau domestique. D'ailleurs, les équipements de type adaptateurs Wi-Fi, routeurs ou point d'accès à cette norme ne sont quasiment plus commercialisés, à l'exception des assistants personnels. Même les modèles les plus récents n'intègrent encore qu'un module Wi-Fi 802.11b. Ce qui suffit pour accéder à l'internet et à quelques fichiers peu volumineux (ce qui sera la vocation première d'un assistant personnel Wi-Fi), bien que la norme soit moins satisfaisante en termes de sécurité.

802.11g : en cours

La norme 802.11g, qui succède au 802.11b, est logiquement plus performante, au moins en termes de débit et de sécurité. Elle s'est généralisée pour tous les usages : connexion à un réseau domestique, d'entreprise, à une borne publique. Le débit théorique du 802.11g atteint 54 Mbps (moins de 30 Mbps en pratique). Le 802.11g utilise une fréquence de 2,4 GHz et a donc une portée sensiblement égale à celle de 802.11b. Par ailleurs, le 802.11g a l'avantage d'être compatible avec la quasi-totalité du matériel 802.11b. Un même réseau pourra ainsi accueillir des anciens équipements 802.11b et du matériel 802.11g. En revanche, les débits seront nivelés par le bas : tous les équipements fonctionneront en 802.11b. Si le 802.11g fonctionne dans un rayon de 100 m à partir d'un point d'accès, de nombreux utilisateurs rencontrent des soucis de portée et donc de débit en intérieur. Il faut en effet prendre en compte les murs, les éventuelles structures métalliques et autres mobiliers qui font souvent obstacle aux ondes, réduisant la portée parfois à moins d'une dizaine de mètres. Certains constructeurs proposent d'augmenter le débit théorique des équipements 802.11g voire de le doubler (108 Mbps) sous des appellations marketing diverses et variées : Super G, SpeedBooster, AirPlusXtremeG, Turbo... Ces technologies, souvent propriétaires, ne relèvent d'aucun standard existant. Elles obligent l'utilisateur à choisir la même marque et la même technologie pour tout le matériel composant le réseau : routeur, point d'accès, adaptateurs pour PC et portables... Sans cela, point de performances. Onéreux et contraignant.

802.11i : sur la sécurité

Cette norme, officiellement certifiée en 2004, a pour unique but d'améliorer la sécurité des transmissions sans fil. Elle s'appuie sur l'AES (*Advanced Encryption Standard*) et offre un système de chiffrement beaucoup plus évolué avec des clés de cryptage pouvant atteindre 256 bits. Il s'agit donc d'une évolution plus qu'une nouvelle norme, qui vient compléter les standards 802.11a, b et g. Elle est intégrée à tous les nouveaux équipements et sa prise en compte pour les équipements datant d'avant 2004 nécessite une simple mise à jour, à condition que ces derniers ne soient pas trop anciens.

802.11n : l'avenir

Le 802.11n devrait être le prochain standard ratifié par l'IEEE (Institute of Electrical Electronic Engineers) en 2006. Il est censé quadrupler le débit des réseaux Wi-Fi, lui faisant atteindre 100 Mégabits par secondes (Mbps), tout en restant compatible avec les standards b et g. Des fabricants ont d'ores et déjà commercialisé des produits qui laissent présager ce que sera le 802.11n : plus de portée et un débit plus constant, grâce à la technologie à multiples antennes MIMO (*Multiple In, Multiple Out*). Toutefois, rien ne garantit que ces équipements soient compatibles avec le futur standard. Prudence, donc

3. Bluetooth

Le protocole Bluetooth permet la communication à courte distance entre deux appareils. Son principal avantage repose sur sa simplicité de mise en œuvre. Un appareil muni du Bluetooth repère automatiquement tous les émetteurs/récepteurs situés à proximité et s'y connecte dès que les deux modules ont accepté la communication. Malheureusement, les données subissent un processus de compression/décompression destiné à les alléger pour accélérer la vitesse de transmission. Cela entraîne une perte de qualité, notamment pour le son, qui se révèle d'autant plus perceptible que les fichiers MP3 sont eux aussi compressés. Dans notre application il s'agit principalement de PDF (ou documents Word ou open Office) la perte de qualité n'est pas vraiment une priorité par rapport à la vitesse de transmission.



Il existe une version Bluetooth Low Energy, qui est un élément du Bluetooth 4.0, ayant une consommation d'énergie très faible (exemple : Bluetooth activé pendant plus d'un an pour vider une batterie). Cette nouvelle version Bluetooth est très intéressante car elle permettra de laisser tourner l'application en fond sans trop affecter la batterie du téléphone de l'utilisateur. Seulement ce Bluetooth Low Energy n'est accessible qu'à partir des versions 4.2 d'Android. De plus dans cette version la portée du signal pourra dépasser les 60 mètres.

Les normes Bluetooth

- IEEE 802.15.1 définit le standard Bluetooth 1.x permettant d'obtenir un débit de 1 Mbit/s ;
- IEEE 802.15.2 propose des recommandations pour l'utilisation de la bande de fréquence 2,4 GHz (fréquence utilisée également par le Wi-Fi). Ce standard n'est toutefois pas encore validé ;
- IEEE 802.15.3 est un standard en cours de développement visant à proposer du haut débit (20 Mbit/s) avec la technique Bluetooth ;
- IEEE 802.15.4 est un standard en cours de développement pour des applications sans fils à bas débit et à faibles coûts. Il est actuellement utilisé par Zigbee pour ses couches basses.

Avantage

- Rapidité de connexion
- Simplicité
- Très faible consommation d'énergie

Inconvénient

- Courte distance
- Faible débit
- Traverse difficilement les murs (en fonction de la propriété du mur)

A long terme le Bluetooth est appelé à être remplacé par le WIFI direct.

