Méthodes et outils d'analyse de données de signalisation mobile pour l'étude de la mobilité humaine

Alexis Sultan

Spécialité : Informatique et Réseaux

Ecole doctorale : Informatique, Télécommunications et Electronique de Paris

Présentée par

Alexis SULTAN

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

Méthodes et outils d'analyse de données de signalisation mobile pour l'étude de la mobilité humaine

Soutenue le 28/09/2016

Devant le jury composé de :

Rapporteurs :

MM. :	**Marco FIORE**	**Chercheur à l'IEIIT, Turin**
	Olivier FLAUZAC	**Prof. à l'Univ. de Reims Ch-Ardenne**

Examinateurs :

Mme. :	**Houda LABIOD**	**Prof. à Télécom ParisTech**
M. :	**Farid BENBADIS**	**Chercheur chez Thales Comm. & Security**

Encadrants :

MM:	**Hossam AFIFI**	**Prof. à Télécom SudParis**
	Vincent GAUTHIER	**Maître de conf. à Télécom SudParis**

N° NNT : 2016TELE0018

Remerciements

Je tiens tout d'abord à remercier Marco FIORE, Chercheur à l'Instituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni, et Olivier FLAUZAC, Professeur à l'université de Reims Champagne-Ardenne, d'avoir accepté la charge de rapporteur.
Je suis reconnaissant à Houda LABIOD, Professeur à Télécom ParisTech, et Farid BENBADIS, Chercheur chez Thales Communications & Security de bien avoir voulu juger ce travail.
Je remercie également Hossam AFIFI, Professeur à Télécom SudParis, et Vincent GAUTHIER, Maître de conférence à Télécom SudParis, de m'avoir accueilli et d'avoir, respectivement, dirigé et encadré mes travaux au sein de TSP pendant ces quelques années.
Je souhaite aussi remercier Alain HAMEL et Stéphane GUINDOLET de m'avoir proposé de réaliser cette thèse dans leur service et de m'avoir fourni les moyens de mettre en place la plateforme décrite dans ce document. Merci à Olivier PASSIEN.
Merci aux membres du laboratoire SAMOVAR ; à Monique BECKER et Michel MAROT. Merci à Fereshteh ASGARI, Nicolas GENSOLLEN, Emad ABD-ELRAHMAN et à Seif-Eddine HAMMAMI.
Un grand merci à Alessio DEIANA et à Mustafa ÖZVEREN pour le travail qu'ils ont fourni ainsi qu'aux équipes de *développements internes* et *services mobiles avancés*.

De même, merci à Marie-Emmanuelle de m'avoir permis (involontairement) de réaliser cette thèse.
Merci à Guillaume et à Lazhar de m'avoir aidé tout au long de ce parcours, et à Benjamin et François pour leurs relectures et corrections.
Merci à mes amis pour leur soutien, leurs conseils, encouragements et pour leur cuisine. Merci à tous ; aux randonneurs, aux nageurs rhénans, à ceux du vendredi soir et à ceux du vendredi matin ; merci à ceux de la rue du Château-des-Rentiers et du Dahomey, à Orgerus, aux Antoniens et Villejuifoise, aux Mulhousiens, Hambourgeois, Dortmundois et Aalenois.

Finalement, je souhaite remercier ma famille ; merci à mes grands-mères, mes parents, mes tantes et oncles. Merci à ma sœur et à Guillaume, merci à mes cousines et cousins.

Abstract

Cette thèse a pour but d'étudier les activités humaines à travers l'analyse du flux de signalisation du réseau cellulaire de données (GTP). Pour ce faire, nous avons mis en place un ensemble d'outils nous permettant de collecter, stocker et analyser ces données de signalisation. Ceci en se basant sur une architecture indépendante au maximum des constructeurs de matériel. À partir des données extraites par cette plateforme nous avons fait trois contributions.

Dans une première contribution, nous présentons l'architecture de la plateforme de capture et d'analyse de la signalisation GTP dans un réseau d'opérateur. Ce travail a pour but de faire l'inventaire des différents éléments déclenchant des mises à jour et aussi d'estimer la précision temporelle et spatiale des données collectées. Ensuite, nous présentons une série de mesures, mettant en avant les caractéristiques principales de la mobilité humaine observées au travers de la signalisation mobile (le temps inter-arrivées des messages de mise à jour, la distance observée des sauts entre cellules lors des déplacements des clients). Finalement, nous présentons l'analyse des compromis qui ont été faits entre la rapidité d'écriture/de lecture et la facilité d'usage du format de fichier utilisé lors de l'échange d'informations entre les sondes de capture et le système de stockage.

Deuxièmement, nous avons été capables de mettre en place un algorithme de reconstitution de trajets. Cet algorithme permet, à partir de données éparses issues du réseau cellulaire, de forger des trajets sur les voies de transport. Il se base sur les données des trajets sous-échantillonnées et en déduit les positions du client sur les voies de communication. Nous avons mis en place un graphe de transport intermodal. Celui-ci porte sur le métro, le train et le réseau routier. Il connecte les différents points entre eux dans chacune des couches de transport et interconnecte les modes de transport entre eux, aux intersections. Notre algorithme se base sur un modèle de chaîne de Markov cachée pour placer sur le graphe les positions probables des individus entre les différentes observations. L'apport de ce travail est l'utilisation des propriétés topologiques du réseau de transport afin de renseigner les probabilités d'émission et de transition dans un modèle non supervisé. Ces travaux ont donné lieu à une publication et à un brevet.

Finalement, notre dernière contribution utilise les données issues de la signalisation à des fins de dimensionnement du réseau mobile d'opérateur. Il s'agit de dimensionner dynamiquement un réseau mobile en utilisant les bandes de fréquences dites

TV-Whitespace. Ces bandes de fréquences sont libérées sous certaines conditions aux USA et soumises à vente aux enchères. Ce que nous proposons est un système basé sur un algorithme de qualité d'expérience (QoE) et sur le coût de la ressource radio afin de choisir où déployer des femtocells supplémentaires et où en supprimer en fonction des variations de population par unité d'espace.
En conclusion, cette thèse offre un aperçu du potentiel de l'analyse des metadata de signalisation d'un réseau dans un contexte plus général que la simple supervision d'un réseau d'opérateur.

1 Introduction

L'étude de la mobilité humaine est exploitée dans différentes disciplines telles que l'analyse de la propagation de maladies contagieuses, le dimensionnement et l'optimisation de réseaux d'opérateurs, le dimensionnement des infrastructures de transport ou encore en sociologie. La miniaturisation et la démocratisation des moyens de mesure et de télécommunication ainsi que l'apparition de plateformes de traitement distribué de données volumineuses ont contribué à l'engouement récent pour ce domaine. À cela s'ajoute l'apparition des *smartphones* ces dix dernières années. Leur connexion quasi permanente au réseau mobile et à Internet permet depuis les réseaux d'opérateurs d'observer en continu l'activité et les mouvements des populations. Ces traces, une fois anonymisées, deviennent un matériau riche pour la recherche et l'étude de la mobilité humaine.
Les avancées technologiques nous permettent également de stocker et d'analyser de grands volumes d'informations. À titre d'exemple, SK Telecom [74], grâce à une architecture distribuée constituée de plus de 1400 nœuds, collecte 250 téraoctets de données par jour et mène des analyses en temps réel sur ces flux. Ces progrès nous autorisent à étudier sur plusieurs mois les variations de motifs de déplacements de groupes d'usagers d'un réseau, d'observer l'apparition de dysfonctionnements ou d'anomalies et de mettre en parallèle les comportements dans différentes zones géographiques. Mais cette inflation de données pose de nouvelles problématiques quant à la capture, au stockage et à l'analyse qui doivent être pensées dans leur globalité.
Nous préciserons dans ce chapitre le contexte et les problématiques de l'étude de la mobilité basée sur des données issues du réseau de téléphonie mobile, nous décrirons les données disponibles (dans ce contexte) et nous détaillerons le protocole que nous analysons, les points du réseau mobile où nous le capturons et les éléments susceptibles de nous fournir des informations sur la mobilité humaine. Enfin, nous présenterons les difficultés abordées de même que nos contributions.

1.1 Contexte et problématiques

Cette thèse s'inscrit dans le contexte de l'étude de la mobilité humaine. Les dernières études sur ce sujet sont basées sur des jeux de données hétéroclites. Parmi ces travaux se distinguent deux catégories de mesures ; les mesures actives et passives.

Nous trouvons dans la première catégorie des travaux utilisant des données issues d'applications pour téléphones mobiles spécialement développées pour l'étude ou des applications de type réseau social [75, 82, 95]. Ces applications permettent à l'utilisateur de signaler manuellement sa présence dans un lieu donné - un restaurant, son domicile ou son université par exemple. Les données résultantes ont l'avantage d'être labellisées, de décrire le type du lieu dans lequel le participant se trouve et de contenir un aspect social qui permet de caractériser le participant ainsi que ses relations avec le reste de la population de l'expérience. La limitation de ces mesures est qu'elles ne couvrent qu'une petite partie de la population ; les participants à l'expérience ou les membres du réseau social. Par ailleurs, elles dépendent fortement de l'implication des utilisateurs et de leur sérieux à déclarer leur position après chaque déplacement.

Les mesures passives ne nécessitent aucune action des participants. Elles proviennent par exemple de systèmes de transport, dont la richesse va de mesures faites par GPS pour des taxis [138], à des vecteurs origine-destination dans le cadre de vélos en partage [144]. De même, elles peuvent être extraites d'infrastructures de télécommunications, répertoriant la position des appelants et des appelés en se basant sur les journaux d'appels [47, 66] (Call Detail Record, CDR) d'opérateurs mobiles. Le spectre de ces mesures varie donc de données précises pour une population et un usage assez restreint, dans le cadre des taxis, à des mesures temporellement et spatialement pauvres pour une population importante, pour les CDR, mais dont l'avantage est de fournir des informations sur les liens sociaux des individus.

Les principaux bénéfices de la mesure passive sont qu'elle couvre l'ensemble des usagers d'un service et qu'elle n'a aucun impact sur le terminal du client. Celui-ci n'a pas d'application à lancer et l'autonomie de son équipement ne souffre pas de la mesure. En revanche, la précision des mesures peut être pauvre. Dans le cadre d'études réalisées sur les réseaux de télécommunication, la précision géographique est faible et s'arrête, au mieux, à l'aire de couverture radio de l'antenne à laquelle le client est connecté. La fréquence de génération de données peut elle aussi être basse et dépend du flux observé. Nous aurons dans le cas des CDR à attendre qu'un appel soit émis pour qu'une entrée apparaisse dans nos traces. Si l'on observe les informations de signalisation, il faudra qu'un utilisateur se déplace suffisamment pour qu'un message signalant sa connexion à une nouvelle cellule se propage et que les éléments de géolocalisation soient mis à jour. Il est également à noter que la collecte de tels volumes de données pose des problèmes relatifs au respect de la vie privée, nécessitant la mise en place de méthodes d'anonymisation. Ces contraintes sont autant de défis pour nos travaux.

Enfin, une caractéristique de cette thèse est son caractère industriel : son sujet et ses applications sont conditionnés par l'opérateur qui m'emploie et par le laboratoire auquel je suis rattaché. Ainsi, les outils mis en place devront permettre également de répondre à des attentes telles que l'optimisation du réseau, son dimensionne-

ment, la détection d'anomalies et ils devront permettre de mesurer l'impact des modifications réseau.

1.2 Le réseau mobile et ses protocoles de signalisation

Un réseau mobile opérateur est une source précieuse de données pour l'analyse de l'activité humaine. Le réseau couvre l'ensemble du territoire national, le parc utilisateur est très important et les équipements des abonnés sont maintenus allumés de manière quasi permanente. La source principalement utilisée dans les récents travaux sur la mobilité humaine est le journal d'appel. Il permet de connaître le lien entre les clients - qui appelle qui, qui écrit à qui - et de connaître la position géographique des deux parties au moment de la communication. L'avantage du lien social est aussi un inconvénient. Nous n'avons d'information que lorsqu'il y a un échange, ainsi la fréquence d'information est potentiellement basse. Nous étudierons ici une autre source d'informations issue du réseau mobile : la signalisation du protocole du cœur de réseau de données, le GTP-C. D'autres travaux étudiant la mobilité utilisent GTP-C, mais, à notre connaissance, tous nécessitent d'autres informations et aucun ne se base que sur ces seules données.

