Introduction

Avec les progrès et l'augmentation dans les technologies informatiques, les capacités des appareils mobiles ont considérablement augmenté au cours des années. Parallèlement, les applications mobiles sont devenues plus sophistiquées et plus complexes. Malgré un développement incessant dans les compétences, les appareils mobiles ont encore des limitations en termes de mémoire, de puissance de traitement, de bande passante réseau et de durée de vie de la batterie par rapport aux ordinateurs de bureau ou les laptops. Ces limitations interrompent l'exécution d'applications à forte intensité de calcul telles que le jeu en haute définition et la diffusion vidéo. Certaines de ces lacunes peuvent être submergées en permettant aux appareils mobiles de tirer parti des services et des ressources fournis par le cloud computing. Cette approche visant à exploiter les ressources et les services en nuage pour améliorer les capacités des appareils mobiles est appelée MCC.

Dans MCC, les périphériques mobiles migrent leur traitement de données, leur exécution d'application et leurs tâches de stockage de données sur le nuage, et seuls les résultats sont affichés. Il améliore les capacités des appareils mobiles pour exécuter des applications à forte intensité de calcul. Toutefois, le fait de s'appuyer sur le serveur cloud pour l'exécution de l'application n'est pas toujours possible en raison de la connectivité réseau faible et intermittente ou de la disponibilité de l'Internet. Dans de telles situations, le cloud mobile ad hoc (MAC) vient et joue son rôle pour exécuter les applications à forte intensité de calcul sans avoir besoin a des points d’accès, ou des réseaux sans fil.

Dans cette thèse, nous allons discuter des différents aspects du domaine informatique, les analyser et les étudier afin de conclure que nous avons besoin de MAC. Ensuite, nous allons discuter les différentes recherches menés dans ce domaine, et développer une plateforme pour démontrer l'utilisation du MAC

CHAPITRE 01

La mobilité et le Cloud Computing

1. Introduction

Malgré l'utilisation croissante des appareils mobiles dans notre vie quotidienne, l'exploitation de son plein potentiel est difficile en raison de ses problèmes inhérents, comme la pénurie de ressources, les déconnexions fréquentes et la mobilité.

Dans ce chapitre, nous allons expliquer les différents aspects de ce domaine tels que la Cloud Computing, Mobile Computing, Mobile cloud computing et les différents problèmes et défis de ces domaines tout en soulignant le besoin de MAC. Finalement, nous proposons une classification basée sur le contexte des MACs afin de discuter les différentes recherches menés dans ce domaine.

2. Cloud Computing

La croissance continue de la demande des ressources et des données a créé un grand problème, comment accéder à ces services et utiliser ces derniers n'importe où et n'importe quand facilement. À cause de ce problème, les chercheurs ont découvert une solution radicale pour répondre à cette question fondamentale, « pourquoi nous n'assemblons pas ces ressources et données dans un nuage pour les rendre faciles et efficaces dans leurs utilisations ? » À partir de ce concept, le terme « Cloud Computing » est apparu.

2.1. Définition du Cloud Computing :

Le cloud Computing est un modèle qui permet d'accéder à un réseau partagé de ressources informatiques configurables (par exemple : réseaux, serveurs, stockage, applications et services), qui peuvent être rapidement provisionnées et diffusées avec un minimum d'effort de gestion ou d'administration de services **(NIST, 2013)**.



Figure 0-1 Cloud Computing (Wikiversity, 2017)

2.2. Les caracteristiques

Les differents caracteristiques du cloud computing sont:

2.2.1. Service à la demande

Un consommateur peut fournir unilatéralement des capacités informatiques, telles que le temps du serveur et le stockage du réseau, selon les besoins automatiquement, sans nécessiter d'interaction humaine avec chaque fournisseur de services **(NIST, 2013)**.

2.2.2. Accès a un réseau étendu

Les capacités sont disponibles sur le réseau et accessibles par l'intermédiaire de mécanisme standard qui favorisent l'utilisation par des plateformes client hétérogènes, mince ou épaisses (par exemple, téléphones mobiles, tablette, ordinateurs portables et postes de travail) **(NIST, 2013)**.

2.2.3. Mise en commun des ressources

Les ressources informatiques du fournisseur sont mises en commun pour desservir plusieurs consommateurs à l'aide d'un modèle multi locataire, avec différentes ressources physiques et virtuelles assignent dynamiquement et réaffectées selon la demande du consommateur. Il y a un sentiment d'indépendance de localisation en ce que le client n'a

généralement aucun contrôle ou connaissance sur l'emplacement exact des ressources fournies, mais peut être en mesure de spécifier l'emplacement a un niveau supérieur d'abstraction (par exemple, pays, état ou Datacenter). Des exemples de ressources comprennent le stockage, le traitement, la mémoire et la bande passante du réseau **(NIST, 2013)**.

2.2.4. Elasticité rapide

Les capacités peuvent être élastiquement provisionnées et libérées, dans certains cas automatiquement, à l'échelle rapidement vers l'extérieur et à l'intérieur en fonction de la demande. Pour le consommateur, les capacités disponibles pour l'approvisionnement semblent souvent illimitées et peuvent être appropriées en n'importe quelle quantité à tout moment **(NIST, 2013)**.

2.2.5. Mesure de service

Les systèmes Cloud contrôlent et optimisent automatiquement l'utilisation des ressources en exploitant une capacité de comptage a un certain niveau d'abstraction approprie au type de service (par exemple, stockage, traitement, bande passante et comptes d'utilisateurs actifs). L'utilisation des ressources peut être surveillée, contrôlée et signalée, assurant la transparence tant pour le fournisseur que pour le consommateur de service utilisé **(NIST, 2013)**.

2.3. Les modèles de déploiement

Il existe trois modèles de déploiement et d'exploitation des CC qui sont :

2.3.1. Cloud Public

L'infrastructure du ce modèle est accessible par le grand public (ce ne veut pas dire que les données sont accessibles à n'importe qui, seulement qui va payez en fonction d'utilisation vas accéder au service du Cloud), cette infrastructure peut être détenu, géré et exploité par une entreprise qui vend leurs services Cloud **(NIST, 2013)**.

2.3.2. Cloud Privée

Le Cloud privé ou le Cloud dédie désigne une infrastructure accessible seulement par une entreprise ou une organisation, c'est-à-dire seulement les serveurs privatifs de l'entreprise qui va gérer l'ensemble de données pour les diffèrent utilisateurs de l'entreprise. Ce modèle de Cloud sera hébergé en interne ou dans un Datacenter [1] **(NIST, 2013)**.

2.3.3. Cloud hybride

Un Cloud hybride est un service Cloud intégré utilisant à la fois des Cloud privés et des Cloud publics pour remplir diffèrent fonctions au sein d'une même organisation **(NIST, 2013)**.

2.4. Les services

Le cloud computing offre trois services, à savoir comme :

2.4.1. Infrastructure comme Service (IaaS)

L'infrastructure comme Service (IaaS) est l'une des trois principales catégories des services de Cloud computing. Comme tous les services de Cloud computing, ce service donne accès aux ressources informatiques dans un enivrement virtualisé, le « Cloud », à travers une connexion publique, généralement Internet. Dans le cas d l'IaaS, la ressource fournie est du matériel informatique virtualisé ou, en d'autres mots, une infrastructure informatique. La définition du service comprend des offres telles que l'espace serveur (stockage), des connexions réseau, des adresses IP et des load balancers [2]. Le consommateur ne gère pas ou ne contrôle pas l'infrastructure, mais contrôle les systèmes d'exploitation, le stockage et les applications déployées, et contrôle limite des composants réseau sélectionne. **(Interoute, 2017)**

2.4.2. Plateforme comme Service (PaaS)

Plateforme comme Service (PaaS) est une catégorie de service Cloud computing qui fournit la plateforme et l'enivrement informatique nécessaire aux développeurs pour mettre

[1] C'est un centre qui centralise des données informatiques.
[2] Est une technique utilisée par les réseaux les plus importants afin d'équilibrer la sollicitation des serveurs.

en place leurs différents services et application sur internet. Les services PaaS sont hébergés dans le Cloud et les utilisateurs y accèdent simplement, par leur navigateur web. **(Interoute, 2017).**

2.4.3. Logiciel en tant que Service (SaaS)

Dans ce type de service, des applications sont mises à la disposition des consommateurs. Les applications peuvent être manipulées à l'aide d'un navigateur web ou installées de façon locative sur un PC, et le consommateur n'a pas à se soucier d'effectuer des mises à jour, d'ajouter des patches de sécurité et d'assurer la disponibilité du service. Gmail est un exemple de tel service. Il offre au consommateur un service de courrier électronique et le consommateur n'a pas à se soucier de la manière dont le service est fourni. **(NIST, 2013).**

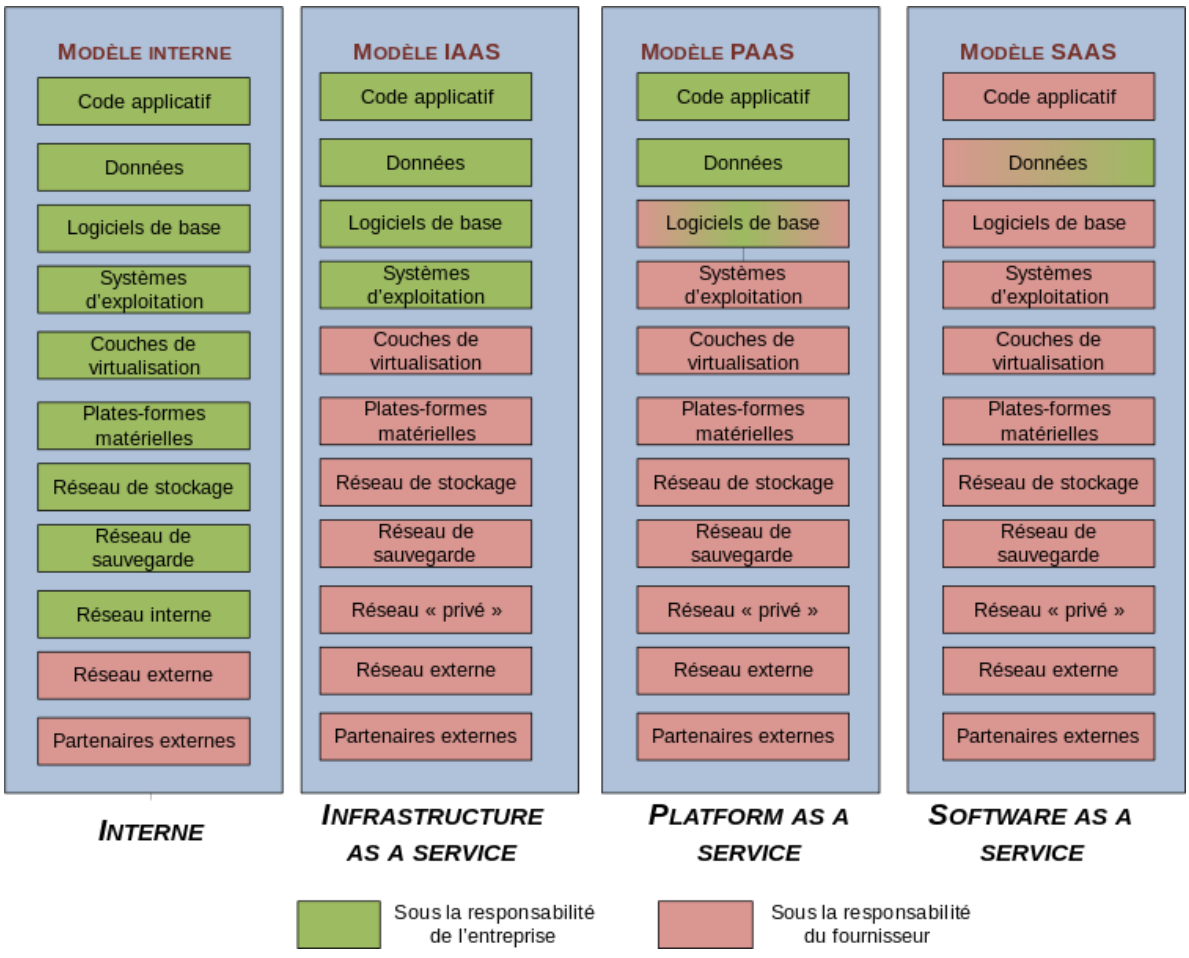


Figure 0-2 Les services du Cloud computing (Wikimedia, 2013)

3. Mobilité et Smartphone

De nos jours, le domaine mobile a reçu un intérêt considérable et l'adoption massive sur l'utilisation des appareils mobiles pour effectuer nos tâches, et ceci preuve la croissance exponentielle de l'utilisation des applications mobiles ainsi leurs développements.

3.1. La mobilité

Le nomadisme numérique ou la mobilité connectée se réfère aux utilisateurs des technologies électroniques et sans fil. Ces technologies permettent l'accès aux médias numériques et à l'information numérique, à les modifier ou à communiquer via la téléphonie mobile ou Internet et à travailler en ligne et hors ligne, quel que soit l'endroit où vous êtes.