4. NFC

Un périphérique NFC est capable de communiquer avec le matériel ISO/CEI 14443 existant, avec un autre périphérique NFC ou avec certaines infrastructures sans-contact existantes comme les validateurs des transports en commun ou les terminaux de paiement chez les commerçants. La NFC équipe aujourd'hui des cartes utilisées dans les transports, dans le commerce ou pour l'accès à certains services publics et de plus en plus de terminaux mobiles. La NFC équipait en effet en 2011 50 millions de tablettes tactiles ou téléphones mobiles, chiffre qui pourrait atteindre 300 millions en 2013¹. Dotés d'un écran, d'un clavier et d'une connexion internet, ces terminaux NFC ont un fort potentiel d'usages en favorisant les interactions entre les machines, les objets et un contexte.



Au contraire d'autres techniques de radio-identification ou du Bluetooth dont la portée est d'une dizaine de mètres, la technique NFC n'est utilisable que sur de très courtes distances (quelques centimètres). Elle suppose une démarche volontaire de l'utilisateur et normalement ne peut pas être utilisée à son insu.

Ce dernier point ne correspond pas l'utilisation de notre application, dont les échanges de données se feront en transparence de l'utilisateur. Il reste néanmoins intéressant mais ne sera pas privilégié.

Il serait par contre intéressant de l'utiliser dans certaine condition, par exemple dans les lieux avec une densité de gens très important, heure de pointe, dans le métro, file d'attente, concert etc... Il faudra le demander explicitement à l'utilisateur dans ce cas. L'utilisation de cette technologie serait intéressante à utiliser pendant les heures de pointes, ou les personnes sont assez proches les unes des autres (métro...).

Avantage

- Bonne garantie de déploiement à l'avenir
- Beaucoup de périphérique NFC à l'avenir
- Communication simple

Inconvénient

- Très courte distance
- Pas vraiment adapté à notre application

La très courte distance et son utilisation n'est pas réellement adapté au principe de notre application, le NFC est plus utilisé pour payer et faciliter les petites transactions, cela serait adaptable dans notre application mais la très courte distance (10 cm) réduit considérablement la performance de la recherche en plein rue, ou dans les lieux communs.

VII. Différence

Cette partie est un rapide récapitulatif des différences entre la vitesse, la distance, la consommation et la fréquence des différences protocoles de communication :

	<i>Vitesse</i>	<i>Distance des protocoles</i>	<i>Consommation</i>	<i>Fréquence</i>
WIFI Direct	WIFI b : 11 Mbps WIFI g : 54Mbps WIFI n : 100Mbps	20m à 300m	0.1 Watt	2,4 GHz
Bluetooth	1 Mbps <i>Version low energy</i> : 200 kbit/s	10 à 20m <i>Version 4.0</i> : 50m	Faible <i>Version low energy</i> : très faible	2,4 GHz
NFC	425kbit/s	10 cm	Faible	13,56 Mhz

Nous constatons que le Wifi direct reste la meilleure solution du point de vue vitesse et distance, suivi par le Bluetooth et le NFC.

VIII. Résultat et choix des technologies

Dans le cadre de notre application, nous recherchons une technologie sans-fil étant la mieux adaptée à notre projet, suite à cet état de l'art nous avons pu juger des avantages et inconvénients de chaque protocole de communication. Nous avons décidé de retenir deux modes de communications, le première étant le Bluetooth, permettant une connexion facile et rapide aux autres appareils et possédant une distance plutôt correcte (20 mètres) pour le périmètre de notre application (métro, foule, file d'attente), cette technologie à l'avantage d'équiper tous les périphériques Android existant (le nombre de périphériques Android sans Bluetooth étant dérisoire il n'est pas pris en compte). Seulement cette technologie est appelée à être remplacée par le WIFI-direct, qui permet à deux appareils de se connecter entre eux sans routeur ou box, permettant un échange de données plus rapide et sur une plus grande distance mais cette technologie est récente et n'est disponible que sur les dernières versions d'Android (la 4.0 et +). Seulement le pourcentage de répartition d'appareils sous Android 4.0 et plus est de 40% et tend à s'élargir ! Nous avons donc décidé de faire cohabiter les deux technologies, le Bluetooth activé par défaut et si le téléphone possède le WIFI-direct, demander l'activation de celui-ci à l'utilisateur. Le NFC, par rapport à sa très faible distance, n'interviendra qu'en troisième choix.

Il est aussi prévu que en 2016, le WIFI Direct et le Bluetooth Low Energy équipe respectivement 2 et 3 millions de terminaux Android, il serait alors judicieux d'anticiper cette évolution dans notre application.

Les protocoles de communication dans fil choisis pour équiper notre application sont donc :

Le Bluetooth 4.0 (version Ice Cream Sandwich)

Le WIFI- Direct

IX. Protocoles et Version retenues

Après avoir étudié toutes les versions d'Android et leurs protocoles de communications nous avons décidé de retenir les choix suivant :

- Nous travaillerons sur la version d'Android 4.0 **Ice Cream Sandwich** avec **l'API 15**



- Nous utiliserons **le Bluetooth** en version 4.

- Bluetooth par défaut et proposerons à l'utilisateur d'activer le **WIFI Direct** s'il le souhaite. Le **NFC** sera une option que l'utilisateur peut enclencher s'il est dans un espace confiné ou aux heures de pointes.



X. Sources

Ci-après les principales sources que nous avons utilisées pour notre état de l'art.

<http://developer.android.com>

<http://www.wikipedia.fr>

<http://www.abiresearch.com/>

<http://www.developpez.com>