Les informations de géolocalisation transitent par différents points du réseau mobile et pour chacun de ces points le protocole utilisé est différent, l'architecture de capture change et la complexité d'analyse varie. Certains protocoles sont utilisés dans des sites répartis dans tout le pays alors que d'autres transitent par moins de dix points sur tout le territoire.
La pertinence du GTP-C en termes de densité d'information et de facilité d'analyse n'est pas le seul critère dans notre choix. Des raisons industrielles sont également à considérer. Le service de l'opérateur chez lequel cette thèse a été réalisée a accès à ce flux et il est difficile, dans une entreprise de taille importante, de placer des équipements et de capturer des flux sur des réseaux sortant de ses attributions.
Nous décrirons rapidement les sous-ensembles du réseau mobile puis nous ferons un état des lieux des protocoles de signalisation que nous pouvons y collecter. Cet état des lieux a pour but de comparer les structures des différents protocoles, et ainsi que la complexité du déploiement de sondes pour l'analyser, et de décrire brièvement la difficulté que présenterait l'extraction de données.

1.2.1 Le réseau mobile et ses sous-ensembles

Pour faciliter la compréhension de la suite du manuscrit, nous décrivons succinctement l'organisation du réseau mobile et ses différents sous-ensembles.
Le réseau mobile est composé du réseau d'accès radio (Radio Access Network, RAN) et du cœur de réseau (Core Network, CN). Le premier est proche de l'abonné et réparti sur tout le territoire. Il a la responsabilité de la gestion de la ressource radio et d'une partie de la gestion de la mobilité. Le second est centralisé, il héberge des services tels que l'authentification, la gestion des appels et sert de passerelle entre le réseau de l'opérateur et le réseau téléphonique commuté (pour le *domaine circuit*) et Internet (pour le *domaine paquet*).
Une autre séparation est faite dans le cœur de réseau : les flux de données et les flux voix empruntent deux chemins différents. On parlera de *domaine paquet* pour le premier et de *domaine circuit* pour le second. Il est à noter que cette distinction n'est plus vraie en LTE où la donnée et la voix empruntent les mêmes canaux.

1.2.2 Dans le cœur du domaine paquet : le GTP

Le GPRS Tunneling Protocol (GTP pour la 2G et 3G [3], GTPv2 pour la 4G [2]) est le protocole fournissant et gérant, dans le cœur de réseau, la connectivité des terminaux mobiles à Internet.
Le GTP comprend deux protocoles distincts. Le premier, appelé GTP-U ou *user plane*, transporte le trafic utilisateur. Le second, appelé GTP-C ou *control plane*, permet aux sessions GTP-U de s'établir, y met fin et les tient à jour pendant toute la durée de leur vie.
Le GTP-C, décrit en Figure 1.1, est transporté par le User Datagram Protocol [102] sur le port 2123. Il est, par conséquent, simple à discriminer du trafic de données en se basant sur le numéro de port.

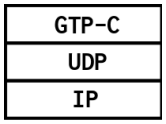


Figure 1.1 : Couches protocolaires de la signalisation GTP.

Les échanges sont organisés sous la forme de requêtes-réponses, identifiés par des numéros de séquence au niveau GTP-C. Les informations à l'intérieur de la couche GTP-C sont structurées en type-longueur-valeur (Type-Length-Value, TLV).
L'utilisation d'UDP et l'identification des séquences d'échange, au niveau GTP, ne nécessitent pas le développement d'une machine à état complexe pour suivre les dialogues. Le protocole ne fait appel qu'à une seule spécification de la 3GPP et se trouve directement au-dessus d'UDP. Tous ces arguments en font un protocole

simple à analyser. De plus, circulant dans le cœur de réseau, le trafic est centralisé et ne nécessite pas le déploiement d'un grand nombre de sondes.

1.2.3 Dans le cœur du réseau circuit : la voix

Nous présenterons ici, afin de comparer la complexité du GTP-C avec son équivalent dans le *domaine voix*, le protocole de signalisation voix de l'UMTS dans le cœur du réseau.
Mobile Application Part (MAP) [4] est le protocole de signalisation utilisé pour de nombreuses tâches telles que l'enregistrement des utilisateurs, leur identification et leur mobilité. Comme nous le voyons en Figure 1.2, le MAP est transporté par le Transaction Capabilities Application Part [69, 70, 71, 72, 73] (TCAP). Celui-ci permet d'établir plusieurs sessions en parallèle entre un même couple de machines. Dans le cas le plus simple, où ces flux transitent sur IP, l'empilement protocolaire

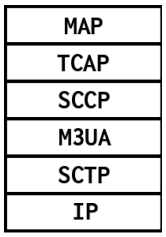


Figure 1.2 : Couches protocolaires de la signalisation voix en UMTS dans le cœur de réseau.

est tel que décrit en Figure 1.2. Le Signalling Connection Control Part [68] (SCCP) gère l'établissement des sessions sémaphores. Alors que le Message Transfer Part 3 User Adaptation Layer [67] (M3UA) est une couche d'adaptation permettant de faire transiter du Message Tranfer Part (MTP) sur IP, le MTP gère le routage entre les différents points du réseau. Enfin, la couche OSI de transport est assurée par le Stream Control Transmission Protocol (SCTP). Il s'agit d'un protocole orienté message nécessitant une initiation du dialogue entre les différentes parties.

Bien que MAP soit un protocole de cœur de réseau, ne nécessitant donc pas plus de sondes que GTP-C, son analyse est plus complexe et requiert beaucoup de développements du fait de sa structure qui contient 3 couches de plus que GTP-C.

1.2.4 En entrée du cœur de réseau

Les interfaces d'interconnexion des flux de signalisation entre le RAN et le CN sont l'Iu-Cs (Figure 1.3a) pour le domaine circuit, et l'Iu-Ps (Figure 1.3b) pour le domaine paquet. Représentés en Figure 1.3, ces flux sont riches et transportent des

données de gestion de mobilité : Mobility Management (MM) pour le domaine circuit et GPRS MM (GMM) pour le domaine paquet [5]. Ils transportent également les messages de gestion de session de données (Session Management, SM), de la ressource radio [5] (Radio Resource, RR) et la gestion des appels (Call Control, CC). Ces informations sont transportées par le Radio Access Network Application Part [6]

(a) Domaine circuit (Iu-Cs)

(b) Domaine paquet (Iu-Ps)

Figure 1.3 : Couches protocolaires sur les interfaces Iu.

(RANAP) qui gère la liaison entre les équipements de l'UTRAN et les équipements du cœur, ainsi que le paging. Interviennent ensuite, comme pour la signalisation voix (section 1.2.3), le SCCP et les couches sous-jacentes.
Nous sommes donc en présence d'un empilement protocolaire faisant intervenir deux recommandations de l'ITU-T et deux spécifications de la 3GPP, le tout transporté par un protocole nécessitant la mise en place d'une machine à état pour assurer le suivi de sessions. Nous avons, comme pour la voix, un empilement complexe. À cela s'ajoute la position de ces flux en bordure du CN, ce qui nécessite plus de sondes que dans les deux autres solutions où les flux sont agrégés.
Pour conclure, le GTP-C est de notre point de vue le protocole de signalisation du réseau mobile le plus simple à capturer et à analyser. Il convient de s'assurer de ses qualités pour l'analyse de la mobilité.

1.3 Les protocoles GTP

La signalisation GTP permet l'établissement des tunnels, leur suppression et leur mise à jour. À travers le contenu de ces messages, nous sommes en mesure de connaître, entre autres, les caractéristiques de la connexion des clients et leur géolocalisation via l'identifiant de la cellule à laquelle ils sont connectés.
L'élément du protocole nous informant de la position géographique du client est le User Location Information (ULI), il contient le code pays (Mobile Country Code, MCC) et le code réseau (Mobile Network Code, MNC) de l'opérateur. Il indique aussi le code du groupe de cellules (en 2G et 3G : Location Area Code, LAC ; en 4G : Tracking Area Code, TAC). Nous y récupérons aussi l'identifiant de l'abonné (International Mobile Subscriber Identity, IMSI), l'identité unique du terminal client

(International Mobile Station Equipment Identity, IMEI) dont le préfixe identifie le modèle et le fabricant. Finalement, le GTP-C nous renseigne sur la technologie radio utilisée (Radio Access Technology, RAT). Le GTP-U, quant à lui, est un protocole de type tunnel tel Generic Routing Encapsulation [54] (GRE) ou IP in IP [101] encapsulant le trafic IP utilisateur. À la différence des deux autres, comme nous le voyons en figure 1.4 et figure 1.5, le GTP-U est basé sur UDP. Il est à remarquer sur ces figures le rôle des SGSN et S-GW qui servent de passerelles entre le RAN et le CN.

Le GTP-U ne fournit aucun service supplémentaire particulier excepté celui d'identi-

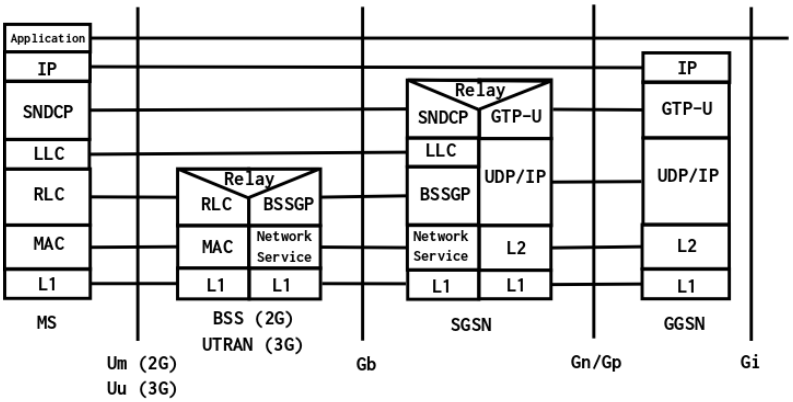


Figure 1.4 : Encapsulation GTP [3]

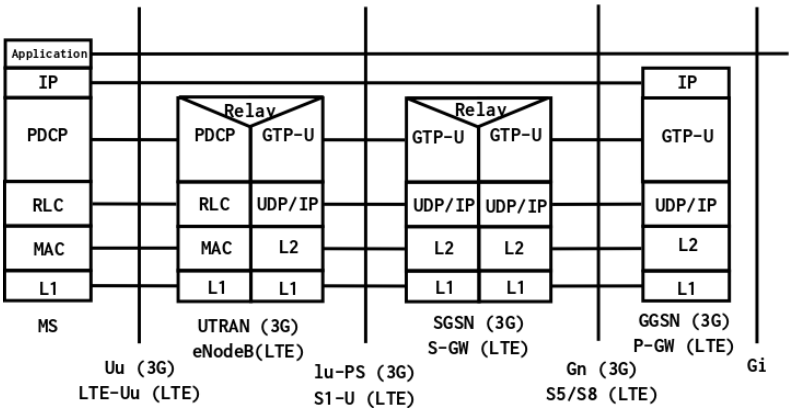


Figure 1.5 : Encapsulation GTPv2 [2].

fier le flux utilisateur par un Tunnel Endpoint IDentifier (TEID), permettant de faire le lien avec le plan de contrôle. Avec la LTE, et plus particulièrement la voix sur LTE (Voice over LTE, VoLTE), le GTP-U acquiert plus d'importance, car il transporte la voix, celle-ci n'ayant plus de réseau à part.