Les modèles de mobilité caractérisent le patron du mouvement des éléments mobiles et peuvent être divises en deux catégories selon que la mobilité est contrôlable **Modèle de mobilité incontrôlable**, cette catégorie produire comme la mise à jour de l'emplacement et le routage de données, et **Modèle de mobilité contrôlable**, cette catégorie comme un contrôleur du mouvement et la trajectoire du conception et routage prenant en charge la mobilité.

3.2. Les smartphones

Les Smartphones sont des appareils extrêmement sophistiqués, qui fournissent des fonctionnalités en plus celles des téléphones mobiles classiques comme la navigation web, consultation et l'envoie des courriers électroniques, messageries vocales et visuelles.

Dernièrement, on a aussi vu les Smartphones sophistiqués bénéficiant rapidement de la connaissance et synthèse vocale. Les Smartphones exécutent tous divers logiciels et application grâce à des systèmes d'exploitation spécialement conçus pour les mobiles et ils peuvent être personnalisés en y installant des applications additionnelles telles que des jeux ou des utilitaires grâce aux magasins d'application en ligne (store).

3.3. Les applications mobiles

Une application mobile c'est toute application qui peut être exécutée sur un dispositif mobile tel que téléphone mobile, une tablette, ou tout autre périphérique mobile. Ces applications seront alors disponibles aux utilisateurs n'importe où et n'importe quand.

4. Mobile Computing

Au cours des dernières années, le développement rapide des technologies sans fil nous a permis de développer des téléphones mobiles équipés de technologies à accès multiple, avec des interfaces cellulaires classiques (ex. 2G et 3G),

4.1. Définition

Ce sont des dispositifs sans fil qui sont connectés à une BS via une interface aérienne cellulaire, tout en pouvant coopérer et communiquer via des liaisons à courte portée, comme illustré à la Fig. 1. En fait, un cloud mobile peut être considéré comme un arrangement local collaboratif de dispositifs mobiles multistandards, visant à s'entraider, à fournir de meilleures performances et à améliorer l'efficacité de la communication tout en offrant de nouveaux services et en exploitant davantage les ressources efficacement. **(Hamidreza B, et al).**

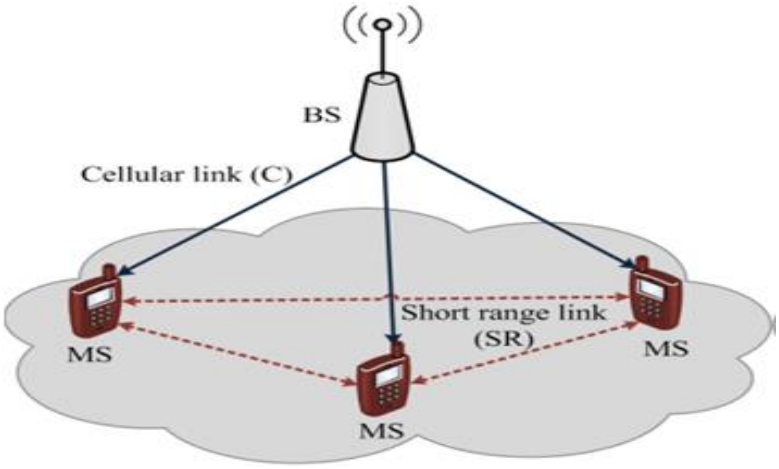


Figure 0-3 Architecture du mobile computing. (Hamidreza B, et al)

5. Mobile Cloud Computing

L'évolution du matériel et des logiciels mobiles a permis aux utilisateurs d'effectuer des tâches qui n'étaient autrefois possibles que sur des ordinateurs personnels et des appareils spécialisés comme les appareils photo numériques et les systèmes de navigation personnelle GPS, cependant les appareils mobiles sera toujours limité, car ils ne peuvent pas exécuter certaines applications par elles-mêmes sans aucune contrainte, à cause de ce problème, Mobile cloud computing est inventé.

5.1. Définition

Le paradigme du MCC a été introduit en tant que technologie capable de remédier aux problèmes et aux défis de l'informatique mobile, comme le manque de ressources dans les appareils mobiles, en exploitant les ressources sur les plateformes en nuage **(Mobile cloud computing Lab, 2013)**.

MCC est la combinaison du cloud computing, de mobile computing et les réseaux sans fil pour apporter des ressources informatiques riches aux utilisateurs mobiles. Le but ultime de MCC est de permettre l'exécution d'applications mobiles riches sur une abondance de périphériques mobiles, avec une expérience utilisateur riche.

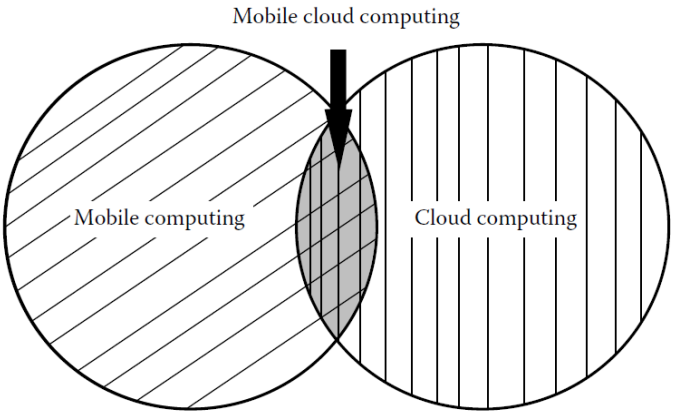


Figure 0-4 l'origine du MCC (Debashi De, 2016)

5.2. Architecture du MCC

5.2.1. Architecture orientée services

L'architecture orientée service (SOA) de MCC se compose de trois couches **(Debashi De, 2016)**, comme la figure 1-5 montre.

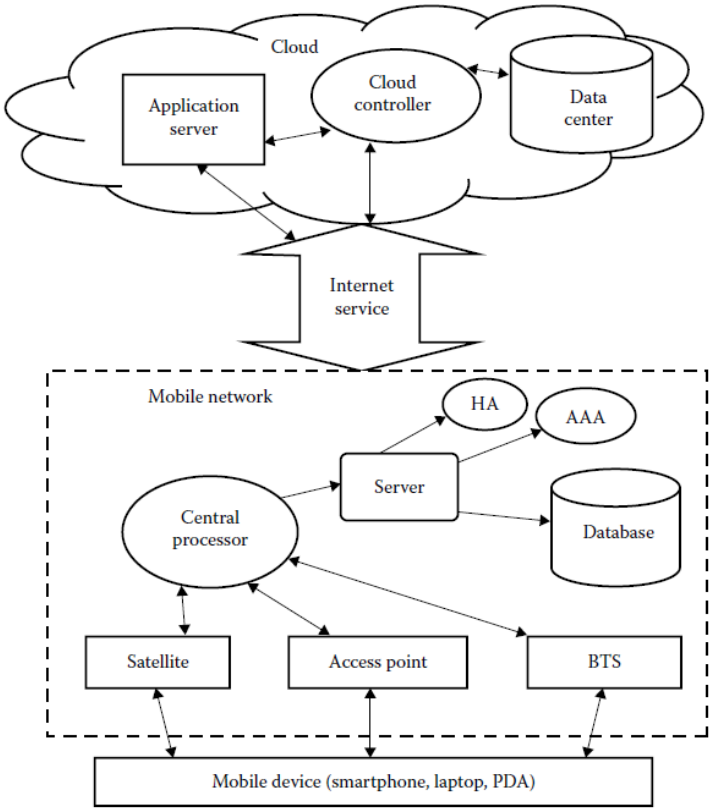


Figure 0-5 l'architecture SOA du MCC (Debashi De, 2016)

a. Reseaux mobile:

Un réseau mobile contient des périphériques mobiles et des opérateurs de réseau. Les appareils mobiles peuvent être des smartphones, des PDA, des ordinateurs portables, etc. Ils sont connectés à l'opérateur du réseau via les BTS (Stations d'émetteur-récepteur de base), ou les points d'accès. Les requêtes et les informations d'un appareil mobile, telles que l'ID et l'emplacement, sont transmises au processeur central et aux serveurs des fournisseurs de réseau. Ici, les opérateurs fournissent divers services tels que AAA (authentification, autorisation et comptabilité) en fonction des données HA (agent d'accueil) et d'abonné stocké dans la base de données **(Debashi De, 2016)**.

b. Service Internet :

Le service Internet joue le rôle de pont entre le réseau mobile et le cloud. Les demandes d'abonné sont livrées au cloud via un service Internet haute vitesse. En utilisant des connexions câblées ou des technologies avancées 3G ou 4G tels que HSPA, UMTS, WCDMA, LTE, etc.[3], l'utilisateur peut obtenir un service transparent à partir du cloud **(Debashi De, 2016)**.

c. Service cloud :

Après avoir reçu toutes les demandes des utilisateurs, le contrôleur du cloud traite les demandes et leur fournit un service en fonction de leurs besoins. Cloud possède quelques couches de fourniture de services **(Debashi De, 2016)**. Ces couches de service sont discutées comme suit :

- **Couche Data Center :** Fournis les installations matérielles et l'infrastructure pour le cloud.
- **IaaS :** Fournis un stockage, des serveurs, des composants réseau et du matériel à ses clients sur une base de « pay as you use ».
- **PaaS :** Fournis un environnement ou une plate-forme intégrée permettant aux utilisateurs de créer, tester et déployer plusieurs applications.
- **SaaS :** Fournis de nombreux types de solutions logicielles telles que CRM (Customer Relation Management), ERP (Enterprise Resource Planning), MIS (management information system), HRM (Human Resource management), etc., à la demande sans aucune installation dédiée sur le site client.

5.2.2. Architecture Agent-Client

Dans cette architecture, les appareils mobiles ne sont pas directement connectés au cloud. Ils sont connectés au nuage via certains agents tels que Femtocell **(Zhang J, et al)**, Cloudlet **(Satyanarayanan M, et al)**, ou les deux, comme le montre la figure 1-6 :

[3] Les différentes technologies sans-fil existent

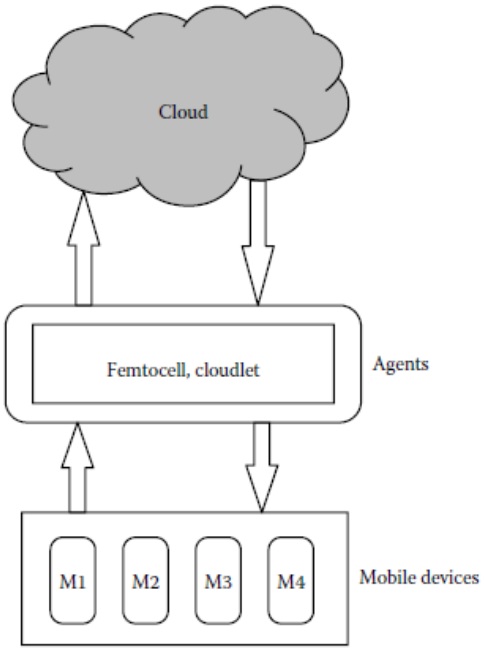


Figure 0-6 l'architecture Agent-Client du MCC (Debashi De, 2016)

Ces agents peuvent répondre à la demande des utilisateurs avec une large bande passante élevée, une faible latence et un minimum du coût.

- **Cloudlet :** Cloudlet est un ordinateur riche en ressources ou un cluster d'ordinateurs qui est fiable et bien connecté à Internet haute vitesse et disponible pour les appareils mobiles.
- **Femtocell :** Femtocell connus sous le nom de « station de base », est la plus petite version des micro cellules traditionnelles. Les Femtocelles sont déployées à l'intérieur pour assurer une bonne couverture. Les appareils mobiles sont connectés aux femtocells, et les femtocells sont connectés au réseau mobile avec un DSL résidentiel, un câble large bande, des fibres optiques ou des technologies sans-fil. Ils peuvent fournir un service 3G et 4G transparent à l'utilisateur et peuvent être utilisés pour connecter des appareils mobiles avec un Cloudlet ou un Cloud.

5.2.3. Architecture Collaborative

De nos jours, les smartphones fonctionnent indépendamment en utilisant leurs capacités locales de calcul, de détection, de mise en réseau et de stockage. Lorsque les données sont partagées avec d'autres appareils via un serveur ou un serveur centralisé, cela

nécessite un téléchargement et un téléchargement coûteux. Cela peut être évité par l'informatique collaborative **(Dinh HT, et al)**. Dans cette architecture, les ressources d'un appareil mobile sont utilisées en considérant l'appareil comme faisant partie d'un Cloud. Le serveur en Cloud peut être le contrôleur et le planificateur pour la collaboration entre les périphériques. En combinant les données des smartphones et l'informatique.

5.3. Défis et challenge du MCC

Dans MCC, les smartphones ont besoin de point d'accès 3G ou réseau pour effectuer certaines tâches. Cependant, cette technologie n'est pas toujours disponible et parfois coûteuse lorsqu'il s'agit de partage des services (GPS, Stockage, Calcul, etc.) ou de la messagerie entre les différentes personnes.

Connexion à un point d'accès réseau ou la 3G n'est pas toujours possible par exemple, dans un théâtre, les bâtiments d'une usine, les tunnels, les musées, etc. où les constructions de bâtiment peuvent bloquer le 3G et les points d'accès ne sont pas disponibles. Donc, la connexion entre un groupe de personnes pour échanger les services et les données sera un problème dans ce cas.