1.3.1 Architecture du cœur de réseau de données

Du statut de protocole uniquement *cœur de réseau*, le GTP a progressivement gagné de l'importance et du terrain dans le réseau mobile. Il fait maintenant l'interface

entre le cœur de réseau et le réseau d'accès radio. Nous présentons ici les points du réseau où il circule, les équipements qu'il y connecte et les flux qu'il y transporte.
Les différentes interconnexions où nous le trouvons se nomment interfaces et sont définies en fonction des équipements en présence et du trafic acheminé. En 2G et 3G (cf. Figure 1.6), les plans de contrôle et utilisateur transitent entre le Serving GPRS Support Node (SGSN) et le Gateway GPRS Support Node (GGSN) sur l'interface Gn. Le SGSN est l'équipement de bordure du CN qui traduira les protocoles du RAN en GTP. Le GGSN est l'équipement *ancre* des connexions. Une fois qu'une session est établie sur un GGSN, elle n'en changera pas. Le déplacement du client peut provoquer un changement de SGSN, mais le GGSN restera le même. À partir de cet équipement, les données sont *dé-encapsulées* et on ne trouve plus que le trafic utilisateur, la signalisation disparaît. Il est également à noter que le trafic des clients d'un opérateur sortira toujours sur Internet depuis le réseau de celui-ci. Ainsi le trafic du client français de l'opérateur *A* en voyage au Japon, transitera jusqu'en France pour sortir par le réseau *A*.
En Figure 1.6, nous avons mis en couleur les zones sur lesquelles le GTP circule, on remarquera que le lien entre le Radio Network Controller (RNC) et le GGSN est mis en valeur. Le RNC est l'élément contrôlant les ressources du RAN et par lequel son trafic passe. Dans le but de soulager le SGSN et simplifier le réseau, le plan utilisateur peut transiter entre le RNC et le GGSN alors que le plan de contrôle continue à se faire entre le SGSN et le GGSN. Cette technique se nomme *Direct Tunnel*. Ainsi, nous trouvons du GTP-U entre le RAN et le CN.

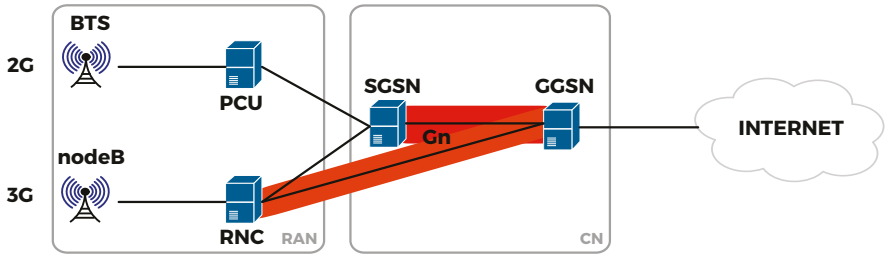


Figure 1.6 : Réseau 2G et 3G.

Le LTE simplifie encore l'architecture et rapproche le GTP du client. Comme on peut le voir en Figure 1.7, il existe une interface entre l'EnodeB et la Serving Gateway (S-GW) : la S1-U. Sur cette interface nous retrouvons le GTP-U. Le plan de contrôle passe par le S1-MME jusqu'au Mobility Management Entity (MME) (il ne s'agit pas de GTP ici, mais de S1AP [1]). Le MME gère les tâches de gestion des canaux logiques de données, de signalisation et de sécurité du terminal client. Il gère également la mobilité et s'adresse au Home Subscriber Server (HSS) pour les tâches requérant l'interrogation de la base client. Le MME dialogue avec la S-GW sur le S11. De la S-GW transitent le plan de contrôle et le plan utilisateur à destination de la PDP Gateway (P-GW) de l'opérateur via le S5 ou vers la P-GW d'un partenaire dans le cas de mobilité internationale via le S8.

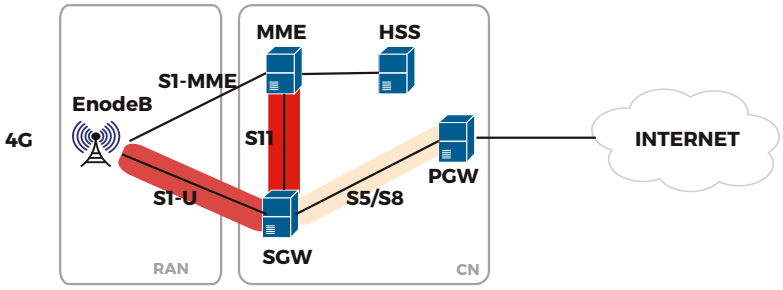


Figure 1.7 : Réseau 4G.

1.3.2 Caractérisation des messages GTP-C

La fréquence et le contenu des messages de signalisation du protocole sont des facteurs importants dans l'étude de la mobilité. Il est important de savoir à quel moment nous serons capables de collecter des informations contenant la géolocalisation des individus et quels sont les critères déclenchant les paquets de signalisation associés.
Pour chacun des tunnels actifs, les équipements traversés (SGSN/RNC - GGSN, ou EnodeB - S-GW - P-GW) stockent en mémoire une structure décrivant ces flux : le Packet Data Protocol (PDP) Context. Le PDP Context est créé, détruit et mis à jour par des messages GTP-C spécifiques.
Nous décrirons ici les messages GTP-C que nous avons décidé de stocker. Ces choix ont été faits en fonction de la présence d'éléments contenant des renseignements à propos de la mobilité du client ou des informations quant au statut de sa connexion. Nous avons aussi fait des choix par défaut, en fonction des interfaces par lesquels ces paquets passent et des interfaces auxquelles nous avons accès. Dans le but d'évaluer l'influence de l'activité des clients sur l'émission de messages de signalisation, nous énumérerons les évènements déclenchant ces messages.

Création et destruction des sessions GTP

Les deux types de messages les plus évidents sont les messages de création et de destruction de contexte. La première catégorie intervient à l'activation de la connexion de données par le mobile. Cette activation peut survenir lorsque le mobile est mis sous tension, après la perte du signal et sa récupération ou suite à une manipulation volontaire de l'utilisateur visant à activer/ré-activer la connexion de données.
Les messages de création sont des *create PDP context request* en GTP et *create session request* en GTPv2. Ils contiennent, entre autres, l'identité du client - l'IMSI, celle de son équipement - l'IMEI, les données de géolocalisation - l'ULI, ainsi que la qualité de service négociée et la technologie radio utilisée - RAT.
Les réponses à ces requêtes sont respectivement un *create PDP context response* et

un *create session response*. Elles indiquent si la requête a été acceptée et donne la cause en cas de refus. Ces messages contiennent aussi l'adresse IP attribuée à l'équipement client et l'adresse IP de l'équipement de terminaison sur lesquels établir le tunnel GTP-U.
La destruction de session intervient en cas de perte durable du signal radio, de mise hors tension de l'équipement client ou de la déconnexion volontaire de celui-ci. Mais elle peut aussi être causée par les bascules des clients d'une cellule à l'autre. Dans certains processus de bascule, où la nouvelle cellule est dans une zone géographique dépendant d'un autre GGSN (ou d'une autre S-GW) que la précédente, il est nécessaire de faire coexister deux tunnels le temps du processus et de détruire le plus ancien lorsque la bascule est effective. Un contexte peut aussi avoir à être détruit lors d'un changement de RAT si, par exemple, le client passe de la 3G vers la 4G ou inversement.
Ces requêtes sont faites par un *delete PDP context request* ou un *delete session request*. Ces requêtes ne contiennent que l'identifiant du tunnel à détruire et les réponses, *delete PDP context response* et *delete session response*, ne font qu'acquitter la demande.

Mises à jour des sessions GTP

Une fois un PDP Context établi, des messages le concernant continuent à transiter sur le réseau. Ces messages sont émis pour modifier la connexion du client, dans le but de signaler dans le cœur de réseau des modifications survenues dans le réseau d'accès radio - modification de la technologie radio, changement de zone géographique ou modification du chemin emprunté par les paquets ou encore pour notifier un équipement d'un message ou d'un dysfonctionnement. Les types de messages de mise à jour sont peu nombreux en GTP et nous n'en avons gardé qu'un. En GTPv2, nous dénombrons un peu moins de 30 messages différents. Parmi cet ensemble, nous n'avons conservé que les messages apportant des informations de géolocalisation, décrivant la qualité de la connexion ou la joignabilité du terminal.
En GTP, nous enregistrons l'*update PDP context* qui nous informe de la localisation du terminal mobile (ULI), de la technologie radio utilisée (RAT) et de la qualité de service négociée. Nous rencontrons ce message sur la Gn lorsque le mobile change de RAT, lors d'un changement de LAC, de SGSN ou de re-négociation de qualité de service. Dans le cas du *Direct Tunnel*, comme nous l'avons vu plus haut, le tunnel GTP-U est établi entre le RNC et le GGSN et la signalisation continue à transiter entre le SGSN et le GGSN. Dans ce cas, le *Direct Tunnel* a un identifiant (TEID) qui lui est propre. Si le client change de RNC, le TEID changera, ce qui donnera lieu à un *update PDP context*. De même, si le mobile passe en veille et libère son slot de données, le *Direct Tunnel* sera détruit et le nouvel identifiant de tunnel - entre le

SGSN et le GGSN - sera propagé via la Gn. Nous tenons à souligner que le champ de localisation (ULI) est optionnel dans l'*update PDP context*. Mais celui-ci est également présent dans ces messages sur le réseau de l'opérateur chez lequel cette thèse a été réalisée.
En GTPv2, comme nous l'avons annoncé plus haut, nous n'analysons qu'une partie des messages disponibles. Ces messages sont de deux catégories. Dans la première, nous trouvons des messages convoyant des informations géographiques. Nous y trouvons le message *Modify Bearer* qui circule du MME à la P-GW et ressemble à l' *update PDP context* GTP. Il intervient lors d'une modification de zone géographique (TAC), d'un passage en veille, du changement de RAT ou d'une bascule d'une cellule à une autre. Le message *Update Bearer*, quant à lui, est envoyé de la P-GW au MME dans le cas d'une demande de modification de qualité de service initiée par le CN. Ce n'est que dans la réponse du MME à ce message que nous trouvons l'ULI.
La deuxième catégorie de messages nous donne des informations sur la connectivité du mobile. Le *Downlink Data Notification* est envoyé de la S-GW au MME dans le but de ré-activer le lien vers un mobile en veille afin de lui transmettre un paquet. Le *Downlink Data Notification Failure Indication* est envoyé à la S-GW par le MME, suite à la réception du message précédent. Il est envoyé dans le cas où le terminal mobile ne répond pas ou s'il est déjà attaché au réseau. Finalement, *Release Access Bearer* est une demande explicite de libération de ressource en cas de veille. Ces derniers messages ne nous informent pas de la position du mobile, mais d'informations intéressantes : son activité ainsi que le statut de sa connexion.

Les messages et les évènements déclencheurs associés

Afin de résumer ce qui précède, nous avons regroupé dans le tableau 1.1 les différents messages que nous analysons et sauvons ainsi que les différents évènements, déclenchés par le RAN ou le CN causant ces messages.

Pour mieux identifier les évènements de l'activité humaine que nous pouvons capter en analysant cette signalisation et pour identifier les moments où le terminal mobile n'est pas joignable, donc non localisable, nous avons fait ce travail de recensement des *déclencheurs*. Un *déclencheur* est un évènement réseau, tel que le changement de RAT ou de qualité de service ; ou un évènement géographique, tel que le changement de cellule, provoquant un message de signalisation. Nous avons regroupé ces déclencheurs par catégories en fonction de la signification que le message déclenché par cet évènement peut avoir. Nous avons des évènements provoquant des messages ne nous donnant que des informations concernant la géolocalisation du mobile (jaune), d'autres ne décrivant que le statut de la connexion (bleu) et d'autres encore pouvant mettre à jour les deux catégories précédentes (vert). Ainsi, le mes-

GTP	Type de message	Déclencheurs
v1	Update PDP Context	changement de RAT
		changement de SGSN
		changement de LAC
		entrée ou sortie de veille
		négociation de la QoS
		création / destruction d'un *direct tunnel*
		changement de RNC d'un *direct tunnel*
v2	Modify Bearer	changement de TAC
		vérification périodique du TAC
		passage en veille radio
		handover vertical
		handover horizontal
v2	Downlink Data Notification Failure	le terminal ne répond pas
		le terminal est déjà ré-attaché
v2	Update Bearer	modification de QoS
		modification des ressources
v2	Downlink Data Notification	réveil d'un mobile en veille Radio
v2	Release Access Bearer	inactivité du mobile

Table 1.1 : Tableau récapitulatif des messages GTP et de leurs déclencheurs.

sage notifiant d'un changement de RAT propagera le nouveau type de technologie utilisée par la connexion, mais il mettra également à jour l'identifiant de la cellule à laquelle le client est connecté. De même, un *Update Bearer* nous apprendra que les paramètres de qualité de service ont été renégociés, mais aussi, dans la réponse, confirmera la position du mobile en communiquant l'identifiant de cellule auquel celui-ci est connecté.
Nous avons donc un ensemble de messages satisfaisants dans lequel nous trouvons des informations liées à la mobilité. Nous verrons dans les chapitres suivants si la densité temporelle et géographique est elle aussi satisfaisante.

1.4 Problèmes abordés et contributions

Les problèmes liés à cette étude découlent des volumes de données en entrée des sondes, de ceux à copier, à stocker et à analyser. Et, paradoxalement, de la faible fréquence des mises à jour des données géographiques et de l'imprécision de ces dernières.

1.4.1 Plateforme de capture, de stockage et d'analyse

Les sondes sont installées dans des sites répartis à travers le pays. La distribution des sessions GTP sur ces sites n'est pas géographique. C'est pourquoi un utilisateur ne transitera pas forcément par les équipements réseaux les plus proches de lui physiquement. De plus, les différentes sessions successives d'un même utilisateur établies dans une période donnée ne traverseront pas toutes le même site. Or nous souhaitons pouvoir traiter l'intégralité des mesures faites par zone géographique. Pour ce faire, nous pourrions distribuer nos traitements sur les différents sites et ne centraliser que les résultats. Cette solution a divers inconvénients. Au fil des traitements des résultats intermédiaires devront être échangés entre les nœuds - lorsque nous regroupons les évènements par zone géographique par exemple, le trafic ainsi généré peut avoir un lourd impact sur le réseau de l'opérateur, sans compter la lenteur provoquée par l'utilisation d'un réseau national et partagé. De plus, nous souhaitons que l'administration de notre plateforme soit facile et que le stockage des sondes ne soit pas un souci supplémentaire. C'est pourquoi nous souhaitons mutualiser l'infrastructure de stockage et être capables de la faire évoluer facilement, afin de pouvoir stocker des données sur une longue période pour étudier les évolutions du réseau, les usages et les habitudes des individus ou pour entraîner des modèles d'apprentissage supervisé. Pour toutes ces raisons, nous avons décidé de centraliser le stockage et le traitement. Ce choix, comme nous le verrons ci-dessous, a des implications sur l'ensemble de notre plateforme et des méthodes que nous allons développer.

Les sondes

Une solution ne nous demandant que très peu de développements tout en profitant d'une infrastructure centralisée aurait été de sauver les flux tels quels dans des fichiers PCAP et de les centraliser pour analyse. Les outils pour cette solution existent : écrire des fichiers dans ce format est très simple et une bibliothèque développée par le RIPE [106] permet de distribuer le traitement de fichiers PCAP sur une infrastructure HADOOP.
Mais cette solution n'est pas réalisable dans notre situation, les débits en entrée de nos sondes étant trop élevés, les écritures sur disque seraient trop importantes et les fichiers à centraliser trop volumineux. L'alternative, nécessitant des développements, est d'extraire des flux réseau les informations nous intéressant, de communiquer ces données dans un format performant (en termes de lecture, d'écriture et d'occupation disque) à une grappe de stockage et de calcul.
L'analyse de données réseau à des débits avoisinants les 20Gb/s pose des problèmes qu'il n'est pas aisé de régler avec du matériel et une architecture système et logicielle

classique. C'est pourquoi nous utilisons une carte embarquant de la mémoire vive permettant de supporter les pointes de trafic et un module permettant de filtrer les flux et de les équilibrer à destination de l'espace utilisateur. L'utilisation de cette carte nous permet de nous passer de l'allocation mémoire dynamique de la pile réseau du noyau du système d'exploitation et de n'implémenter en espace utilisateur que les protocoles nous intéressant.
Nous avons donc dû adapter la séparation classique du système d'exploitation entre ses diverses couches - du driver au noyau et à l'espace utilisateur - afin de supporter un tel débit à traiter. Le tout en développant une solution d'analyse modulaire, permettant d'analyser divers protocoles en parallèle.

Plateforme d'analyse

Nous voulons une plateforme d'analyse capable de stocker et de traiter d'importants volumes de données, fournissant une interface pratique de programmation ou d'interrogation et ayant de bonnes performances.
Les données en entrée de cette structure ont deux caractéristiques qui rendent l'utilisation d'une base de données difficile. Elles sont volumineuses - plusieurs centaines de gigaoctets par jour - et la plateforme qui les génère est expérimentale. Or, une base de données ne fonctionne bien que lorsque les données sont indexées et les champs de ses tables correctement déclarés. La modification d'un index, ou sa déclaration et le changement du type d'un champ sont des tâches demandant beaucoup de ressources sur de gros volumes. Sur un jeu de données expérimental, ces modifications sont probables et nos volumes de données sont élevés.
C'est pourquoi nous avons décidé d'utiliser HDFS pour le stockage et Spark pour l'analyse. HDFS est un système de fichier distribué et redondant permettant de stocker d'importants volumes de données et Spark un framework permettant de les interroger facilement et fournissant des bibliothèques avancées d'apprentissage et de graphe.

Contributions

— La description de cette architecture a donné lieu à une parution : Alexis Sultan et al. « Mobile Data Network Analysis Platform ». In : *Proceedings of the 6th International Workshop on Hot Topics in Planet-Scale Measurement*. ACM, 2015, p. 13–18.

— Les données extraites de cette architecture ont permis de valider un modèle de dimensionnement dynamique, basé sur un système d'enchères de bandes de fréquences : A Sultan et al. « A dynamic femto cell architecture using TV Whitespace improving user experience of urban Crowds ». In : *2015 Interna-*

tional Wireless Communications and Mobile Computing Conference (IWCMC). Août 2015, p. 886–891

— L'analyse avant industrialisation du protocole a été réalisée avec scapy et l'amélioration du support du GTP et l'ajout du GTPv2 ont été publiés : SECDEV. *Secdev/scapy*. `https://github.com/secdev/scapy`

1.4.2 Analyse de la mobilité humaine via la signalisation GTP

Comparées à un système qui récolterait fréquemment (au moins une fois toutes les minutes) les coordonnées GPS des participants à une expérience, les données extraites de notre plateforme sont loin d'être idéales, mais elles ont des avantages intéressants au niveau de la capture et du parc de la population observée.
Les principales limitations proviennent du manque de précision spatiale et temporelle de nos données. Les différentes mises à jour sont provoquées par des évènements liés à l'activité de l'utilisateur et de l'utilisation qu'il a de son équipement mobile. On peut par exemple, dans le cas du Tracking Area Update en LTE [5], rester (dans la situation la moins favorable) 54 minutes sans connaître la zone dans laquelle le client est connecté. L'autre limitation est spatiale : dans un comportement normal, l'information que nous avons n'est pas causée par une perte de signal, mais par une mise à jour classique. Nous ne connaissons que la cellule principale de connexion du client et n'aurons d'information de changement que si le mobile se connecte sur une cellule faisant partie d'un autre groupe que la précédente. De plus, lorsqu'il est connecté à une cellule, la précision géographique se réduit à l'aire de couverture de l'antenne étudiée. Ainsi la difficulté de notre travail a été de donner du sens aux informations collectées et ainsi d'être capable de suivre les déplacements géographiques d'une population.

Contributions
Un algorithme, CT-Mapper, a été développé, se basant sur les données sous-échantillonnées issues de la plateforme associées à un graphe de transport utilisant les propriétés topologiques pour calculer les propriétés d'émissions et de transition du modèle de Markov caché, permettant de placer les trajets sur les axes de communication. Ceci a donné lieu à :

— une parution : Fereshteh ASGARI et al. « CT-Mapper : Mapping Sparse Multimodal Cellular Trajectories using a Multilayer Transportation Network ». In : (2016). arXiv : `1604.06577 [cs.SI]`

— un brevet : Vincent GAUTHIER et al. « Procédé d'estimation de trajectoires utilisant des données mobiles ». Brev. 2015

1.5 Plan du manuscrit

La suite de cette thèse contient six chapitres. Le deuxième chapitre propose un état de l'art couvrant l'ensemble des thèmes abordés par la thèse, de la capture réseau de paquet à l'analyse de la mobilité. Nous présentons en détail dans le chapitre trois notre infrastructure de capture, de stockage et d'analyse. Le chapitre quatre présente des résultats obtenus grâce à notre plateforme et caractérisant les flux de signalisation et la mobilité observée depuis le réseau mobile. Nous y abordons également des techniques de filtrage nous permettant d'éliminer des points aberrants de nos trajets. Puis dans les chapitres cinq et six nous présentons des applications basées sur les données collectées. La première, CT-Mapper reconstruit des trajets sur les réseaux de transport à partir des données de signalisation. La seconde se base sur ces données pour estimer les zones où déployer des femtocells dynamiquement en fonction de la charge du réseau dans la journée. Enfin, le chapitre sept clôt cette thèse en en rappelant les contributions et en l'ouvrant vers de futurs travaux.

2 État de l'art

2.1 Introduction

L'analyse de l'activité et de la mobilité humaine peut se faire sous différents angles et dans différents buts. Elle peut chercher à caractériser les déplacements, à les modéliser ou encore à en estimer la périodicité spatio-temporelle. Ces études peuvent faire ressortir la structure des villes et des pays et estimer les éléments y influençant les mouvements. L'urbanisme, l'optimisation des moyens et des équipements de transports sont des domaines qui peuvent bénéficier de ces résultats. Une des difficultés peut être d'estimer le trafic finement et de placer correctement les traces sur les axes de communication. Cette problématique dépend des traces étudiées, traces qui ont subi des modifications ces dernières années.

Les évolutions matérielles - téléphones et outils de mesures, les avancées techniques des infrastructures de transports et de télécommunications, et finalement les innovations concernant la collecte, le stockage et l'analyse des données ont modifié de nombreux éléments dans l'étude de la mobilité et de l'activité humaine. Il est désormais possible de collecter d'importants volumes d'informations et de les traiter dans des délais répondant aux exigences du temps réel.

Les sources et les instruments de mesure sont omniprésents. Le *smartphone*, équipant 58 % de la population française de 12 ans et plus [19], peut jouer le rôle d'une sonde réseau active, capable de mesurer les performances réseau. Il peut aussi permettre au client de signaler sa position géographique déclarative ou, grâce au module GPS qu'il intègre, ses coordonnées précises. On retrouve ces mêmes capteurs GPS équipant des taxis tout autour du monde, nous donnant des informations sur la structure urbaine. Mais la démocratisation du GPS et l'apparition des *smartphones* ne sont pas les seules nouveautés.

Les réseaux de transports urbains, en informatisant et en automatisant leurs infrastructures, permettent de compter et de suivre plus facilement les voyageurs empruntant leur réseau. Les infrastructures de télécommunication mobile couvrant une très grande partie du territoire permettent de connaître les mouvements et l'activité d'une très grande partie de la population - ils connectent 92 % des Français de plus de 12 ans. À cela s'ajoute l'apparition d'architectures de stockage et de trai-

tements distribués à faible coût. Celles-ci permettent l'analyse rapide et efficace des grands volumes de données générés par ces nouvelles sources.