Les problèmes mentionnés au-dessus pourraient être résolus en créant le réseau Ad-hoc entre deux ou plusieurs téléphones mobiles à l'aide de réseau de téléphones Wi-Fi Direct.

6. Les reseaux Ad-hoc

Un réseau Ad-hoc est une collection de nœuds mobiles sans fil (ou routeurs) formant dynamiquement un réseau temporaire sans l'utilisation d'une infrastructure réseau existante ou d'une administration centralisée. Les routeurs sont libres de se déplacer aléatoirement et s'organisent arbitrairement ; ainsi, la topologie sans fil du réseau peut changer rapidement et de façon imprévisible. **(Subir KS, et al).**

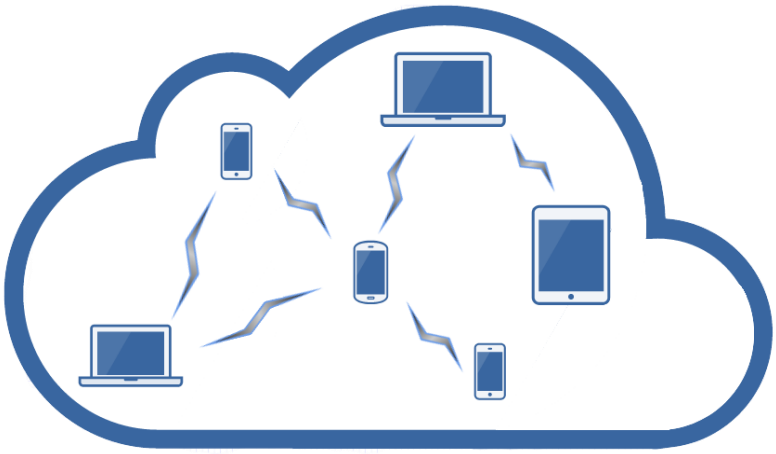


Figure 0-7 Réseaux Mobile Ad-hoc

6.1. Différences entre les réseaux sans fil cellulaires et Ad-hoc

Le tableau suivant donne les différences principales entre les réseaux cellulaires et Ad-hoc :

Cellulaires	Ad-hoc
Réseau avec infrastructure	Réseau sans infrastructure
Fixé, sites cellulaire pré-localisés	Pas de station de base et déploiement rapide
Topologie du réseau statique	Topologies de réseau hautement dynamiques avec multihop
Environnement relativement attentif et connectivité stable	Environnement hostile (bruit, pertes) et connectivité irrégulière
Une planification détaillée avant la station de base peut être installée	Se forme automatiquement et s'adapte aux changements
Coûts élevés d'installation	Rentable
Grand temps d'installation	Moins de temps d'installation

Table 0-1 Différences entre les réseaux sans fil cellulaires et Ad-hoc (Subir KS, et al)

7. Mobile Ad-hoc cloud (MAC)

Le mobile ad-hoc ou MANET est un groupe de périphériques mobiles à proximité qui souhaitent partager leurs ressources les uns avec les autres. Le mac est un nouveau type de mcc. Il est généralement déployé sur des réseaux ad-hoc mobiles, ce qui permet d'exécuter des applications à forte intensité de calcul en utilisant les ressources d'autres appareils mobiles **(zaghdoudi b, et al)**. En outre, mac offre une solution viable pour un appareil mobile pour exécuter une application lorsqu'il n'y a pas de connexion internet sans fil. Dans mac, les appareils mobiles devraient gérer le cloud, authentifier les utilisateurs, surveiller les ressources et planifier les tâches en plus d'exécuter l'application. De telles fonctionnalités supplémentaires consomment des cycles d'énergie et de processeur d'appareils mobiles. Enfin, les appareils stationnaires locaux tels que les ordinateurs personnels et les décodeurs peuvent également devenir membres de mac **(Miluzzo E, et al)**.

7.1. Classification des MACs

À partir de la dernière définition, nous allons faire une classification basée sur le contexte de MAC, qui sera divisé en différents aspects, chaque aspect a ses propres problèmes et solutions proposés. Le tableau suivant représente cette classification :

Aspect	*Problèmes*	*Solutions proposés*
Déchargement des tâches	- Instabilité. - Gaspillage des ressources. - La latence du réseau.	- MinHop - MetComm - MCTComm - MinMinComm - MaxMinCom - SufferageComm - OCS - RCS - CCS
Planification des tâches et allocation	- Mobilité élevée. - Temps d'exécution plus long. - Répartition inappropriée de la charge de travail. - Forte consommation d'énergie.	- Offline centralized algorithme - Online distributed algorithme - Scavenger - Batch-Based - Adaptive probabilistic task scheduler - Task allocation mechanisme

Formation du mac	- Connectivité aléatoire. - Rigidité.	- S-CMA - Ad-hoc mobile cloud - C-Protocol - Transient Cloud - Ad-hoc cloudlet
Confidentialité et sécurité	- Protection d'information du lieu. - Identification des nœuds malveillants.	- TMC - Probabilistic model for vulnerability detection - Trust assurance - Location privacy
Incitations et mobilité	- Coûteux et peu pratique. - Inefficace	- Incentive scheme - Directory-based - Double-sided bidding scheme - Stochastic programming - VCC
La gestion des ressources	- L'inconséquence de l'hétérogénéité des ressources. - Variation de charge dynamique.	- Planet Cloud - MMADC

Table 0-2 Taxonomie basée sur le contexte de mac

8. Conclusion

Les avancements sur MCC ont ouvert un passage vers un nouveau paradigme informatique appelé MAC. Bien qu'il existe de nombreuses études de recherche sur le cloud computing mobile et l'informatique Ad-hoc, la convergence de ces deux domaines accorde d'autres efforts académiques pour l'épanouissement du MAC.

Dans le prochain chapitre, on va discuter les différents efforts de recherche menés dans le domaine du MAC, et examinons les solutions correspondantes utilisant la taxonomie précédant.

CHAPITRE 02

Etat de l’art sur les MACs

1. Introduction

Comme indiqué précédemment, Le but de cette section est de discuter les différents efforts de recherche menés dans ce domaine. Dans ce contexte, nous étudions plusieurs problèmes entravant l'adoption de MAC et examinons les solutions correspondantes en concevant une taxonomie. En outre, nous comparons les solutions existantes dans le contexte du déchargement des tâches, planification des tâches et de l'allocation, formation du MAC, sécurité et confidentialité, incitations et mobilité, et de la gestion des ressources.

Non seulement, ont vas introduit les différentes plateformes mobiles et sa répartition dans le marché, et nous terminons par un coup d'œil sur l'environnement de travail Android OS, son architecture et ces versions existants.

2. Travaux connexe sur les MACs

Le MAC est encore à ses débuts, et une littérature très limitée est disponible sur le sujet. Le but de cette section est de discuter les différents efforts de recherche effectués dans ce domaine. Dans ce contexte, nous étudions plusieurs problèmes entravant l'adoption du MAC et examinons les solutions correspondantes.

2.1. Déchargement des tâches

Les auteurs **(Li B, et al)** sont concentrés sur le problème de décision concernant la façon de décharger les applications à forte intensité de calcul dans le MAC. Pour résoudre le problème, un ensemble d'heuristiques de planification en ligne et par lot, nommément MinHop, MetComm, MCTComm, MinMinComm, MaxMinComm et SufferageCo mm, ont été proposés pour décharger les tâches indépendantes entre les nœuds de manière dynamique. L'heuristique MinHop accorde une tâche en fonction d'un nombre minimum de sauts du nœud client.

L'heuristique MetComm attribue une tâche à ce périphérique qui peut prendre un temps minimum d'exécution pour compléter la tâche. L'heuristique MCTComm affecte les tâches en fonction du temps d'exécution minimum attendu sur un périphérique. Les heuristiques restantes ont été utilisées pour attribuer une tâche en considérant le coût de la communication. Pour étudier la performance des heuristiques proposées, des mesures différentes, telles que la moyenne, le temps d'attente moyen, le ralentissement moyen et l'utilisation moyenne sont utilisés. Les résultats suggèrent que le temps d'exécution prévu doit être pris en compte lors du mappage des tâches. En outre, les heuristiques proposées sont efficaces en termes de performance, mais seule la question du problème est la complexité.

Solutions proposées	Mise au point spécifiée	Avantages	Inconvénient
MinHop, MetComm, MCTComm, MinMinComm, MaxMinComm, SufferageComm **(Li B, et all)**	Mettre l'accent sur le problème de décision sur la façon de décharger les applications de calcul intensif en MAC	Déchargement optimal des tâches à haute performance	Complexité, plus de temps dans le processus décisionnel
OCS, RCS, CCS **(Chen M, et all)**	activer la stratégie énergétique et intelligente afin de décharger la tâche de calcul intensif à l'aide d'Ad hoc Cloudlet	Rentable et flexible	La sélection de nœuds fiables et sécurisés est difficile.

Table 0-1 Comparaison des solutions proposées par déchargement de tâches.

2.2. Planification des tâches et allocation

Les auteurs **(Lu Z, et al)** ont proposé des algorithmes pour résoudre le problème d'allocation de tâche dans l'environnement hétérogène sans fil. L'objectif de cette étude était de minimiser le temps moyen de réponse des tâches pour un ensemble complet de tâches en déterminant si elles doivent être distribuées ou non et sur quel périphérique elles doivent être exécutées. En outre, l'algorithme a également considéré les paramètres tels que le délai de communication, le délai de traitement et le délai de mise en file d'attente, tout en allouant la tâche. D'ailleurs, les auteurs ont classé le problème de l'attribution de tâches comme un NP-hard[4] et proposé deux approches nommées comme, hors connexion centralisée et distribuée en ligne, pour résoudre le problème. Les résultats ont été très prometteurs en termes de temps de réponse dans différents scénarios. Malgré les nombreux avantages des approches proposées, le problème d'équilibrage de la charge restera un problème qui doit être résolu à l'avenir.

(Kristensen MD, et al). À proposer un nouveau système de recherche cybernétique basée sur Scavenger qui permet la répartition des tâches et le mécanisme de planification parmi les nœuds participant à la communication. Pour effectuer la planification, Scavenger considère plusieurs facteurs tels que la localisation des données, la

[4] Lorsqu'une version décisionnelle d'un problème d'optimisation combinatoire s'avère appartenir à la classe des problèmes NP-complets, la version d'optimisation est NP-Hard.

capacité du réseau, la force du périphérique et la complexité de la tâche. En outre, le planificateur aide à déterminer si l'exécution de la tâche serait possible sur le périphérique local ou dans un environnement distant. Le système proposé introduit une amélioration significative au niveau de la performance dans l'exécution des applications mobiles et économise la consommation d'énergie. Cependant, la planification d'une petite tâche dans Scavenger cause un gaspillage de temps, car il nécessite plus de temps que le temps d'exécution réel.

Outre cela, les chercheurs **(Shi T, et al)** ont formulé le problème de planification de tâches à haute efficacité énergétique dans les nuages mobiles locaux. Dans ce contexte, un planificateur probabiliste adaptatif a proposé d'aider pour planifier différentes tâches en satisfaisant les contraintes de temps de la tâche tout en conservant la faible consommation d'énergie pour des applications en temps réel intensif en calcul. Le planificateur proposé peut offrir plusieurs avantages, tels que l'ordonnancement, l'évolutivité et la flexibilité en fonction de l'efficacité énergétique. Cependant, une grande complexité est l'un des inconvénients.

Un nouveau mécanisme d'allocation de tâches sert à réduire la consommation d'énergie et le coût de calcul a été proposé par **(Guo X, et al)**. De plus, un jeu Stackelberg en deux étapes est également conçu pour déterminer le nombre d'unités d'exécution que les nœuds esclaves sont prêts à offrir, tandis que le nœud principal définit les stratégies de prix pour les différents nœuds esclaves en fonction de leurs ressources partagées. Bien que la solution proposée contribue à résoudre le problème d'allocation de tâches dans le cloud mobile Ad-hoc, cependant, l'inconduite de la ressource et l'hétérogénéité opérationnelle des appareils mobiles, tout en allouant les tâches est l'un des inconvénients.

Solutions proposées	Mise au point spécifiée	Avantages	Inconvénient
Scavenger **(Kristensen MD, et al)**	permettre la répartition des tâches et le mécanisme de planification entre les nœuds participant à la communication	Amélioration de performance, et économie d'énergie	Perte de temps
Offline centralized and online distributed **(Lu Z, et al)**	minimiser le temps moyen de réponse des tâches pour un ensemble complet de tâches en déterminant si les tâches doivent être distribuées à un périphérique mobile ou non et sur quel périphérique mobile il doit être exécuté	Exécution rapide des tâches et efficacité énergétique	Déséquilibré de charge
Adaptive probabilistic scheduler **(Shi T, et al)**	programmer les tâches tout en conservant une faible consommation d'énergie	Économe d'énergie, scalable, flexible	complexité haute
Task allocation mechanism **(Guo X, et al)**	activer l'allocation de tâches pour les mobiles ad hoc cloud	réduire la consommation d'énergie et le coût de calcul	Incohérence de la ressource des appareils mobiles et de l'hétérogénéité opérationnelle

Table 0-2 Comparaison de la planification des tâches et des solutions proposées basées sur l'allocation.