La nouveauté de ces jeux de données est leur disponibilité quasi immédiate. Là où le dénombrement de véhicules transitant sur un réseau routier se faisait manuellement en visionnant des enregistrements vidéo [55], où le comptage des voyageurs en Île-de-France se fait depuis les quais, une fois tous les 4 ans par ligne de trains de banlieue [121]. Les plateformes d'analyse connaissent aujourd'hui la position des objets étudiés avec une fréquence supérieure à une fois toutes les 10 secondes. Le matériau utilisé est donc à jour et valide. Désormais, l'enjeu des différentes recherches visant à estimer le trafic routier est d'accomplir cette tâche en temps réel.

Seront abordées dans ce chapitre les différentes sources et méthodes de mesure de la mobilité, actives et passives. Nous étudierons également les éléments d'architecture d'une plateforme d'acquisition et de traitement de flux réseau. Finalement nous aborderons l'analyse de la mobilité et les méthodes permettant de placer le résultat des mesures dans un référentiel géographique et d'en extraire des trajets.

2.2 Mesures classiques

La méthode de mesure classique de la mobilité humaine consiste à relever régulièrement la position géographique des individus étudiés. Cette position peut être déclarative ou obtenue par un appareil de mesure. Elle peut être signalée par l'intervention de l'individu ou obtenue automatiquement, sans son concours actif.
Nous aborderons ici ces différentes mesures, actives et passives.

2.2.1 Mesures actives de la mobilité

Une catégorie de mesure active se base sur la signalisation manuelle et volontaire par l'utilisateur de sa position géographique via son *smartphone*. Cette déclaration est faite en utilisant des applications développées pour la recherche, ou grâce à des applications de type *réseau social*. Cette deuxième catégorie est connue sous le terme de réseau social géolocalisé (Location-Based Social Network Service, LBSNS). À chaque fois que l'individu se rend dans un lieu, il le signale en lançant l'application idoine et en choisissant le lieu parmi ceux proposés. La valeur de cette information, par rapport à celles extraites de mesures passives, est qu'elle est labellisée. Et cette labellisation est hiérarchique, elle inventorie des catégories de lieu - restaurant, bar, bureaux - et les nomme. On peut ainsi en extraire les lieux de résidence et de travail et la densité des points d'intérêt dans une zone géographique. Il est également à no-

ter la dimension *sociale* de ces applications, l'utilisateur renseigne un profil. Profil qu'il peut lier à celui d'autres utilisateurs de l'application. Jiang et al. [75] étudient l'occupation spatio-temporelle de Chicago grâce à une application ad hoc. Ils caractérisent les déplacements par catégories de population et l'occupation de la ville. Dans la même catégorie Roth et al. [95] utilisent des données provenant du réseau social Foursquare et Liu et al. [82] celles d'un réseau social chinois. Nous parlerons de ces deux travaux plus loin. Ces mesures ont l'inconvénient de dépendre de l'implication de l'individu. Celle-ci peut dépendre de l'intérêt qu'il a pour l'expérience ou pour le réseau social, de l'influence de l'application sur l'autonomie de son téléphone ou encore de la mode. Les mesures passives ne souffrent pas de ces défauts.

2.2.2 Mesures passives de la mobilité

Les infrastructures fournissant le plus de données aux mesures passives sont les transports et les télécommunications. Au nombre de ces sources, nous trouvons les traces de taxis new-yorkais [27] dont les modules GPS mesurent à intervalles réguliers et de manière précise les trajets. Liu et al. [81] basent leur analyse sur ce type de données pour la ville de Shanghai. Cette analyse permet de faire émerger, grâce à un algorithme de détection de communauté, une division de la ville basée sur les trajets. A l'autre extrémité du spectre de densité spatio-temporelle, se trouvent les réseaux de partage de vélos, comme ceux par exemple de la région de San Francisco ou de Philadelphie [21, 62]. Les jeux de données qui en sont extraits ne contiennent que des matrices Origine-Destination (OD) associées aux dates d'emprunt et de dépôt. Zaltz Austwick et al. [144] étudient ces systèmes dans les villes de Londres, Washington, Minneapolis, Denver et Boston, et font ressortir, via un algorithme de détection de communauté, la structure de la ville et les points communiquant en fonction des heures de la journée et du jour de la semaine. De type OD également, nous trouvons les données issues des infrastructures de transports en commun, comme le métro [111] ou le bus [124].

2.3 Mesures réseau

Les mesures réseau nous intéressent à plus d'un titre. Différents travaux ont prouvé qu'il est possible d'obtenir, à partir des traces provenant de réseaux mobiles d'opérateur, des informations décrivant les mouvements de population. Mais l'analyse de cette source a aussi un intérêt industriel, il permet de détecter des anomalies dans l'infrastructure mobile et d'estimer la qualité d'expérience du client.
Les deux aspects de ces mesures seront étudiés, ceci tout en maintenant la séparation entre mesures actives et passives.

2.3.1 Mesures réseau actives

La première catégorie de ce type de mesures actives cherche à qualifier le réseau. Quelques initiatives font l'inventaire des cellules des opérateurs et de leurs positions géographiques depuis des applications installées sur des *smartphones* [91, 133, 37]. Toujours depuis le mobile client, d'autres projets se concentrent sur des mesures concernant la qualité du réseau. Parmi eux, MobiPerf [88], basé sur Mobilyzer [26] (une bibliothèque de développement permettant de construire rapidement des applications de mesure), permet à Rosen et al. [109] de mesurer un élément de la gestion de ressource radio du LTE. Ces outils nous permettraient de faire des mesures de terrain pour vérifier des hypothèses basées sur d'autres sources de données, ou de vérifier la cohérence du référentiel géographique de l'opérateur. Mais le développement de tels outils et le suivi de telles mesures sont chronophages. De plus, de telles applications ont un impact sur la consommation d'énergie du mobile, ce qui pourrait ainsi pousser l'utilisateur à cesser de s'en servir, voire à les désinstaller. De surcroît, de nombreuses restrictions des Software Developement Kits (SDK) officiels fournis par les fabricants de smartphones, limitent et parfois interdisent l'accès aux informations des couches basses du système d'exploitation de l'appareil. Ces limitations restreignent la plage des mesures possibles, mais aussi le choix de la plateforme sur laquelle effectuer lesdites mesures (le framework cité plus haut fonctionne uniquement sur Android [49]) et ainsi limitent la population des testeurs potentiels.

Nous trouvons également des mesures actives faites par le réseau lui-même. Ficek et al. [38] ont construit une plateforme permettant depuis le cœur de réseau, grâce à l'utilisation conjointe de messages de signalisation Signalling System n° 7 et (SS7) de SMS de classe 0 (invisibles pour l'utilisateur), de connaître en temps réel la position d'un équipement mobile. Bien que ce type de mesures risque de saturer le réseau (la fréquence d'émission de SMS par cellule est limitée, et les canaux de signalisation sont restreints), ce système permet de suivre des milliers d'utilisateurs avec une fréquence d'un message toutes les deux minutes, et ceci sans monopoliser les ressources. Les auteurs présentent également la possibilité de remplacer le SMS de classe 0 par d'autres mécanismes comme un *ping* pour les utilisateurs dont la connexion de données est active. Une autre solution proposée est d'interroger directement les SGSN en SS7 pour connaître la cellule d'attachement d'un terminal mobile. Cette plateforme, bien qu'efficace, ne permet que difficilement de restreindre les travaux sur la mobilité humaine à une zone géographique ou de constituer un échantillon de mobiles équirépartis sur le territoire pour une étude sur la France entière. La clef des requêtes étant le MSISDN, des travaux préliminaires ayant comme but de lier le mobile à un zone géographique seraient nécessaires et cette tâche n'est pas triviale.

2.3.2 Mesures réseau passives

Les journaux d'appels (Calling Detail Records, CDR) du réseau mobile sont des sources populaires dans les études portant sur la mobilité humaine. Ils répertorient chacun des appels en donnant l'appelant et l'appelé dans le cas d'un appel, ou la source et la destination dans le cas d'un SMS. Est également indiquée la position géographique de chacune des parties. Des jeux de données, une fois anonymisés, ont été publiés dans différents challenges à l'initiative d'opérateurs. On trouve le Data for Developpement (D4D) [99] à l'initiative d'Orange (concernant des données de pays d'Afrique de l'Ouest) et le BigData Challenge [125] de Telecom Italia, ce jeu comporte également des données issues du réseau de données mobiles. Ces journaux peuvent être utilisés sous l'angle de la mobilité, en observant les déplacements entre les communications, mais aussi d'un point de vue social, en étudiant les interactions. C'est sous cet angle que [129] traite ces données. Il utilise les matrices OD d'appel et en fait une analyse de détection de communauté : qui appelle qui en Irlande. Il en découle une caractérisation des appels en fonction de la distance et de l'heure de la journée, et un regroupement géographique des usagers en fonction de leurs appels.
Zhang et al. [145] ont utilisé pour la région de Shenzhen trois jeux de données qu'ils agrègent et comparent. Ing et al. se basent sur des CDR, les traces GPS des taxis, et les oblitérations dans le métro et le bus. Ils en concluent que les CDR manquent de précision. C'est pour cette raison que nous souhaitons explorer une autre source provenant du réseau de télécommunications, le GTP.
L'analyse de la signalisation du domaine de données n'est pas beaucoup représentée dans la littérature. Mais nous ne sommes pas les premiers à nous y intéresser. Metzger et al. dans [87] mesurent et caractérisent des flux GTP, allant du nombre des sessions établies à la charge des équipements. Aggarwal et al. [9], eux, estiment la qualité d'expérience sur le GTP. Xu et al. [139] estiment la précision de la géolocalisation à partir de traces de signalisation GTP et de statistiques de bascule de cellule à cellule.

La complexité de la collecte dépend du type de données à extraire. Dans le cas de CDR, l'opérateur a l'obligation de conserver ces informations pour, entre autres, des contraintes liées à la facturation. Ces journaux sont donc centralisés et stockés de manière structurée, et leur mise à disposition ne demande pas de développement. L'analyse d'autres évènements réseau peut être plus ardue. La manière la plus simple est celle exposée par Xu et al. [139], elle consiste à récupérer des statistiques générées par les équipements. Une déclinaison est l'analyse de journaux d'activités système des mêmes équipements. Mais ces deux solutions dépendent fortement du constructeur du matériel, du format et des types de statistiques générées et de la structure de ce qu'il produit. Le contenu des jeux de données utilisés varie,

METAWIN, la plateforme utilisée par Wolf et al. leur fournit pour leur article une suite de lignes contenant les champs RAT, le code TAC, l'heure de l'évènement et le type de message reçu. Ce qui dans ce cas est suffisant. Alors que Xu et al. [139] utiliseront également les champs compris dans User Location Information (ULI) et plus particulièrement le Location Area Code (LAC).

De tels volumes, répartis sur tout un pays nécessitent d'être centralisés. Le traitement sur de longues périodes et sur un espace géographique aussi vaste soulève quelques problèmes.

2.4 Capture réseau, centralisation, stockage et transformations des données

Mener des mesures passives sur le réseau mobile d'un opérateur entraîne un ensemble de contraintes sur la construction d'une structure globale de traitement de données. Celle-ci doit être capable de traiter d'importants volumes en entrée sur des sites installés à différents endroits du pays, de centraliser les résultats et de les analyser efficacement, pour rendre exploitables des données bruitées.
Nous considérons dans cette partie les éléments de cette chaîne.

2.4.1 Plateformes de capture, de stockage et d'analyse

Les points d'interconnexion d'un opérateur mobile entre son réseau voix et le réseau téléphonique commuté, et entre son réseau de données et Internet sont distribués sur tout le pays. Par conséquent, si l'on souhaite analyser le trafic sur tout le territoire et sur de longues périodes, la capture doit être distribuée et le trafic centralisé. La conception d'une architecture mêlant la capture de débits proches de 10Gb/s par interface, et le stockage et l'analyse de plusieurs centaines de gigaoctets de données par jour pose différentes questions. Ceci aussi bien au niveau du format d'échange entre les différentes entités, de ses performances et de son intégration, que du stockage et des outils d'analyse de données. Pour y répondre, nous avons étudié les différentes solutions traitant de ce sujet de collecte et de centralisation. Nous introduirons ce sujet en présentant les plateformes de stockage et d'analyse de gros volumes de données pour présenter plus clairement le reste des architectures dans leur ensemble. Les architectures présentées collectent aussi bien des données issues de mesures actives de mobilité que de mesures visant à étudier la qualité d'un réseau informatique.

Capture

La capture réseau ne dépend pas d'un équipementier ni d'un constructeur, les protocoles analysés sont identiques chez tous les opérateurs. Au nombre des sondes passives, Xioali et al. [136] ont développé des sondes de capture GTP. Ils le font en utilisant des processeurs réseau, déléguant le traitement réseau hors du système d'exploitation. Un autre type de sonde, développé par Wolf et al. [134], se base également sur des processeurs réseau pour analyser les flux. Dans cette étude ne sont analysés que les entêtes des couches réseau et de transport. Ce qu'apporte l'architecture présentée est son caractère distribué et la mise en avant de la séparation entre les traitements temps réel et ceux pouvant se faire hors-ligne.

L'intérêt de sortir une partie du traitement des flux réseau de l'espace utilisateur, comme présenté dans les deux travaux précédents, est de simplifier son traitement. Comme nous le voyons avec Windmill [86], le filtrage n'est pas une chose aisée et un moyen de le faciliter (et de soulager le noyau et l'espace utilisateur) est de le traiter au niveau de la carte réseau. Ce choix a un impact sur les performances globales.

La gestion des flux réseau par les systèmes d'exploitation est une tâche complexe et n'est pas adaptée à l'analyse de flux dont les débits avoisinent 10Gb/s. Les coûts générés par les appels système et d'allocation mémoire à l'arrivée d'un paquet sont trop importants. De plus, le caractère exhaustif de l'implémentation des protocoles n'est pas indispensable pour une sonde. On peut limiter l'analyse aux protocoles que l'on veut traiter et aux éléments qui nous intéressent. Luigi Rizzo dans [108] annonce que sur un lien de 10Gb/s un paquet est émis toutes les 67.2 nanosecondes, et que ce même paquet parcourt, au sein du système d'exploitation, le chemin entre le lien et l'application en 10 à 20 fois plus de temps. Il propose ainsi un système, appelé *netmap* permettant d'accéder directement aux buffers de la carte réseau et de ne copier qu'une fois le paquet dans un buffer circulaire auquel l'application a accès. Cette technique est également utilisée par DPDK [35] et PF_RING [96]. Les fournisseurs de cartes de capture tels Napatech [92] ou Emulex [36], font des traitements sur la carte, mais fournissent un accès en espace utilisateur. Dans tous ces cas, le paquet ne traverse pas le noyau et il est directement disponible depuis l'espace utilisateur.

Stockage et analyse

Dans le but de mener des analyses exploratoires sur les données que nous collectons, l'utilisation d'une base de données a été écartée. Celle-ci nécessite de struc-

turer et d'organiser nos informations d'une manière trop contraignante à ce stade de la construction de notre infrastructure. L'approche choisie est une approche *Big-Data*. Nous la définirons comme l'utilisation d'un stockage de fichiers distribué et d'un traitement lui aussi distribué desdits fichiers. Disco [32] et Hadoop [11] répondent à ces critères, ils possèdent tous les deux un système de fichiers distribué - respectivement Disco Distributed Filesystem (DDFS) et Hadoop Distributed Filesystem (HDFS) - et d'un système de distribution de tâches sur les nœuds d'une grappe de type MapReduce. Le processus Map extrait les données des fichiers, les traite, les trie et envoie le résultat au processus Reduce qui agrège les flux et y applique un autre traitement et potentiellement renvoie le résultat à un autre processus Map. De par sa maturité et sa plus large utilisation, nous nous concentrerons sur Hadoop. Dans l'écosystème d'Hadoop, de nombreux outils sont disponibles et utilisés dans les travaux présentés. HBase [15] est une base de données utilisant HDFS comme stockage, Hive [12] un outil permettant de lancer des travaux MapReduce depuis une interface SQL et Pig [17], lui aussi, fournit une interface haut niveau pour accéder aux données.

Il existe une alternative à Hadoop MapReduce. Spark [14] est un outil, qui comme Haddop MapReduce, lit les données dans HDFS. Et comme son alternative permet de lancer des travaux de type MapReduce. L'apport de Spark est la structure distribuée et résiliente (Resilient Distributed Datasets,RRD) [143], celle-ci est stockée en mémoire et distribuée sur les nœuds d'une grappe. Elle est le résultat d'une transformation (un map, un filter ou un join) sur des données stockées dans HDFS ou sur un autre RDD. La gestion de la résilience de ces datasets n'est pas gérée par le stockage des résultats intermédiaires d'une transformation, mais par la conservation des étapes de modifications faites aux données. Par conséquent, si un RDD est perdu il sera facilement reconstruit à partir des données d'origine. Ce stockage en mémoire permet d'atteindre des performances élevées pour les algorithmes itératifs. PageRank appliqué à une extraction de 54G de Wikipedia a des performances 7.4 fois supérieures [143] à celle d'Hadoop MapReduce.

Plateformes distribuées d'analyse de la mobilité

Des plateformes distribuées de mesures de la mobilité sont construites sur ce principe. Mobile Millennium Project [60] et CANDS [141] en sont deux exemples. Les deux se basent sur des données GPS indiquant la position d'automobiles sur le réseau routier. Le premier vise à en estimer la charge en utilisant Spark. Le deuxième calcule le plus court chemin en fonction de la charge du réseau. Ce calcul se fait grâce à un système de graphe distribué développé pour ces travaux.

Les mesures réseau distribuées peuvent elles aussi se baser sur le même schéma, des nœuds distants envoyant leurs données et une plateforme les centralisant et fournissant des outils d'analyse. Nous pouvons citer RIPE-NCC Atlas [105], où un ensemble de sondes connectées à Internet et disséminées autour du monde font des mesures (ping, traceroute, requêtes DNS). Ces mesures sont orchestrées par un système centralisé conçu par le RIPE-NCC. Les résultats des mesures sont stockés dans un cluster Apache, HBase [15] central qui est interrogé via Hive [12] (fournissant ainsi une interface SQL à l'utilisateur final). Quant à Measurement-Lab [85], une autre plateforme de mesures actives, celle-ci fournit une plateforme centralisée de traitement et d'analyse basée sur Google BigQuery [50] et Google Cloud Storage. Cette plateforme permet aux projets de recherche de stocker leurs données et de les analyser grâce à la l'infrastructure de Google, et ainsi de se focaliser sur la mesure et l'analyse plutôt que sur la construction des outils.

Format de stockage et d'échange de données

Finalement, la question du format de sortie des sondes est importante. Il a une influence sur de nombreux points, les performances (de lecture et d'écriture), la facilité des échanges entre les sondes et l'infrastructure d'analyse et l'occupation réseau et disque. Nous trouvons dans la littérature des outils et bibliothèques permettant de lire et d'extraire des données des fichiers PCAP - un des standards dans les formats de captures réseau. Le RIPE-NCC a produit un outil capable de lire des fichiers PCAP et de les traiter grâce à Hadoop MapReduce [106]. Packetloop propose une interface Apache Pig [100] dans le même but, mais ces fichiers sont trop volumineux et par conséquent trop lents à traiter. Un format utilisé dans le domaine de l'analyse réseau est Netflow, il est produit par des équipements réseau pour comptabiliser les flux les traversant. Différents projets permettent de les analyser grâce à Hadoop [78, 25, 146]. Or ce format est pensé pour décrire des flux de données, il ne nous permet pas, par exemple, de décrire les éléments reçus dans les paquets de signalisation.
Dans cette optique, nous avons cherché des formats nous permettant de lire et d'écrire dans les langages présents sur les différentes parties de la plateforme (C/C++, Java/Scala, Python). Nous cherchons un format sérialisé, afin de lire facilement les données dans le format dans lequel elles ont été écrites, efficace et de préférence binaire, ne prenant pas beaucoup d'espace et supportant la compression. Un des critères est aussi sa compatibilité avec les outils d'analyse disponibles. Json [76], BSON [24], Protocol Buffers [52], ou encore Avro [16] sont des formats sérialisés, les trois derniers sont binaires et seul Avro supporte la compression nativement. Avro, de surcroît dispose de bibliothèques disponibles dans les langages que nous utilisons et se lit nativement depuis Hadoop et Spark.

2.4.2 Filtrage des données de mobilité du réseau mobile

Les données de mobilité issues des infrastructures de télécommunications doivent être filtrées pour être exploitables. Des aberrations dans les transitions entre les cellules sont à écarter.

Iovan et al. reconstituent les trajets des individus à partir de CDR. Ils inventorient dans cet article [64] les différentes étapes visant à nettoyer ces données pour supprimer le bruit contenu dans ces trajets. Ils suppriment les doublons, les effets de *ping-pong* entre les cellules pour les clients situés en bordure de zones. Ce phénomène se manifeste par un aller-retour entre deux cellules dans un court laps de temps (10 secondes dans cet article). Ils filtrent les transitions entres cellules en se basant sur le temps entre la connexion à la cellule a et l'arrivée en cellule b et sur la distance les séparant, ils en déduisent une vitesse, qui en plus de l'angle entre les segments reliant les cellules du parcours, sert de paramètre à leur filtre. Horn et al. [57] utilisent des données hybrides, provenant du réseau de voix et de données, ils cherchent également à supprimer les transitions atypiques. Et comme dans [64], ils se basent sur la vitesse de transition pour construire leurs filtres *Recursive Naive Filter* et *Recursive Look-Ahead Filter* dont ils comparent les résultats avec ceux d'un filtre de Kalman. Il en résulte que les performances du *Recursive Naive Filter* sont proches de celles du filtre de Kalman, et ceci avec une complexité d'implémentation bien moindre. Finalement, Wu et al. [135] se concentrent sur les oscillations et leur suppression. Une oscillation est la connexion d'un mobile sur une période de a minute(s) sur un ensemble de b cellules contenant c cellules différentes. Ils présentent leur algorithme de détection et leurs critères de suppression, et comparent leur résultats à des mesures de la localisation réelle de l'individu pour lequel ces traces ont été nettoyées. Leur algorithme supprime bien les cellules les plus éloignées des positions réelles des individus et conserve les points qui en sont les plus proches.

2.5 Analyses de la mobilité et de l'activité humaine

Étudier l'activité humaine à partir des données collectées, observer les déplacements de population et leurs motifs sont les buts de nos travaux. Nous souhaitons aussi pouvoir reconstruire des trajets et estimer la charge des réseaux de transports. Nous nous penchons ici sur les analyses de la mobilité humaine et sur les procédés permettant de placer les observations sur les axes de transports.

2.5.1 Mobilité et activité humaine

Un volume considérable d'études sur la mobilité humaine a été dédié à l'analyse des trajectoires d'individus basées sur leurs traces. Des caractéristiques spatiales telles que le centre de leur masse, le rayon de giration et des caractéristiques statistiques ont révélé un nombre de propriétés d'échelle des trajectoires humaines : Gonzalez et al. dans [48], et Brockmann et al. [23] ont montré que la distribution de la longueur des sauts suit une loi de puissance tronquée. Il a été observé que la plupart des individus ne voyagent que sur une courte distance, et qu'il n'y en a que très peu qui voyagent sur plus de cent kilomètres [120]. D'autres études [117, 48] ont montré que les motifs de déplacement se résument à une simple distribution de probabilité spatiale, indiquant que, malgré la diversité de leur historique de déplacement, les êtres humains suivent un motif simple et reproductible. De plus, des analyses statistiques confirment que les mouvements des individus suivent un motif spatio-temporel [44, 43, 28] pouvant aider à la définition de modèles de mobilité. D'autres [124] ont montré que ce caractère prédictible implique également de croiser régulièrement les mêmes personnes sur son trajet.