2.3. Formation du MAC

Les auteurs **(Zaghdoudi B et al)** ont proposé un protocole-C qui est responsable de la gestion et du déploiement des nuages mobiles p2p sur MANET. Pour établir le MAC, le protocole C utilise quatre types de messages tels que l'installation du cloud, l'ajout de fournisseur, l'ajout de client et la configuration du cloud. Le protocole proposé gère les nœuds mobiles de manière dynamique. Dans l'environnement sans infrastructure, les nœuds mobiles peuvent facilement diviser leurs tâches à forte intensité de calcul pour effectuer l'exécution en utilisant l'architecture proposée de la plate-forme MAC. L'établissement de MAC peut offrir plusieurs avantages tels que l'ubiquité, la disponibilité, l'abordabilité, les

opportunités et la spontanéité. Néanmoins, des défis tels que convaincre les utilisateurs de collaborer à travers leurs appareils mobiles en tant que nœuds fournisseurs et formation légère nécessite une attention particulière.

Une plate-forme collaborative nommée Transient Cloud a été proposée par **(Penner T, et al)** qui permet aux périphériques mobiles à proximité de former un réseau mobile Ad-hoc et de partager leurs ressources. De plus, une version modifiée de la méthode hongroise sert à réaliser l'affectation des tâches dans le cloud Ad-hoc est proposée. Parmi les avantages de cette méthode est l'équilibrage de charge et la collocation des exécutions. La plate-forme proposée permet aux utilisateurs de créer un MAC en utilisant des appareils mobiles On-The-Fly[5] disponible à proximité. La seule limitation du travail est que les technologies actuelles ne permettent qu'une mise en œuvre partielle du nuage transitoire, mais encore, elles peuvent montrer le potentiel du MAC.

(Ordonez-Morales EF, et al). Ont proposé une solution sporadique d'augmentation mobile (S-CMA) basée sur le cloud qui permet aux utilisateurs de prêter les ressources à partir d'un cluster Ad-hoc d'appareils mobiles mouvants. La couche de virtualisation est utilisée pour aborder la complexité qui découle de la mobilité du cluster. S-CMA permet le partage et l'allocation de ressources dans un cluster mobile Ad-hoc. De plus, il fournit une solution aux approches existantes qui peuvent améliorer l'expérience des utilisateurs mobiles vers l'adaptation de la plate-forme de cluster mobile Ad-hoc. En outre, la S-CMA proposée aide à faire face à de nombreux défis associés à la CMA traditionnelle, tels que le calcul notable, le coût de communication de la migration de tâches à forte intensité de calcul pour les serveurs distants et la latence du réseau. Malgré de divers mérites de S-CMA, l'autonomie et l'adaptation au problème de la mobilité sont des défis majeurs.

Une architecture de jeux Ad-hoc basée sur le cloudlet a été proposé par **(Chi F, et al)**. L'architecture comprend deux modules. Le premier module permet aux utilisateurs mobiles de télécharger les ressources de jeu à partir des serveurs cloud ou des utilisateurs mobiles proches. Le deuxième module est basé sur l'allocation de tâches basée sur le cloudlet qui permet aux utilisateurs d'exécuter leurs tâches sur les périphériques mobiles

[5] Décrit les activités qui se développent ou se produisent dynamiquement

locaux à proximité de manière dynamique. Pour formuler le problème pour les deux modules, plusieurs algorithmes ont été proposés, ce qui réduit la consommation d'énergie par rapport à l'architecture de jeu basée sur le cloud. Le seul problème dans les algorithmes proposés est d'ignorer les ressources hétérogènes des dispositifs mobiles formant mobile Ad-hoc cloudlet, tout en allouant une tâche qui entraîne un gaspillage de ressources en termes de consommation d'énergie et de temps d'exécution.

Une architecture cloudlet a été proposée par **(Verbelen T, et al)** qui aide à gérer les applications au niveau des composants. L'architecture proposée permet aux utilisateurs de former dynamiquement le cloudlet en trouvant des périphériques mobiles avec les ressources disponibles dans un réseau local. De plus, l'architecture cloudlet proposée fournit également un cadre qui est responsable de la gestion et de la distribution des applications basées sur les composants. Ces applications ont généralement des exigences strictes en temps réel. En dépit de nombreux avantages de l'architecture proposée, tels que l'exécution rapide d'applications à forte intensité de calcul et l'analyse rapide des données, plusieurs défis en matière de déploiement, de calcul et d'ordonnancement doivent encore être pris en considération.

Les auteurs **(Maly F, et al)** ont proposé la solution Ad-hoc mobile basé sur le cloud computing appelé m-cloud. En raison de l'ouverture du code source de la plate-forme Android, la solution est implémentée. L'application (Proof-of-concept) peut télécharger des modules dynamiquement à partir d'un serveur, puis il est possible de les exécuter. Le m-Cloud proposé permet aux appareils mobiles d'utiliser les technologies mobiles dans les situations d'urgence. En outre, une politique de sécurité a également été introduite pour éviter le téléchargement et l'exécution de codes malveillants. Malgré les nombreux avantages du travail, le manque de mise en œuvre complète et l'évaluation rigoureuse sont quelques-unes des limites qui seraient étudiées dans le futur, comme cela a été mentionné dans l'étude.

Solutions proposées	Mise au point spécifiée	Avantages	Inconvénient
C-Protocol **(Zaghdoudi B, et al)**	gérer et déployer un p2p cloud mobile sur MANET	Ubiquité, disponibilité, Abordabilité, spontanéité	Manque de régimes incitatifs
Transient cloud **(Penner T, et al)**	Permet aux périphériques mobiles proches de former un réseau mobile ad hoc et de partager leurs ressources en tant que nuage	Activer l'exécution des tâches à forte intensité de calcul de manière distribuée	Mise en œuvre partielle, manque de régimes incitatifs
S-CMA **(Ordonez-Morales EF, et al)**	Permet aux utilisateurs de prêter les ressources à partir d'un cluster ad hoc d'appareils mobiles	Calcul remarquable, Minimiser les coûts de communication, Faible latence du réseau	Négligence du facteur de mobilité
Ad hoc Cloudlet **(Chi F, et al)**	Proposer une architecture de jeu basée sur ad hoc (Cloudlet)	Solution alternative pour l'infrastructure, Moins d'environnement	Cher en raison des incitations
A fine-grained Cloudlet architecture **(Verbelen T, et al)**	permettre aux utilisateurs de dynamiquement à partir du Cloudlet en trouvant les périphériques mobiles avec les ressources disponibles dans un réseau local	Exécution rapide, analyse données rapides	Programmation inefficace, calcul complexe
Ad hoc mobile cloud **(Maly F, et al)**	Permet aux périphériques mobiles pour former un ad hoc mobile nuage	Formation sécurisée	Mise en œuvre partielle, manque d'évaluation rigoureuse

Table 0-3 Comparaison de la solution proposée basée au sein de la formation du MAC.

2.4. Confidentialité et sécurité

Les auteurs **(Lacuesta R, et al)** ont proposé un algorithme de confiance qui permet au réseau de cloud mobile Ad-hoc spontané sécurisé. L'algorithme assure la fiabilité et la sécurité des nœuds responsables de la transmission et de la communication dans le MAC. En outre, l'algorithme permet également de gérer le mécanisme de liaison et de sortie du nœud. L'algorithme est basé sur le cryptage AES qui utilise une fonctionnalité de gestion de clé simple. L'algorithme peut assurer la communication sécurisée entre les nœuds formant MAC. Le seul problème de l'algorithme est le retard provoqué par un mécanisme de cryptage par rapport à celui sans procédure de sécurité.

(Gong Y, et al) ont étudié les problèmes de confidentialité et a proposé un cadre qui garantit la confidentialité de l'emplacement tout en allouant la tâche aux appareils mobiles dans MAC. Le cadre est basé sur la confidentialité différente et la répartition géographique qui permet aux périphériques de partager leurs ressources en nuage mobile Ad-hoc en assurant la confidentialité des informations de localisation. En outre, des modèles analytiques et des stratégies d'allocation de tâches ont été développés. Le cadre proposé ne garantit pas seulement la confidentialité sans affecter la qualité des services, mais réduit également les frais généraux du système dans MAC. Malgré de nombreux avantages du cadre proposé, tels que la sécurité et de la confidentialité de localisation, les frais généraux bas du système et la qualité du service, l'intégrité et la confidentialité des données d'utilisateur sont les préoccupations restantes.

Un système de gestion de confiance (TMC) pour les nuages mobiles Ad-hoc a été proposé par **(Hammam A, et al)**. L'objectif du système proposé est d'empêcher les nœuds mobiles malveillants de participer à un cloud mobile Ad-hoc. TMC a construit PlanetCloud qui a été introduit en termes d'informatique omniprésente. Il surveille les nœuds une fois que le nuage Ad-hoc est formé et identifie les nœuds bons et mauvais en examinant leur comportement. Après avoir observé le comportement, TMC calcule la valeur de confiance et l'enregistre dans le dépôt de nuages. Les périphériques mobiles valident la valeur de confiance à partir des valeurs stockées, puis permettent à tout nœud mobile de participer au cloud mobile Ad-hoc. Le nœud avec une plus grande valeur de confiance sera

considéré comme plus fiable et sécurisé. Le seul problème associé à TMC est la surcharge de stockage de valeur de confiance dans le nuage.

(Mandal S, et al) Ont proposé une solution qui autorise l'établissement et la distribution de clés par paire pour les périphériques participant au MAC. L'objectif de cette étude était de permettre une communication sécurisée entre les appareils mobiles formant le MAC. Les résultats ont démontré que la solution a réduit jusqu'à 75 % du nombre de SekGens requis pour établir des clés dans le MAC par rapport aux schémas naïfs non optimisés. Cependant, le schéma proposé repose sur une approche centralisée au lieu d'une distribuée, ce qui entraîne un manque de support d'exécution parallèle.

L'auteur **(Idowu IO, 2012)** à expliquer comment la qualité du service peut être améliorée pour les systèmes d'infrastructure critiques utilisant un modèle probabiliste. Le modèle peut aider à détecter les vulnérabilités, à synchroniser les capteurs mobiles en utilisant ad hoc et à sécuriser les réseaux dans le cloud computing. Le modèle proposé garantit une qualité élevée des services dans la protection des infrastructures critiques en déployant SaaS et PaaS dans le cloud computing. En outre, le modèle proposé aide à surveiller et à prédire le comportement des nœuds sans fil aux utilisateurs mobiles. Le modèle proposé peut offrir de nombreux avantages tels que la détection et la reconnaissance fiables d'une condition qui peut permettre d'atténuer le risque de protection critique du système. Le seul problème du modèle proposé est sa mise en œuvre et son évaluation qui semblent très complexes.

Solutions proposées	Mise au point spécifiée	Avantages	Inconvénient
TMC **(Hammam A, et al)**	Prévenir Les nœuds mobiles malveillants de participer à MAC	Assurance confiance, fiabilité, sécurité	Frais généraux de stockage de valeur de confiance
Location privacy **(Gong Y, et al)**	Assurez la confidentialité de l'emplacement tout en allouant la tâche aux appareils mobiles dans MAC	Garantie de confidentialité de l'emplacement, frais généraux bas du système, haute qualité du service	Manque de La confidentialité et Intégrité des données
Trust Assurance **(Lacuesta R, et al)**	Assurer la fiabilité et la sécurité des nœuds responsables de la transmission et de la communication au MAC	Communication sécurisée	Retard dans la communication
A probabilistic model for vulnerability detection **(Idowu IO, 2012)**	Détecter les vulnérabilités et synchroniser les capteurs mobiles en utilisant des réseaux bayésiens ad hoc et sécurisés dans le cloud computing	Détection fiable, et reconnaissance de condition	Mise en œuvre complexe
A pairwise key establishment scheme **(Mandal S, et al)**	Activer par paire établissement clé pour les appareils participant au MAC	Minimisation du nombre d'exécutions SekGens, et communication sécurisée	Manque de support d'exécution parallèle

Table 0-4 Étude comparative des solutions proposées par la confidentialité et sécurité.

2.5. Incitations et mobilité

Les auteurs **(Miluzzo E, et al)** ont présenté une vision de la nouvelle architecture MCC appelée MAC et ont également tracé ses premières considérations de conception. Le concept du fournisseur de cloud et des nœuds de client en nuage a également été introduit. De plus, le programme d'incitation avec des exemples mathématiques a également été proposé. Les auteurs ont envisagé que les appareils mobiles pourront former un cloud Ad-hoc en raison de leurs capacités de traitement et de mémoire élevée. L'objectif de l'étude était de donner une vision aux chercheurs que, à l'avenir, le MAC constituera un nouveau paradigme informatique permettant aux utilisateurs d'exécuter leur tâche à forte intensité de calcul dans l'environnement dynamique et sans infrastructure.

(Huerta-Canepa G, et al) ont proposé un cadre de cloud virtuel qui permet à l'informatique MAC d'utiliser les appareils mobiles dans le voisinage local. Le cadre détecte

les nœuds disponibles dans une zone spécifique en vérifiant sa stabilité et son modèle de mobilité et forme un nuage virtuel qui permet aux appareils mobiles d'exécuter des tâches complexes. L'architecture du cadre proposé comprend cinq composants, à savoir le gestionnaire d'applications, le gestionnaire de ressources, le gestionnaire de contexte, le composant P2P et le gestionnaire de déchargement. Le travail était une porte préliminaire et ouverte pour des recherches futures en termes de gestion des tâches (lorsque le nœud exécutant la tâche quitte soudainement le cloud) et la sélection du nœud mobile sécurisé pour l'exécution du travail.

Un schéma de répartition de la charge de travail a été proposé **(Truong-Huu T, et al)** qui considère le caractère aléatoire du temps de connexion parmi les dispositifs coopérants en adoptant une approche de programmation stochastique à plusieurs étages. Le système permet aux périphériques mobiles de former un MAC et de répartir leur charge de travail les uns avec les autres en considérant le facteur de mobilité. Une fois que le cloud Ad-hoc est formé et que la charge de travail est distribuée aux périphériques voisins, des difficultés surviennent lorsque les nœuds fournisseurs peuvent se déplacer hors de portée avant d'envoyer les résultats au nœud source. Pour faire face à ce problème, on utilise une approche de programmation stochastique qui permet une prise de décision optimale. De plus, pour obtenir une répartition optimale de la charge de travail, des paramètres tels que la capacité informatique, la bande passante du réseau et les contraintes énergétiques ont été proposés. Les résultats de l'évaluation montrent que l'approche stochastique permet non seulement une répartition optimale de la charge de travail, mais traite également l'aléatoirement du temps de connexion qui se produit après la distribution de la charge de travail. Néanmoins, pour motiver les utilisateurs à participer à l'exécution de tâches à forte intensité de calcul, des mécanismes incitatifs doivent être fournis.

Les auteurs **(Tang L, et al)** ont proposé et analysé un mécanisme d'enchères doubles où chaque utilisateur qui veut exécuter sa tâche et le fournisseur qui est prêts à partager ses ressources de périphérique peut soumettre une offre, bien que la fonction de prix de la ressource de demande et la fonction de ressource à la source. Malgré de nombreux avantages du mécanisme proposé, par exemple attractif et nominal, le mécanisme d'appel d'offres entraîne une consommation d'énergie inutile. **(Yousafzai A, et al)** ont proposé un cadre basé sur les répertoires pour suivre les récompenses et les évaluations de

récompense (en termes d'énergie économisée et consommée) pour les appareils même après leur passage d'un environnement Ad-hoc à un autre. Le cadre proposé peut aider à motiver les utilisateurs mobiles à partager leurs appareils dans l'environnement MAC. Toutefois, la participation du tiers qui surveille les récompenses et les évaluations de récompenses peut susciter des problèmes de gestion, de confidentialité et de sécurité.

Solutions proposées	Mise au point spécifiée	Avantages	Inconvénient
Stochastic programming approach **(Truong-Huu T, et al)**	Répartir la charge de travail en considérant le L'aléatoire du temps de connexion entre les Coopérant en adoptant un Multi-étages	Distribution optimale de la charge de travail, aléaudité du temps de connexion	Niveau faible du QoS
Virtual cloud **C**omputing **(Huerta-Canepa G, et al)**	Répartir la charge de travail en considérant le caractère aléatoire de la connexion chronométré parmi les Coopérant en adoptant un multi-étages Approche de programmation stochastique	Modèle de mobilité, et stabilité du nœud	Retard important lors de la formation du ad hoc cloud
Incentive scheme **(Miluzzo E, et al)**	motiver l'utilisateur mobile à opter pour une participation MAC	Nominal, véridique	Il n'est pas conçu en considérant la rationalité des utilisateurs mobiles individuels.
Directory-based Architecture **(Yousafzai A, et al)**	Suivre la rétribution et évaluations de récompenses	Fournir une motivation à l'utilisateur pour participer au MAC	La participation de tiers peut susciter des problèmes de sécurité et de confidentialité
Double-sided bidding mechanism **(Tang L, et al)**	Facilite l'utilisateur et le fournisseur en fournissant un mécanisme d'enchères doubles	Attrayante, nominale	Consommation d'énergie inutile

Table 0-5 Comparaison des incitations et des solutions proposées basées sur la mobilité.

2.6. La gestion des ressources

Un cadre MAC multihop a été proposé par **(Malhotra A, et al)**. Le cadre améliore l'utilisation des ressources et fait également face aux problèmes d'évolutivité et de connectivité dans MAC. Le multihop MAC comprend trois types de nœud, à savoir les nœuds mobiles (CN), les nœuds de consommation et le nœud de matchmaker (MN). Si le CN tente à exécuter une tâche, il envoie d'abord une demande à MN qui conserve la liste des nœuds de consommation disponibles ; de cette façon, un nuage est formé. Le MN est généralement un houblon multiple du CN qui peut dégrader les performances de la tâche en termes de performance globale du système. L'outil proposé a abordé ce problème en divisant le réseau Ad-hoc en clusters statiques et dynamiques. Alors que l'ancien divise le cluster en une taille fixe, celui-ci se divise en tailles dynamiques. Le cadre proposé peut aider à résoudre de nombreux problèmes liés à l'évolutivité, l'espace mémoire et les capacités de traitement. Le seul inconvénient du travail est le temps supplémentaire nécessaire pour sélectionner la tête de cluster.

Les auteurs **(Khalifa A, et al)** ont proposé PlanetCloud qui peut fournir un support intrinsèque pour un MAC hautement mobile et hétérogène. En outre, il permet au MAC d'adapter les variations dynamiques en temps réel dans son infrastructure sous-jacente en isolant les problèmes de gestion du code et du matériel. PlanetCloud est alimenté par une couche d'application qui est responsable de l'encapsulation des applications en nuage et permet une exécution sûre et fiable dans l'environnement MAC hétérogène des ressources. Les résultats d'évaluation de la plate-forme PlanetCloud montrent qu'il peut très bien fonctionner en termes de temps d'exécution avec très peu de migrations de machines virtuelles, même si un grand nombre de nœuds ont quitté le MAC. Malgré de nombreux avantages de PlanetCloud tels que le temps d'exécution rapide et les retards de retard minimum, la complexité est une préoccupation majeure.

Solutions proposées	Mise au point spécifiée	Avantages	Inconvénient
MMADC **(Malhotra A, et al)**	améliorer l'utilisation des ressources et faire face aux problèmes d'évolutivité et de connectivité dans MAC	Scalabilité, espace mémoire, capacités de traitement	Sélection de la tête de grappe cause un gaspillage de temps
PlanetCloud **(Khalifa A, et al)**	Fournir un soutien intrinsèque Pour très mobile Et des MAC hétérogènes	Exécution des tâches rapide, et Surcharge de retard minimum	Complexité

Table 0-6 Comparaison des solutions proposées basées sur la gestion des ressources.

3. Synthèse

Nous avons étudié les différents efforts de recherche dans le domaine MAC. Nous avons analysé plusieurs problèmes entravant l'adoption du MAC et analysé les solutions correspondantes en concevant une classification. Nous avons comparé les solutions proposées existantes en soulignant les avantages et les inconvénients.

En conséquence, nous avons conclu que le MAC est à son stade initial de développement, et il a certains défis à attendre pour résoudre, comme la consommation d'énergie et la confidentialité des utilisateurs ect..

4. Conclusion

Le MAC permet l'exécution d'un appareil mobile en ligne sans être connecté sur internet sans fil. Des solutions et des plates-formes ont été proposées et des statistiques mentionner pour les traiter et réaliser une application avec ces conditions ; sous l'Android OS.

Dans le chapitre suivant, nous allons présenter notre application et les résultats obtenus avec une discussion qui apparaitre ces performances.

CHAPITRE 03

Conception et Implémentation

1. Introduction

Ce chapitre donne l'aperçu de l'étude et fournit des aspects fondamentaux de l'étude tels que les composants et les scénarios. En outre, la structure et l'architecture de la plate-forme sont étudiées et les différentes fonctions seront discutées.

2. Motivation

L'objectif principal de cette étude est de créer un système simple et fonctionne sans aucune manipulation supplémentaire dans le périphérique mobile. De ce point de vue, Système d'exploitation Android a été choisi car il offre API approprié. Ce système a main plus forte, compte tenu de sa popularité et le fait que la plupart des téléphones intelligents sur le marché à l'aide du système d'exploitation Android.

3. Environnement de travail

3.1. Matériel

a. Acer Laptop V3 – 771G :

- **Processeur** : Intel i7-3632QM 2.20GHz
- **Ram** : 16 GB
- **Carte Graphique** : Nvidia Geforce 710M
- **System** : Windows 10 Pro 64-Bit

3.2. Logiciel

Android Studio est préféré comme un environnement de développement intégré (IDE) parce que c'est l'environnement officiel pour la plate-forme Android.

4. Architecture

Le projet comprend 2 paquets et 7 classes. Les classes de base telles que **GPSActivity.java**, **JoinActivity.java** et **CreateActivity.java** sont regroupées sous **com.maci.shm.Client** et **com.maci.shm.Server** package respectivement alors que l'activité principale et les classes de diffusion (**MainActivity.java**, **P2PBroadcastReceiver.java**) sont regroupées à partir du paquet principal, il existe deux autres paquets qui sont relativement moins réactifs, les packages **com.maci.listener** et **com.maci.values** incluent des interfaces afin de simplifier la facilité d'accès aux méthodes d'établissement de connexion et aux valeurs statiques du projet, respectivement. Enfin, le package **com.adhoc.service** est formé pour gérer les méthodes de transfert de fichiers et d'audio. Chaque classe de ce paquet contient différentes sous-classes responsables de la connexion socket, de l'enregistrement audio, de la lecture de la piste et du stockage des fichiers.

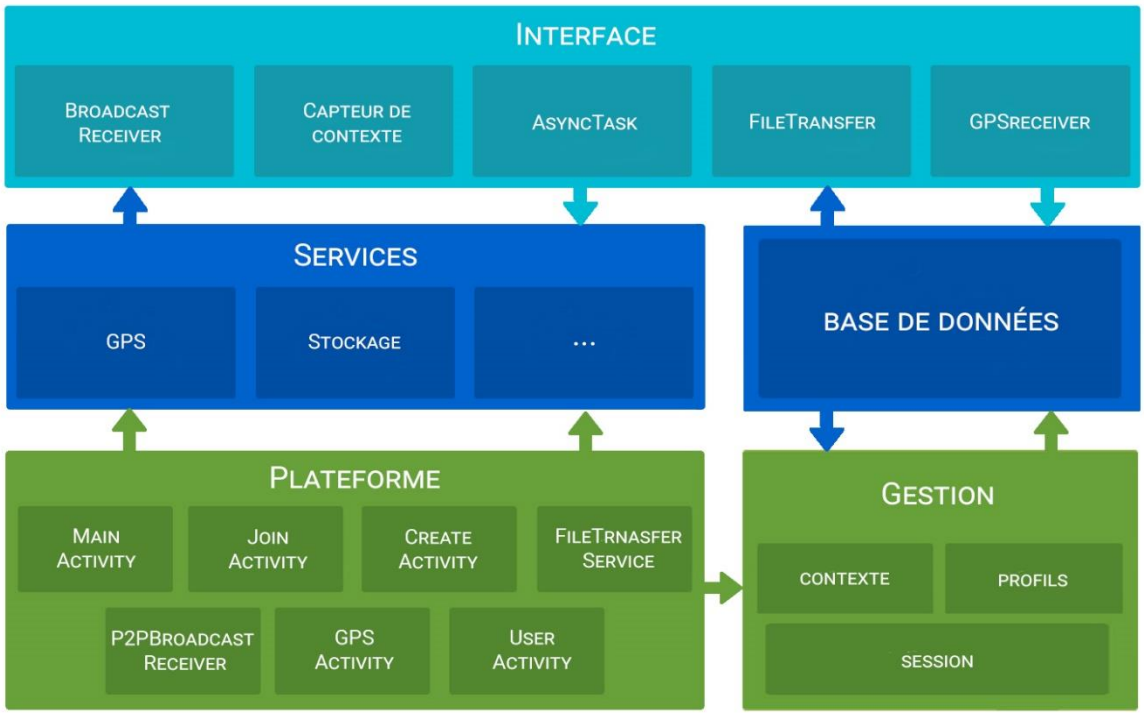


Figure 0-1 Architecture du notre plateforme

L'application commence par créer des récepteurs de radiodiffusion instanciation. Ces récepteurs contrôlent brièvement les changements dans les pairs P2P, la connexion P2P, l'état P2P et le statut P2P du périphérique hôte. Ceci est nécessaire pour garder le contrôle de la session pendant ou avant la connexion. La description détaillée concernant les récepteurs de radiodiffusion est fournie dans cette section. Ensuite, l'interface utilisateur et la gestion de l'interaction de l'utilisateur définissent toutes les sélections nécessaires.