Des études [111, 84] se sont concentrées sur la structure urbaine en mettant au point des méthodes permettant d'identifier des centres d'activité au sein de villes. La première en se basant sur les débits entrants des nœuds du réseau du transport urbain et de leur proximité spatiale. La seconde mesure la densité de population par zone géographique. Xu et al. dans [138], étudient la structure de la ville sous l'angle des trajets en taxi, et font ressortir les nœuds du réseau routier urbain.

Viennent ensuite les mesures faites sur les réseaux sociaux géolocalisés par Noulas et al. [95] et Liu et al. [82]. Alors que la première étude tend à prouver que les déplacements intra-urbains à Houston, San Francisco et Singapour ne suivent pas une loi de puissance, mais qu'ils sont fonction de la densité des points d'intérêt par zone géographique. Le second article lui, montre que les déplacements interurbains en Chine sont conditionnés par la masse des différents ensembles urbains, et que les interactions spatiales décroissent en fonction de la distance.

Enfin les travaux de Wang et al. [130] et Abadi et al. [7] quant à eux infèrent le trafic routier à partir des CDR et d'un jeu de données éparses provenant de capteurs placés au bord du réseau routier.

Le sujet des transports et de l'occupation des réseaux nous intéresse, malheureusement pour les données basées sur des informations éparses telles des matrices OD, la reconstitution est souvent basée sur la recherche du chemin le plus court entre l'origine et la destination [117, 142, 44, 43]. De plus, ces études ne tiennent

compte que d'un seul mode de transport et sont, d'après nous, incomplètes. Nous chercherons dans ce qui suit à pallier ce manque.

2.5.2 Reconstitution des trajets et placement des points sur les axes de transport

Afin d'estimer le plus finement possible la charge des réseaux de transport, il convient de localiser le plus précisément possible les individus et d'identifier le mode de déplacement qu'ils empruntent. Les sources de données permettant de localiser la population et les techniques pour le faire sont nombreuses.
En même temps que les études sur la mobilité, des applications telles que des systèmes de navigation, de supervision du trafic routier et des réseaux de transport en commun ont utilisé des données GPS afin de suivre les individus ou n'importe quel objet en mouvement [126, 46, 94, 58, 137, 60]. Une variété d'approches statistiques telles que l'algorithme d'espérance-maximisation (EM) [60], le filtre de Kalman [58, 137] et le Modèle de Markov Caché (MMC) [127, 126, 46, 94, 59] ont été utilisés pour projeter des données séquentielles bruitées de géolocalisation sur des réseaux de transport. La plupart de ces algorithmes ont utilisé des données GPS car elles apportent des données précises ayant une marge d'erreur d'environ 50 mètres. De plus, utilisant des données labellisées, des méthodes supervisées ont été utilisées et entraînées pour optimiser automatiquement les paramètres du modèle. Une fois les modèles entraînés, ils sont utilisés pour trouver le chemin le plus probable dans le réseau à partir de séquences de géolocalisation bruitées. Toutefois, la plupart de ces algorithmes de projection sont développés pour projeter des données bruitées sur des réseaux routiers sans considérer d'autres types de mobilité.

Récemment, grâce à la forte croissance du nombre de téléphones portables, les journaux d'appels (Call Data Records ou CDR) ont fourni des jeux de données valables pour les études de la mobilité. Les CDR, toutefois, ont des limitations importantes : premièrement ils sont rares dans le temps, car ils ne sont générés que lorsqu'un appel est émis ou lors d'un échange de message texte, et ils ont une précision spatiale plus faible que les données de géolocalisation GPS. Leur précision est à l'échelle des cellules du réseau mobile (avec une erreur moyenne allant de 175 mètres en milieu dense à 2 kilomètres dans un milieu moins dense). Toutefois, le fait que le taux d'équipement de la population soit très élevé [118] permet d'étudier des aspects importants de la mobilité humaine, tel qu'inférer le moyen de transport emprunté. Les informations du réseau de données étaient par exemple utilisées pour classifier différents types de transport pour des trajets de longue distance [118, 34]. Thiagaran et al. dans [126] ont utilisé des données de signal combinées à des capteurs pour développer un algorithme non supervisé de projection afin de surmonter les limitations des données GPS.

Les études des villes intelligentes ont été dans le passé limitées à l'analyse du réseau multimodal de transport sans considérer des données de mobilité à grande échelle. Le but principal des études de mobilité multimodale est d'améliorer la supervision des transports publics et des réduire les congestions de trafic [80, 93, 10].

3 Capture, stockage et analyse

La construction d'une plateforme capable de transformer les flux du réseau mobile en un jeu de données reflétant la mobilité de la population d'un pays sur l'ensemble de son territoire est décomposable en trois lots.
Le premier porte sur la création d'un ensemble assurant l'extraction, depuis les flux du cœur du réseau mobile, des informations nous permettant d'étudier la mobilité humaine. Puis vient la mise en place d'une infrastructure de stockage pour ces données, portant sur l'ensemble du territoire et couvrant plusieurs semaines de trafic. Enfin vient la mise à disposition d'outils permettant une analyse fine et efficace des données stockées.
Le contexte de la capture a été présenté dans l'introduction, en 1.3.1. Nous y avons présenté le GTP ainsi que les interfaces sur lesquelles le capturer. La problématique abordée ici est l'élaboration d'une méthode nous permettant d'extraire les informations nous intéressant, ceci depuis des débits élevés et sans perdre de données. Ceci demande l'utilisation de matériel approprié et un développement adapté. Le stockage quant à lui, doit pouvoir accepter des volumes de données importants et doit pouvoir suivre les mises à jour capacitaires du réseau. Finalement, cette infrastructure doit nous permettre de mener, rapidement et facilement, des travaux à partir des données collectées.

Nous présenterons le réseau de l'opérateur et les points de capture. Nous décrirons les sondes, aussi bien au niveau matériel que logiciel. Pour conclure, nous présenterons la manière dont nous stockons les données et les outils disponibles pour les analyser.

3.1 Architecture globale de la plateforme et des points de capture GTP

À titre de rappel, les équipements de terminaison des tunnels GTP sont concentrés dans quelques sites dans le pays - moins d'une dizaine au total pour l'opérateur chez qui notre plateforme a été déployée. Ces sites sont des Points Of Presence (POP). Il s'agit des points naturels où installer nos sondes et où capturer les flux des signalisations et de données.

La plateforme, composée des sondes et de la plateforme de stockage, est représentée, au sein du POP et de l'architecture du réseau mobile, en Figure 3.1.

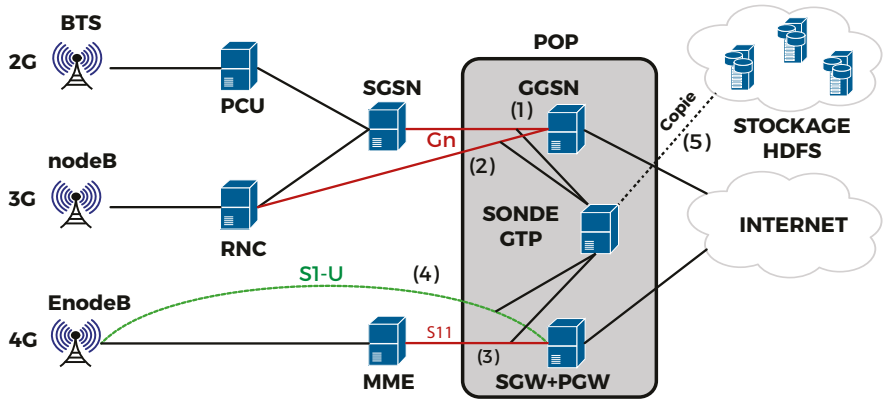


Figure 3.1 : Réseau cellulaire de données

En 2G et 3G, les sondes capturent le GTP-C et le GTP-U sur l'interface Gn, entre le SGSN et le GGSN, *(1)* sur la figure. Dans le cas du *Direct Tunnel*, le GTP-U transite directement entre le RNC et le GGSN, *(2)*.
La signalisation et le trafic utilisateur transitent sur deux interfaces différentes dans le cas du LTE. Le GTP-C emprunte l'interface S11 entre le MME et la SGW, représenté en *(3)*. Le GTP-U passe par la S1-U, directement de l'EnodeB à la SGW, en *(4)*.
Une fois les données extraites des paquets réseau, les sondes produisent des fichiers qui sont transmis, en *(5)*, à une plateforme centralisée dans le but d'être conservés et analysés.

La conception de sondes capables de supporter les débits des flux transitant dans les POP du réseau mobile, d'extraire, sans perte de paquets, les informations utiles à l'observation de la mobilité humaine est exposée ci-dessous.

3.2 Capture

Placées au cœur du réseau mobile et soumises à d'importants débits de données - avoisinants *20* Gb par seconde et par équipement, nos sondes doivent être capables d'analyser différents protocoles et d'en extraire des données signifiantes sans perdre de paquets. Elles doivent également produire des données facilement intégrables par des systèmes d'analyse.
La solution de capture doit être capable de copier rapidement les paquets vers le programme d'analyse en utilisant le minimum de ressources. L'analyse doit pouvoir

être faite à partir de différentes méthodes de capture et son architecture doit être modulaire. Elle doit nous permettre de facilement intégrer un nouveau protocole et ne pas avoir à modifier l'intégralité du code source de la lecture et du filtrage des paquets pour le faire. De même, le format de sortie doit pouvoir être modifié et intégré facilement.

3.2.1 Socle

Dans les principes, les traitements qu'une sonde passive applique aux flux qu'elle capture sont les mêmes que ceux que la pile réseau d'un système d'exploitation applique à ceux qui lui sont destinés. Elle analyse les paquets qui lui arrivent, vérifie leur validité et les filtre. Elle maintient une machine à états qui lui permet de suivre les sessions, de réassembler les paquets fragmentés et de réordonner certains paquets. Les différences entre ces deux systèmes, outre que la sonde ne fait que capturer les flux et qu'elle n'y répond pas, sont qu'une sonde peut avoir à garder en mémoire une donnée plus longtemps que ne le ferait un système d'exploitation, et qu'elle doit maintenir et traiter plus de sessions simultanément.
Le volume de sessions qu'elle doit supporter entraîne la parallélisation de ces traitements. Il peut être utile de distribuer la gestion des tables de sessions TCP entre différents microprocesseurs. La recherche dans une table pouvant être coûteuse, et ce coût croissant avec le nombre d'entrées, déléguer soulagera le processus principal occupé à lire les paquets. Ce traitement concurrent nécessite la mise en place de machines à états supplémentaires, celles-ci aiguillant les flux vers les processus de traitements parallèles.

Pour optimiser le traitement, il est nécessaire de filtrer les flux et de n'envoyer qu'un protocole par processus de traitement. Or le filtrage, si l'on considère un système fortement sollicité analysant plusieurs protocoles, a un inconvénient : il n'y a pas de mécanisme de mutualisation des filtres. Prenons le cas où nous désirons capturer en parallèle et depuis la même machine les flux SSH (TCP, port 22) et TELNET (TCP, port 23). Nous aurons dans le noyau, pour le premier un filtrage sur le protocole de transport, à la sortie duquel les flux TCP seront filtrés à la recherche des paquets dont le port source ou destination est égal à 22. La même opération sera répétée pour le deuxième flux, en basant cette fois le deuxième filtre sur le port 23. Nous sommes donc dans la situation où le noyau filtre deux fois les mêmes flux à la recherche du protocole TCP. Le Windmill Packet Filter (WPF) présenté dans [86] corrige ce problème et fournit une interface permettant un filtrage hiérarchique, permettant de réduire les règles présentées ci-dessus à trois et de ne filtrer qu'une seule fois le TCP.

En Figure 3.2 est représentée l'architecture d'une sonde classique. Les différents traitements sont répartis dans les couches *Matériel*, *Noyau* et *Couche utilisateur*. Le matériel fait l'interface entre le signal optique ou électrique et un signal exploitable par le système informatique. La partie noyau a la charge de transformer ces données en une structure exploitable et de la filtrer pour la transmettre à l'espace utilisateur. Toutes les tâches d'analyse et d'extraction des données sont réalisées dans l'espace utilisateur.