À la suite de la procédure d'activité et de l'intent du Brodcast Receiver, les activités de base se déroulent lorsqu'elles gèrent toutes les manipulations et les méthodes, tant dans la liste des pairs que dans l'établissement de la connexion. Les opérations de base de l'ensemble du système sont effectuées avec ces deux classes, JoinActivity.java et CreateActivity.java

Après toutes les négociations, les découvertes par les pairs et les configurations de connexion, les classes de service trient la gestion des fonctionnalités de transfert de données de fichiers et de GPS. Ces services sont appelés lorsque la connexion est établie entre les périphériques et que l'utilisateur choisit une de ces fonctionnalités et s'arrête après l'opération de déconnexion. Par conséquent, ils fonctionnent essentiellement en arrière-plan et attendent simplement la connexion disponible.

La découverte par les pairs et le transfert de données sont maintenus, comme il est visualisé dans la figure 3-2. Les deux pairs peuvent exécuter la méthode "Discover Peer" et celui qui invoque la méthode est affecté comme client alors qu'un autre est un serveur. Une fois que le client et le serveur sont affectés, la connexion de la demande du client s'effectue via la méthode « Demande de connexion ». La méthode envoie une demande à un autre serveur partagé et attend une réponse. Si le résultat est "True", le pair se connecte automatiquement au serveur et se tient debout pour transférer des fichiers ou recevoir des données GPS.

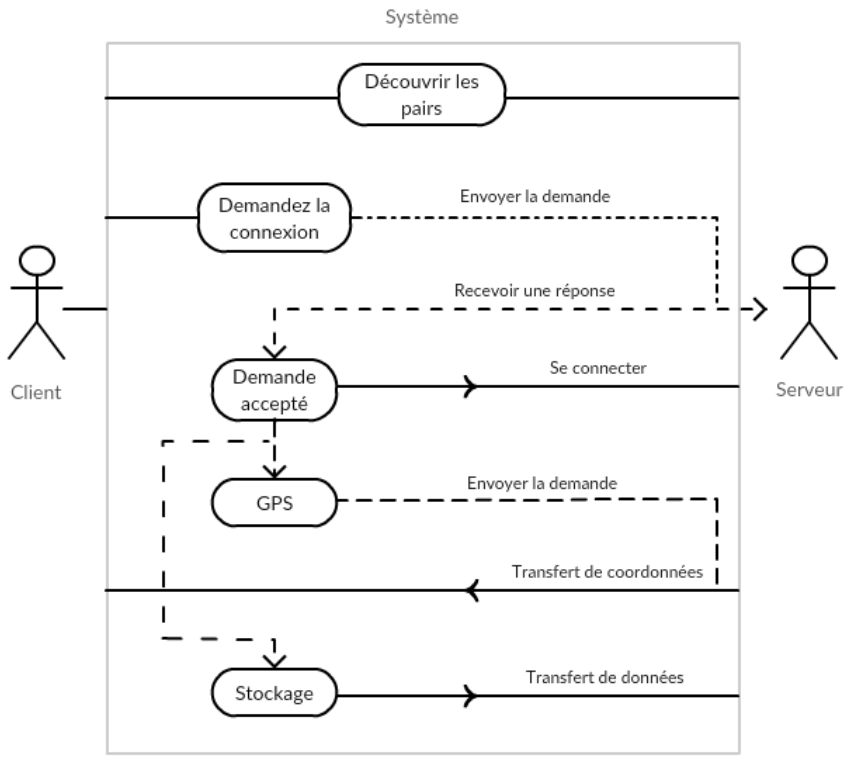


Figure 0-2 Diagramme de cas d'utilisation

5. Conception et Implementation

5.1. Conception

5.1.1. Scenarios

La motivation pour MAC peut être exprimée par l'exemple de la participation en conférence en temps réel, où un utilisateur mobile prend une vidéo du locuteur ou du participant et enregistre toutes les informations importantes dans le but de documenter le discours. En outre, les smartphones ont une capacité de stockage limitée et peuvent être remplis surtout si la vidéo a été prise en HD. Dans ce scénario, l'utilisateur mobile peut ne pas disposer de ressources suffisantes, telles que la capacité de stockage pour effectuer cette tâche sur son appareil individuel. Par conséquent, il a décidé de former un MAC pour effectuer la tâche spécifiée en temps réel. Supposons qu'il y ait 5 sessions, 50 participants dans chaque session, et 10 d'entre elles sont prêtes à participer à MAC. De plus, tous les participants ont une capacité de 8Gb minimum dans leurs smartphones personnels.

L'utilisateur mobile peut enregistrer 40 min de vidéo HD avec une vitesse d'enregistrement de 24 Mb/s sans utiliser MAC, mais le problème se produit lors de l'enregistrement de toutes les sessions, il nécessite en principe 80Gb pour enregistrer toutes les 5 sessions si nous supposons qu'une session prend 80 min, ce qui signifie 400 min d'enregistrement pour les 5 sessions. Avec l'aide de MAC qu'il a formé, il peut obtenir 8Gb * 10 nœuds, cela signifie 80Gb de stockage pour l'aider à accomplir la tâche.

Ce scénario démontre clairement comment l'adoption de MAC peut étendre les ressources d'un utilisateur en exécution en temps réel. Cependant, il reste un certain nombre de défis à relever. Les données qui seront stockées dans d'autres périphériques appartenant à différents participants au MAC sont vulnérables, elles peuvent être supprimées ou modifiées par la personne participée. C'est un problème majeur qui doit être adressé si nous voulons protéger nos données.

5.1.2. **Composants**

Afin de fournir une vue détaillée concernant le mécanisme du système, le projet peut être regroupés en trois segments, Voici les Broadcast Receiver, AsyncTask.

a. **Broadcast Receiver :**

Broadcast Receiver permet à l'utilisateur de recevoir des Intents diffusées par le système Android, afin que l'application puisse répondre aux événements auxquels vous êtes intéressé **(Android, 2012)**. Une Intent est une description abstraite d'une opération à effectuer. Son utilisation la plus significative est le lancement d'activités, où il peut être considéré comme la colle entre les activités. Il s'agit essentiellement d'une structure de données passive contenant une description abstraite d'une action à exécuter **(Android, 2012)**. En utilisant un récepteur de diffusion, l'utilisateur peut effectuer des opérations lorsqu'il est reçu par sa méthode **onReceive ()**.

Dans ce projet, il est utilisé pour recevoir des intentions rouges de la classe d'activité, JoinActivity.java. Lorsque l'activité commence, le récepteur est enregistré pour être exécuté dans le thread d'activité principale en utilisant registerReceiver (rBroadcast, intentionFilter) ; la méthode rBroadcast sera appelée avec l'intention de diffusion qui correspond à Intent-Filter dans le thread d'application principal. Ici, l'Intent consiste en différentes actions

précédemment attribuées et prêtes à être diffusées lorsque des événements se produisent, telles que la découverte par les pairs ou les changements d'état d'un périphérique. Liste des actions et tout le processus est traité plus loin dans ce chapitre.

b. **AsyncTask :**

Les opérations de connexion et de transfert de données sont effectuées par les classes JoinActivity.java, GPSActivity.java et StorageActivity.java à travers les méthodes :

```
public static class FileServerAsyncTask extends
AsyncTask<> {...}
public static class GPSsyncAsyncTask extends
AsyncTask<> {...}
```

Le processus AsyncTask fournit un moyen simple de maintenir un processus d'arrière-plan sans travailler sur d'autres détails tels que les threads et les boucles de messages. Ses méthodes de rappel aident les utilisateurs à planifier les tâches et à actualiser l'interface utilisateur (UI). Lorsqu'une tâche est exécutée, elle passe par 4 étapes. Cependant, pour utiliser AsyncTask classe, au moins une méthode, **doInBackground ()**, doit être remplacée. Les méthodes sont :

doInBackground()	Cette partie s'exécute dans un thread séparé de l'interface utilisateur et comprend tout le code que l'application exécute. Il est appelé après onPreExecute () et avant onPostExecute ().
onProgressUpdate()	Permet la méthode doInBackground () de transmettre les données au thread UI.
onPreExecute()	Appelé avant que le thread ne commence à fonctionner. Il est utilisé pour configurer une tâche qui doit être effectuée à l'avance.
onPostExecute()	Fonctionne après le thread d'arrière-plan. Il prend des résultats de la méthode doInBackground ().

Dans ce projet, AsyncTasks sont invoqués sur le côté serveur - **public void onConnectionInfoAvailable ()** {...} méthode dans JoinActivity.java class, une fois qu'un

propriétaire de groupe est affecté. Toutes les tâches entrent dans un processus d'arrière-plan où elles attendent la connexion de socket provenant de services.

Les services sont conçus pour ouvrir, relier un socket client et établir une connexion entre le serveur. Lorsqu'une connexion est établie, les fichiers et les données GPS sont stockées et copiées dans un flux (Stream) où les données réelles sont transférées. En raison de sa caractéristique native, il fonctionne en arrière-plan et attend une opération de transfert possible. Ils s'appellent immédiatement après avoir appuyé sur le bouton correspondant.

Les services et les AsyncTasks sont utilisés dans cette étude pour des raisons particulières. Par définition, le service est utilisé s'il est nécessaire d'exécuter tout processus long en arrière-plan. Le partage de fichiers et les données GPS sont maintenus à l'aide du service afin de ne pas bloquer tout processus de premier plan et de garantir que l'opération ne sera pas interrompue. Les services sont les meilleures pratiques lorsqu'il existe des services pour des opérations critiques, comme le partage ou le transfert de données. D'autre part, AsyncTasks fournit un moyen approprié de maintenir le traitement d'arrière-plan sans gérer les détails de bas niveau, tels que les threads, les boucles de messages. Les méthodes de rappel sont utiles pour planifier les tâches et mettre à jour l'interface lorsqu'elle est requise. Puisque le partage et le transfert de données sont effectués essentiellement isolés liés à l'interface utilisateur, l'utilisation de AsyncTask est la meilleure pratique.

5.2. Implementaion

Le projet est conçu dans quatre parties où chaque partie est responsable de différents aspects. Essentiellement, MainActivity vérifie si WiFi-Direct est désactivé ou non, JoinActivity.java gère les méthodes d'instanciation et le processus de pré-connexion pour l'API WiFi-Direct, tel que la découverte par les pairs, les opérations de connexion / déconnexion et le statut. Il comprend également des méthodes standard du cycle de vie d'Android, ex. OnPause (), onResume () et onDestroy (). En particulier, les fonctions onDestroy () et onKeyDown () gèrent le processus de post-configuration pour configurer l'appareil pour une utilisation régulière. Tuer les services ou les mettre en pause peut être nommées comme ces processus.

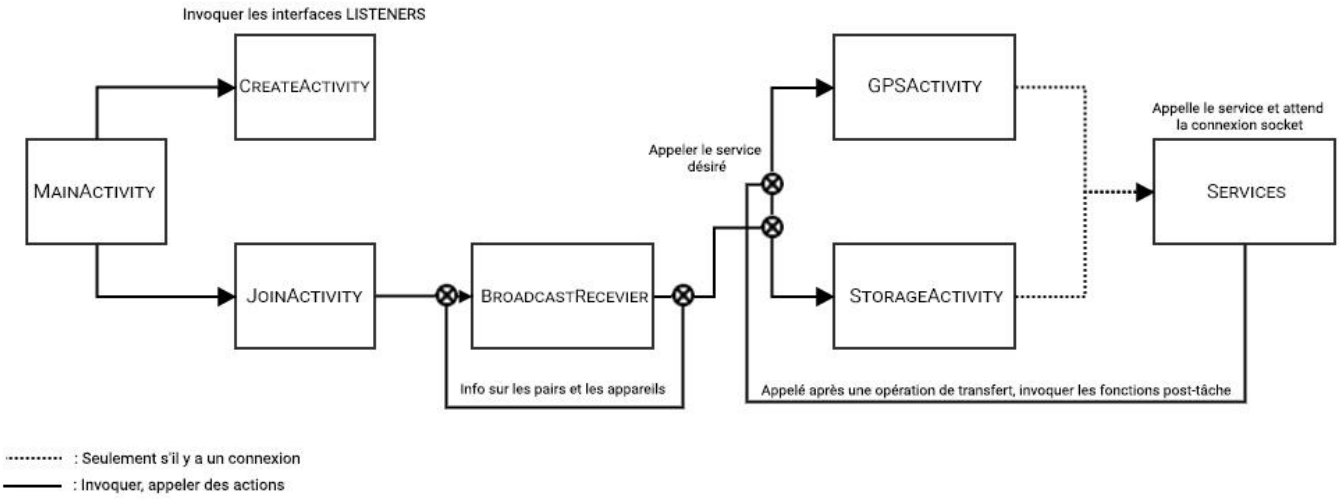


Figure 0-3 Aperçu du système

Comme le montre la figure ..., les deux classes d'activités GPSActivity.java et StorageActivity.java sont principalement appelées dans la classe BroadcastReceiver afin de gérer le service choisi en ce qui concerne les détails du pair et du périphérique, ou récupérer les informations de connexion avec l'état du périphérique. À cet égard, les classes mettent en œuvre des auditeurs et sont donc associées à des méthodes supplémentaires. On peut le comprendre plus clairement en vérifiant leurs auditeurs (Listeners) ci-dessous.

```
public class JoinActivity extends
AppCompatActivity implements
WifiP2pManager.ConnectionInfoListener {…}

public class JoinActivity extends
AppCompatActivity implements
WifiP2pManager.PeerListListener {…}
```

Ici, **WifiP2pManager.ConnectionInfoListener** est associé à la méthode **requestConnectInfo ()**, et **WifiP2pManager.PeerListListener** est associé à la méthode **requestPeers ()**.

5.2.1. **MainActivity**

MainActivity se compose de deux méthodes get et set, à savoir comme :

- **setWifiP2pEnables(boolean)** : True, pour activer le Wifi-Direct. False pour le desactiver.
- **isWifiP2pEnabled()** : Retourner Wifi-Direct statut s'il est activé ou désactivé.

Le rôle de l'activité principale est de déterminer quel service a choisi l'utilisateur, soit un client (récepteur), soit un serveur (l'expéditeur). Lorsque l'utilisateur fait son choix, l'activité vérifiera l'état du Wifi-Direct, si elle est activée, une Intent est invoquée pour l'emmener à l'interface suivante, si elle est désactivée, une boîte de dialogue apparaîtra pour indiquer à l'utilisateur que le WiFi-direct est désactivé et s'il veut l'activer.

D'autre part, les méthodes de rappel sont déclarées afin d'effectuer des opérations entre les états d'activité. Le cycle de vie de l'activité, c'est-à-dire les états d'activité, est un aspect important du système d'exploitation Android où il garantit le contrôle du processus d'activité. Dans la figure ..., le diagramme d'activité est présenté pour saisir brièvement le concept. La classe MainActivity comprend seulement trois de ces rappels, à savoir comme onCreate (), onPause () et onResume ().

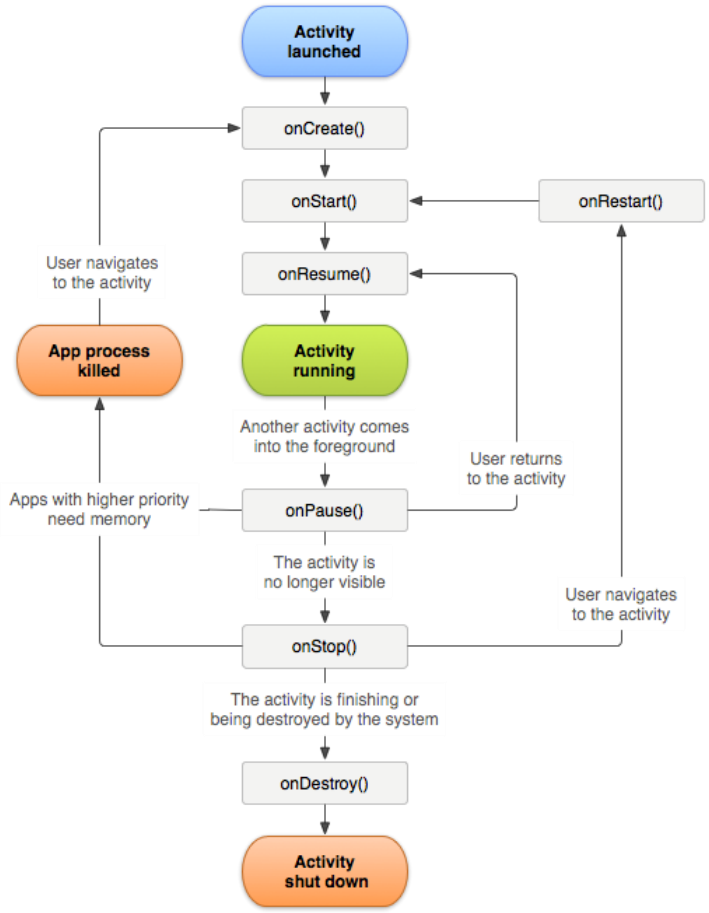


Figure 0-4 Une illustration simplifiée du cycle de vie de l'activité. (Android, 2017)

5.2.2. **JoinActivity**

Cette activité contient les différentes procédures et fonctions pour lancer le service P2P / Wifi-Direct, quatre méthodes surchargées telles que **onPause ()** et **onResume ()**, venant d'**android.app.Activity** API les deux autres provenant des fonctions **WifiP2pManager**, cinq méthodes créées explicitement, à savoir comme :

resetData ()	Pour redémarrez les données lorsque la connexion est terminée
Connect (WifiP2pdevice)	Pour établir une connexion entre les deux appareils en fournissant la configuration de la connexion
Disconnect ()	Pour terminer la connexion établie
getDeviceStatus (int)	Pour retourner l'état de l'appareil, soit l'appareil disponible ou connecté…
thisDevice (WifiP2pdevice)	Récupérer les informations du mobile exécutant l'application

Dans cette activité, la méthode **onCreate ()** détient un méthode intérieur particulier, **initialalizeWifiDirect ()** pour initialiser Wifi-Direct.

```
private void initializeP2P(){
    Mmanager = (WifiP2pManager)
getSystemService(Context.WIFI_P2P_SERVICE);
    Mchannel = manager.initialize(this,
getMainLooper(), null);
}
```

Ici, manager.initialize (...); La méthode enregistre l'application avec un structure Wi-Fi. C'est la première fonction appelée avant l'exécution des opérations P2P. Suite à l'initialisation de P2P, l'Intent est créée et ajoutée où BroadcastReceiver vérifie.

```
//Initiating P2P Broadcasts.
intentFilter
        .addAction(WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION);
intentFilter
        .addAction(WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION);
intentFilter
        .addAction(WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION);
intentFilter
        .addAction(WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION);
```

Ces actions sont incluses dans le filtre correspondant. Comme description, ce sont :

WIFI_P2P_PEERS_CHANGED_ACTION	Diffusé lorsque l'état de la connexion WiFi de l'appareil change
WIFI_P2P_CONNECTION_CHANGED_ACTION	Diffusez quand **discoverPeers ()** est appelé. Il est également nécessaire d'appeler **requestPeers ()** afin d'obtenir la liste mise à jour des pairs
WIFI_P2P_STATE_CHANGED_ACTION	Diffusé lorsque WiFi-Direct est activé ou désactivé
WIFI_P2P_THIS_DEVICE_CHANGED_ACTION	Diffusé lorsque les détails de l'appareil ont changé

Les méthodes **OnPause ()** et **onResume ()** sont simplement pour l'enregistrement et le non-enregistrement du récepteur de diffusion lorsque l'utilisateur retourne à l'activité, ou une autre activité arrive au premier plan. Ces méthodes sont : **registerReceiver (mBR, intentionFilter)** pour **onResume ()** et **unregisterReceiver (mBR)** pour **onPause ()**.

En outre, JoinActivity.java consiste en une autre méthode supplémentaire pour gérer et contrôler l'animation de recherche, la méthode appelée **mAnim ()**. Cette méthode fait une transition entre les éléments de l'interface et crée un arrière-plan ondulé pour rendre l'UI plus facile à utiliser.

Enfin, il existe deux méthodes supplémentaires dans cette classe qui sont chargées de l'exploitation de la connexion et de l'organisation des détails des périphériques / pairs. Les principaux sont **connect (..)** et se **disconnect ()**. Comme leurs noms le suggèrent, ils

établissent et annulent la connexion entre les pairs. La fonction **connect ()** prend (WifiP2pDevice device) comme paramètre pour crée un WifiP2pConfig qui représente une configuration WiFi-P2P pour la configuration d'une connexion.

```
public void connect(WifiP2pDevice device) {

    WifiP2pConfig config = new WifiP2pConfig();
    config.deviceAddress = device.deviceAddress;
    config.wps.setup = WpsInfo.PBC;

    manager.connect(channel, config, new
       WifiP2pManager.ActionListener() {

        @Override
        public void onSuccess() {
            // WiFiDirectBroadcastReceiver will notify us. Ignore for
              now.
            Toast.makeText(act, "Config created succesfully",
            Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onFailure(int reason) {
            Toast.makeText(JoinActivity.this, "Connect failed. Retry.",
            Toast.LENGTH_SHORT).show();
        }
    });
}
```

Manager.connect (channel, config, nouvelle fonction WifiP2pManager.ActionListener () {...} lance une connexion P2P à un périphérique avec la configuration spécifiée. L'application est notifiée par un succès ou un échec lorsque la demande est retournée. Par ailleurs, la fonction **disconnect()** supprime le groupe P2P et appelée la méthode **resetData ()**. En appelant cette méthode, les données vont réinitialiser.

```
public void resetData() {
    transition = null;
    mIcon = (ImageView)
        findViewById(R.id.foundDevice);
    mName = (TextView) findViewById(R.id.DeviceName);
    mIcon.setVisibility(View.INVISIBLE);
    mName.setVisibility(View.INVISIBLE);
    mName.setText("");
    mPeers.clear();
}
```

De plus, après la réinitialisation de toutes les données, manager.removeGroup () supprime le groupe P2P actuel. Identique à la fonction **connect ()**, l'application est notifiée ici à nouveau pour lancer le processus via les rappels d'écoute **onSuccess()** ou **onFailure(..).**

```
public void disconnect() {
    resetData();
    manager.removeGroup(channel, new WifiP2pManager.ActionListener() {

        @Override
        public void onFailure(int reasonCode) {
            Toast.makeText(JoinActivity.this, "Disconnection failed",
            Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onSuccess() {
            Intent i = new Intent(JoinActivity.this, MainActivity.class);
            startActivity(i);
            finish();
        }
    });
}
```

5.2.3. **P2PBroadcastReceiver**

Cette classe vérifie les intentions que l'utilisateur s'intéresse à la méthode **onReceive()**. Les Intents qui sont définies dans cette classe sont expliquées brièvement dans la section du composant (5-1-2). Cette classe est la mise en œuvre d'un BroadcastReceiver typique où elle effectue les actions nécessaires lorsqu'une intention est reçue.

```
public class P2PBroadCastReceiver extends BroadcastReceiver {

    private WifiP2pManager manager;
    private WifiP2pManager.Channel channel;
    private JoinActivity activity;

    public void setActivity(JoinActivity activity) {
        this.activity = activity;
    }


    public P2PBroadCastReceiver(WifiP2pManager manager, WifiP2pManager.Channel
channel) {
        super();
        this.manager = manager;
        this.channel = channel;
         }

    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();

        if (WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION.equals(action)) {
            // Determine if Wifi P2P mode is enabled or not, alert
            // the Activity.

        } else if (
        WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION.equals(action)) {
            //Available peers on the network
            manager.requestPeers(channel, activity);

        } else if(
        WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION.equals(action)) {
            // Connection state changed!
            NetworkInfo networkInfo = (NetworkInfo) intent
                    .getParcelableExtra(WifiP2pManager.EXTRA_NETWORK_INFO);

            if (networkInfo.isConnected()) {
                manager.requestConnectionInfo(channel, activity);
            } else if (activity != null) {
                activity.resetData();
            }

        } else if (
        WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION.equals(action)) {
            if (activity != null) {
                activity.thisDevice((WifiP2pDevice)
        intent.getParcelableExtra(WifiP2pManager.EXTRA_WIFI_P2P_DEVICE));
            }
        }
    }
}
```

Chaque condition vérifie les actions d'Intent et exécute le code associé à l'action souhaitée. La deuxième condition est responsable d'appeler la méthode manager.requestPeers() afin de récupérer la liste des pairs actuels. La classe JoinActivity.java est invoquée à cet égard car elle est associée à l'interface WifiP2pManager.PeerListListener.

La troisième condition vérifie si l'état de la connectivité Wi-Fi P2P a changé. Si oui, les informations réseau sont extraites à l'aide de la classe NetworkInfo. Une fois la connectivité réseau existante et possible d'établir des connexions, la classe JoinActivity.java est invoquée par rapport à sa méthode associée **onConnectionInfoAvailable ()** à l'interface WifiP2pManager.ConnectionInfoListener.

Enfin, une action Intent est émise indiquant si les détails de l'appareil ont changé. La mise à jour des détails du périphérique, l'annulation des tâches courantes ou l'effacement de tous les champs sont gérés ici selon l'état WiFi-Direct du périphérique.

5.2.4. Transfert de données et services

Le transfert de données est établi par les services. Ces services servent de client et se connectent au socket du serveur avec un socket client et éventuellement transfert des données. Deux services sont responsables de l'opération, FileTransferService et GPSService.

a. **FileTransferService :** La classe étend IntentService qui comprend une méthode appelée **onHandleIntent ()**. Cette méthode est appelée sur le thread de travail avec une requête à traiter. Après le processus d'initialisation, le socket est ouvert et le processus de liaison commence. La méthode **socket.bind (null)** lie ce socket à l'adresse hôte et au port local spécifiés. Ici, en raison de l'adresse locale est défini comme nul, ce socket sera lié à une adresse locale disponible sur n'importe quel port gratuit (Android, 2012). Lors de ce processus, le socket se connecte à l'adresse et au port hôte de l'hôte donné.

```
String host = intent.getExtras().getString(EXTRAS_GROUP_OWNER_ADDRESS);
int port = intent.getExtras().getInt(EXTRAS_GROUP_OWNER_PORT);
Socket socket = new Socket();
...
socket.connect((new InetSocketAddress(host, port)), SOCKET_TIMEOUT);
```

L'adresse et le port de l'hôte spécifiés sont envoyés à partir de la classe parent JoinActivity.java comme valeurs statiques.

Après avoir récupéré l'Uri de données dans StorageActivity.java et la crypter avec la fonction **encryptFile (...)**, les données sont copiées pour diffuser en utilisant la même méthode sur le côté serveur, **copyFile (is, stream)**. Ici, les données sont écrites dans le flux de sortie via le socket et donc le transfert est terminé. Le processus final ferme simplement la prise et définit le message de contrôle par rapport au type de demande.

b. **GPSService :** Essentiellement, il a la même fonction que StorageActivity.java mais sans cryptage, GPSStorage.java recapture les coordonnées GPS (Longitude, Latitude) et invoque le service de transfert de fichiers qui le mettra dans un flux et l'envoie comme les données précédentes.

5.2.5. Cryptage et decryptage:

Comme nous l'avons déjà dit dans notre scénario, les données peuvent être compromises et endommagées par d'autres utilisateurs. Afin de les protéger, nous proposons une solution qui crypte les fichiers que nous enverrons aux autres. Les fichiers seront cryptés par **encryptFile (...)** méthode en générant une clé utilisée pour le cryptage, la clé est générée par la méthode **generateKey (..)**. Cette clé est générée à partir de l'adresse MAC du smartphone, donc il sera unique, après avoir généré une clé, **encryptFile (...)** l'utilisera comme paramètre pour crypter les fichiers.

```
public static byte[] generateKey(byte[] macPass) throws Excepti
{
    KeyGenerator mGenrator = KeyGenerator.getInstance("AES");
    SecureRandom sRandom = SecureRandom.getInstance("DATA",
                                                    "Crypting");
    sRandom.setSeed(macPass);
    mGenerator.init(128, sRandom);
    SecretKey mKey = mGenerator.generateKey();

    return mKey.getEncoded();
}

public static byte[] encryptFile(byte[] key, byte[] Data) throw
Exception
{
    SecretKeySpec mSpec = new SecretKeySpec(key, "AES");
    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.ENCRYPT_MODE, mSpec);
    byte[] dProtected = cipher.doFinal(Data);

    return dProtected;
}
```

Le décryptage est simple, lorsque les données sont reçues, l'activité récupère l'adresse Mac et l'utilise comme clé pour décrypter les fichiers.

```
public static byte[] decryptFile(byte[] key, byte[] cryptedData)
throws Exception
{
    SecretKeySpec mSpec = new SecretKeySpec(key, "AES");
    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.DECRYPT_MODE, mSpec);
    byte[] mDecrypt = cipher.doFinal(cryptedData);

    return mDecrypt;
}
```

6. Résultat

Le but du projet est d'établir une connexion entre les appareils mobiles et de maintenir le transfert de fichiers et de données sans utiliser de service 3G ni de point d'accès. En plus de cela, une tâche de sécurité a été effectuée afin de protéger les données envoyées.

En utilisant wifi-direct et ses différentes fonctionnalités a rendu le développement de l'application plus efficace et agile. Les connexions p2p sont maintenues entre les téléphones Android et la découverte par les pairs a réussi, et surtout le transfert de données est réalisé. De plus, la connexion est établie avec succès parmi deux appareils. Bien que les nombreux utilitaires de l'étude tels que le transfert de données, la découverte par les pairs et la connexion, il présente également peu d'inconvénients en ce qui concerne le transfert de données. Le projet ne fournit pas une solution satisfaisante, car la consommation d'énergie n'a pas été incluse dans le projet.

En conséquence, le projet garantit une solution pratique au problème indépendamment des problèmes mineurs.

7. Discussion

Il existe essentiellement deux problèmes qui restent le problème possible et doivent être résolus. En outre, il existe un aspect qui n'est pas un problème majeur, mais peut être considéré comme un inconvénient mineur. Cet inconvénient concerne essentiellement les

messages fournis à l'utilisateur en cas de problème. Les messages peuvent être revus afin de maintenir une sortie conviviale.

Les deux questions ainsi que des informations supplémentaires concernant le problème lui-même :

- **Éliminer les groupes** : le bouton de déconnexion est enfoncé, les groupes sont effectivement supprimés, mais le menu indique toujours que le pair est toujours connecté. Ce n'est pas seulement une mise en page, mais aussi un problème de découverte qui fonctionne essentiellement à travers les Broadcast Receiver. Ainsi, lorsque le groupe est supprimé, le système recherche automatiquement des pairs et découvre le pair instantanément. Par conséquent, la liste des pairs n'est pas actualisée et la reconnexion ne peut être établie que lorsque l'application est redémarrée.
- **Les pairs(nœud)** : Un autre problème concerne davantage la classe WiFiP2pManager. Après une connexion réussie et un transfert de données possible, les pairs restent connectés pendant un temps limité, soit environ 5 minutes. Pendant cette période, les pairs peuvent se connecter - déconnecter et transférer des données à plusieurs reprises. Cependant, l'opération de découverte doit être invoquée à nouveau pour la recherche de pairs après 5 minutes. Cela entraîne une incohérence dans le système et restreint la capacité de connexion constante.

En ce qui concerne le problème de fonctionnalité, l'application a des bugs, parfois il est difficile de détecter une paire car le mobile est déjà connecté à un autre ou n'est pas disponible, et cela crée des problèmes pour l'application.

8. Conclusion

Cette étude visait à résoudre le problème de l'échange de données avec un groupe de personnes où la connexion à un point d'accès au réseau ou 3G n'est pas toujours possible dans des domaines particuliers. Ce problème est résolu avec cette étude via une application mobile utilisant WiFi-Direct. Par conséquent, l'objectif est atteint avec cette application où la connexion par les pairs peut être obtenue.

Conclusion

Des progrès immédiats dans MCC ont ouvert la voie à un nouveau paradigme informatique appelé MAC. Bien qu'il existe de nombreuses études de recherche sur le cloud computing mobile et l'informatique ad hoc, la convergence de ces deux domaines accorde d'autres efforts académiques pour l'épanouissement du MAC.

Dans cette thèse, nous avons étudié les différents aspects de Computing domaine et nous avons étudié l'état de l'art du MAC et les différentes recherches dans ce domaine, nous avons analysé plusieurs problèmes entravant l'adoption du MAC et analysé les solutions correspondantes en créant une classification.

En plus de cela, nous avons développé une plate-forme pour démontrer le réseau MAC. Nous avons proposé avec succès une solution pour la confidentialité des utilisateurs.

Bibliographie

Chen M, Hao Y, Li Y, Lai C-F. On the computation offloading at ad hoc cloudlet: architecture and service modes. IEEE Communications Magazine 2015.

Chi F, Wang X, Cai W, Leung VCM. Ad hoc cloudlet based cooperative cloud gaming. In Cloud Computing Technology and Science, 2014 IEEE 6th International Conference on, IEEE, 2014.

Dan M. On android compatibility. Android developers blog. 2010

Debashi De, Mobile cloud computing: Architectures, Algorithmes and Applications, 2016.

developer.android.com. BroadcastReceiver. Available at: http://developer.android.com/reference/android/content/BroadcastReceiver.html [Accessed August 2012]

Gong Y, Zhang C, Fang Y, Sun J. Protecting location privacy for task allocation in ad hoc mobile cloud computing. IEEE Transactions on Emerging Topics in Computing 2015.

Guo X, Liu L, Chang Z, Ristaniemi T. Data offloading and task allocation for cloudlet-assisted ad-hoc mobile clouds.

H. T. Dinh, C. Lee, D. Niyato, and P. Wang, A survey of mobile cloud computing: Architecture, applications, and approaches, Wireless Communications and Mobile Computing, 2013.

Hamidreza Bagheri, Prasanth Karunakaran, Kaveh Ghaboosi, Timo Bräysy, Marcos Katz Center for Wireless Communications, University of Oulu, Finland 90014, 2012

Hammam A, Senbel S. A trust management system for ad-hoc mobile clouds. In Computer Engineering & Systems (ICCES), 2013 8th International Conference on, IEEE, 2013.

Hamza S, Aïcha-Nabila B, Okba K, Youssef A. A cloud computing approach based on mobile agents for web services discovery. 2012 Second International Conference on Innovative Computing Technology (Intech) 2012.

http://www.idc.com/promo/smartphone-market-share/os

http://www.interoute.fr/what-paas

https://fr.wikiversity.org/wiki/Cloud_computing_et_entreprise/Pr%C3%A9sentation_du_Cloud_computing

https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

Huerta-Canepa G, Lee D. A virtual cloud-computing provider for mobile devices. In Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ACM, 2010.

Idowu IO. Ad-hoc cloud networks: a probabilistic model for vulnerability detection in critical infrastructure using Bayesian networks. 2012.

J. Andrews and A. Gatherer, Will densification be the death of 5G?, IEEE ComSoc Technology News, CTN Issue, 2015.

J. Zhang and D. L. Roche, *Femtocells: Technologies and Deployment*, Wiley, Chichester, U.K.,2010.

Khalifa A, AzabM, Eltoweissy M. Towards a mobile adhoc cloud management platform. In Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, IEEE Computer Society, 2014.

Kristensen MD, Bouvin NO. Scheduling and development support in the scavenger cyber foraging system. Pervasive and Mobile Computing 2010.

Lacuesta R, Lloret J, Sendra S, Peñalver L. Spontaneous ad hoc mobile cloud computing network. The Scientific World Journal 2014.

Li B, Pei Y, Wu H, Shen B. Heuristics to allocate high performance cloudlets for computation offloading in mobile ad hoc clouds. The Journal of Supercomputing.2015.

Lu Z, Zhao J, Wu Y, Cao G. Task allocation for mobile cloud computing in heterogeneous wireless networks. In Computer Communication and Networks (ICCCN), 2015 24th International Conference on, IEEE, 2015.

M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, The case for VM-based cloudlets in mobile computing, *IEEE Pervasive Computing*, 2009.

Malhotra A, Dhurandher SK, Kumar B. Resource allocation in multi-hop mobile ad hoc cloud. In Engineering and Computational Sciences (RAECS), 2014 Recent Advances in, IEEE, 2014.

Maly F, Kriz P. An Ad Hoc Mobile Cloud and Its Dynamic Loading ofModules into aMobile Device Running Google Android, in New Trends in Intelligent Information and Database Systems. Springer, 2015.

Mandal S, Yang C, Altaweel A, Stoleru R. An efficient pairwise key establishment scheme for ad hoc mobile clouds. In Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on, IEEE, 2015.

Miluzzo E, Cáceres R, Chen Y-F. Vision : mClouds computing on clouds of mobile devices. In Proceedings of the third ACM workshop on Mobile cloud computing and services, ACM, 2012.

Miluzzo E, Cáceres R, Chen Y-F. Vision: mClouds computing on clouds of mobile devices. In Proceedings of the third ACM workshop on Mobile cloud computing and services, ACM, 2012.

MCC Lab, the Hong Kong Polytechnic University. Mobile cloud computing, 2013.

NIST. Cloud computing standards Roadmap. Special publication 500-291 V2, 2013.

Ordonez-Morales EF, Blanco-Fernandez Y, Bravotorres JF, Lopez-Nores M, Saians-Vazquez V, Pazos arias JJ. S-CMA: sporadic cloud-based mobile augmentation supported by an ad-hoc cluster of moving handheld devices and a virtualization layer. In Innovative Computing Technology (INTECH), 2015 Fifth International Conference on, IEEE, 2015.

Penner T, Johnson A, Van Slyke B, Guirguis M, Gu Q. Transient clouds : assignment and collaborative execution of tasks on mobile devices. In 2014 IEEE Global Communications Conference (GLOBECOM), IEEE, 2014.

Shi T et al. An energy-efficient scheduling scheme for time-constrained tasks in local mobile clouds. Pervasive and Mobile Computing 2016.

Subir Kumar S, Basavaraju TG, Puttamadappa C. Ad hoc wireless networks, 2008.

Tang L, He S, Li Q. Double-sided Bidding Mechanism for Resource Sharing in Mobile Cloud. IEEE Transactions on Vehicular Technology 2016.

Truong-Huu T, Tham C-K, Niyato D. A stochastic workload distribution approach for an ad hoc mobile cloud. In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on, IEEE, 2014.

Verbelen T, Simoens P, De Turck F, Dhoedt B. Cloudlets: bringing the cloud to the mobile user. In Proceedings of the third ACM workshop on Mobile cloud computing and services, ACM, 2012.

Yousafzai A, Chang V, Gani A, Noor RM. Directory based incentive management services for ad-hoc mobile clouds. International Journal of Information Management 2016.

Zaghdoudi B, Ayed HK-B, Riabi I. Ad hoc cloud as a service: a protocol for setting up an ad hoc cloud over MANETs. Procedia Computer Science 2015.

Zaghdoudi B, Ayed HK-B, Riabi I. Ad hoc cloud as a service: a protocol for setting up an ad hoc cloud over MANETs. Procedia Computer Science 2015.