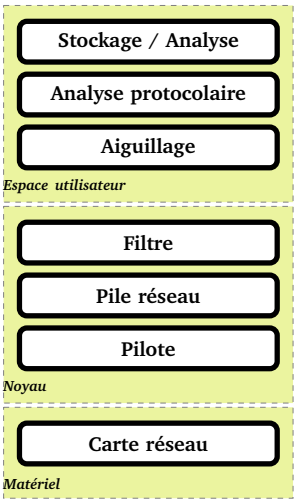


Figure 3.2 : Architecture d'une sonde sans modification du système d'exploitation.

L'architecture de Windmill respecte cette même répartition. Sa contribution se trouve au niveau du filtre présent dans le noyau : là où une sonde classique aura un filtre par flux choisi, WPF n'en a qu'un.

Les limitations du noyau du système d'exploitation

Cette architecture, soumise à des débits importants, souffre de quelques faiblesses. Premièrement au niveau du noyau. À chaque paquet reçu par le système, une interruption est émise par le pilote pour notifier le noyau d'une nouvelle donnée à récupérer. Ce mécanisme peut surcharger le microprocesseur. En cas de fort trafic, celui-ci sera occupé à traiter ces messages. Ce problème a été réglé en remplaçant ce mécanisme par celui d'une scrutation (*polling*) régulière de la carte réseau à la recherche de nouveaux messages. Cette technique peut être activée de manière permanente sous FreeBSD (*Device Polling* ou *Polling*) ou uniquement lors de pics de trafic sous Linux (*New API* ou *NAPI*).
Chaque paquet reçu donne lieu à au moins une allocation mémoire dans le noyau. La première structure allouée stocke des informations décrivant le paquet, telles que sa taille, un drapeau et son type. Cette structure contient également un pointeur vers la zone mémoire où se trouve le contenu du paquet réseau. Sous FreeBSD,

la structure peut stocker la donnée elle-même, si celle-ci est assez petite. Ces structures sont chaînées, elles pointent vers le prochain enregistrement. Cette structure se nomme MBUF sous FreeBSD ou SK_BUFF sous Linux.
L'allocation de ces structures de tailles fixes est faite dans des zones mémoire contiguës et réservées, ce qui est efficace. Mais dans le noyau Linux et FreeBSD, des mécanismes de verrous visant à la protection de l'accès aux ressources sont en place. Ils concernent aussi bien l'allocation et la destruction de mémoire que l'accès aux *sockets*. Or, comme le montrent les mesures réalisées par Rivera et al. dans [107], ces verrous ne sont pas efficaces dans un environnement multicœur. Et en raison de la complexité du noyau et de son code, ce problème est difficile à identifier et à corriger.

Optimisation en fonction de la taille des paquets

Nous avons émis l'idée d'optimiser les allocations mémoire dans le cas du traitement des flux de signalisation GTP sous FreeBSD. En observant la distribution de la taille des paquets de signalisation, nous avons remarqué que si nous étions capables de filtrer en entrée du système les flux pour ne conserver que ces éléments, nous pourrions économiser un grand nombre de ressources d'allocations.
Une structure MBUF peut stocker 168 octets sans avoir à faire appel ni à la création d'un autre MBUF, ni à celle d'une structure externe. En analysant une trace contenant $77\,109\,160$ messages de signalisation, nous avons observé que la taille de 97 % des paquets GTP-C en GTP est inférieure ou égale à 168 octets, de même que pour 98 % déléments en GTPv2. Ces données sont donc stockables dans la structure MBUF elle-même, et ceci nous permettrait d'économiser un nombre conséquent d'allocations mémoire. Or le filtrage des paquets, effectué par *Berkeley Packet Filter* (BPF) dans ce cas, est appliqué aux MBUFs. L'allocation se ferait donc aussi pour les paquets éliminés. Et la distribution des paquets GTP-U n'est pas à notre avantage, car 51 % des paquets ayant une taille supérieure à 168 octets (sur un jeu de données comprenant $16\,120\,618$ paquets). Réaliser cette optimisation nous forcerait à implémenter un filtre dans une couche basse du noyau, ce qui n'est pas trivial. De ce fait, les bénéfices de cette modification ne seraient probablement pas à la hauteur des efforts à déployer.

Une solution autonome, le processeur réseau

Un contournement possible consiste à utiliser des processeurs réseau, comme Xioali et al. [136]. Dans ce cas, aucun traitement n'est réalisé par le système d'exploitation.

Les processeurs réseau (Network Processor, NP) sont conçus pour un traitement parallèle des flux réseau. Ils disposent d'éléments de traitements indépendants, capables de traiter les flux en parallèle, d'éléments de calculs de hash d'opération binaire intégrés. On trouve ces processeurs réseau sous la forme d'une carte PCI disposant de ports réseau. Dans certains cas le bus PCI auquel il est connecté ne sert que d'alimentation électrique, et la communication avec l'extérieur se fait via un flux réseau. Leur utilisation règle les problèmes d'allocation mémoire, et de multiprocessing, car elle est indépendante d'un système d'exploitation classique. Mais le network processor ne se base pas sur une architecture classique. Le NP IXP2400 - utilisé pour l'article précédemment cité, se base sur une architecture non classique. Le développement fait appel à un environnement et à des bibliothèques spécifiques. Ce qui demande un investissement trop important pour une solution très dépendante du constructeur de la carte choisie. Le changement par le constructeur du microprocesseur utilisé pourrait demander la réécriture complète du code d'analyse réseau.

Le traitement des paquets directement en espace utilisateur

Les projets Netmap [108], DPDK [35] et PF_RING [96] apportent eux aussi une solution aux limites des systèmes d'exploitation. Les cartes modernes, telles que les cartes Intel basées sur un circuit 82598EB ou celles de la famille XL710, embarquent des queues (*rings*) d'émissions et de réception de paquets. Ces multiples *rings* associés aux cœurs des processeurs permettent un traitement parallèle des flux. Les projets cités plus haut tirent avantage de ces composants matériel en mettant à disposition de l'espace utilisateur les paquets qui y sont copiés. Et cela sans traverser la couche réseau du système d'exploitation.
Afin d'illustrer ce concept, nous décrirons Netmap. Celui-ci copie directement les données contenues dans les *rings* de la carte vers un *buffer* circulaire situé dans une zone mémoire partagée entre le noyau et l'espace utilisateur. Ce mécanisme a plusieurs avantages. Il réduit le nombre d'appels système en retournant plusieurs paquets à chaque itération. Il économise les copies de paquet et les allocations mémoire, la zone mémoire dans laquelle le paquet est copié est allouée statiquement au démarrage. Le système est efficace : il génère 14.88 Mpps pour un flux UDP, soit 40 fois plus de paquets qu'en employant une socket classique, et 10 fois plus qu'en faisant appel à l'outil de génération de paquets présent dans le noyau Linux [108].
Différents projets autour de la commutation de paquet et du routage utilisent cette technique pour améliorer leurs performances ; un routeur dans l'espace utilisateur [41], un switch utilisant *netmap* [56] et même une base de données profitant de *DPDK* pour accélérer ses performances globales [113].
En transmettant l'ensemble des paquets dans l'espace utilisateur, ces techniques y

déplacent aussi des tâches. Par conséquent, le filtrage des paquets, qui était réalisé dans le noyau est à la charge du développeur.

Les cartes de capture et leurs APIs

À la frontière entre les processeurs réseau et les projets de la catégorie de Netmap, nous trouvons les cartes de capture spécialisées, par exemple celles fabriquées par Napatech [92] et Emulex [36]. Du point de vue de l'utilisateur, le fonctionnement est le même que pour Netmap, les paquets sont disponibles directement dans l'espace utilisateur. Mais ces cartes effectuent plus de traitement que leurs équivalents classiques. Si l'on prend le cas des cartes Napatech, celles-ci embarquent de la mémoire vive et un FPGA reprogrammable. La mémoire vive permet de conserver des paquets en queue en cas de pointe de trafic ou de ralentissement du traitement sur le système. Le FPGA, permet de soulager le système d'exploitation en accomplissant, au niveau matériel, des tâches exigeantes pour le noyau. Il permet de filtrer les paquets, mais il assure également l'équilibrage de charge entre les flux en entrée et la copie en espace utilisateur. Les bibliothèques fournies par le fabricant permettent d'accéder depuis un programme écrit en C, tournant en espace utilisateur, à des descripteurs de fichiers sur lesquels il peut lire les paquets capturés. Ces descripteurs de fichiers peuvent être liés à une interface physique : sur *fd0* nous pouvons lire le trafic capturé par l'interface 0. Mais ces fichiers peuvent aussi être le résultat d'un filtre. Par exemple, *fd0* reçoit tout le trafic de signalisation alors que vers *fd1* on ne copie que le trafic utilisateur. Encore plus intéressant, le trafic peut être réparti sur plusieurs descripteurs de fichiers. Il est possible de filtrer le trafic et d'indiquer quelle fonction d'équilibrage y appliquer et vers quel nombre de descripteurs transmettre le résultat. Ceci offre la possibilité de filtrer le trafic du VLAN 51 et l'envoyer vers 6 flux en fonction d'une clef de session basée sur les éléments suivants : les adresses IP source et destination, le protocole de niveau 4 et les ports source et destination. Cela permet de lancer six processus, lisant chacun un descripteur de fichier sur lequel n'arrive qu'une partie du trafic, mais tous les paquets d'une même session.

Ce dernier point règle un élément important auquel nous sommes confrontés dans l'espace utilisateur. L'équilibrage de la charge de travail entre différents processus nécessite de répartir équitablement les flux en entrée, et cela, comme nous venons de l'aborder, en envoyant tous les paquets d'une même session vers le même processus de traitement. Dans ce but, une des solutions est l'utilisation d'une table de hachage dont la clef est construite à partir de caractéristiques du paquet, et dont la valeur identifie le processus de traitement. Or cette liste en grandissant devient exigeante en parcours et en mémoire, et par conséquent sa gestion est problématique.

Pour conclure, la solution consistant à utiliser des cartes de capture spécialisées semble la meilleure. Elle permet de passer outre la pile réseau du système d'exploitation et des problèmes de gestion des architectures multicœurs. Elle permet de surcroît de réaliser, au niveau matériel, des opérations coûteuses telles que le filtrage et la répartition de charge. Cette solution apporte liberté et souplesse au développement. Le traitement étant fait dans l'espace utilisateur, contrairement aux processeurs réseau, le développement peut se baser sur de nombreuses bibliothèques, le débogage est aisé et l'interconnexion avec d'autres systèmes est élémentaire.

Nous avons utilisé pour nos sondes des cartes Napatech nt20e2, et assuré l'analyse protocolaire en C depuis l'espace utilisateur, sous FreeBSD. Nos serveurs sont des HP Proliant DL 360 possédant 48 gigaoctets de mémoire vive et 2 microprocesseurs Intel X5690 de 6 cœurs à 3.46 GHz.

3.2.2 Dissection

L'architecture logicielle du système de capture développé dans le cadre de cette thèse se divise en trois sous-ensembles. L'acquisition, l'analyse protocolaire et le traitement de la donnée structurée. Cette répartition a été pensée dans la perspective de faciliter les développements, la maintenance et les évolutions du logiciel. De ce fait, la communication entre les différentes strates est réalisée grâce à l'appel de fonctions génériques masquant la complexité des couches sous-jacentes. Mais cette séparation a aussi été faite en fonction du caractère temps-réel des différentes étapes du traitement. Dans les étapes où les paquets doivent être traités de manière séquentielle, telle l'acquisition, le moindre retard entraînera la perte de paquets pour tout le système. Alors que pour les étapes parallélisables ou en fin de traitement, dont aucun autre traitement ne dépend, la contrainte concernant le temps d'exécution est bien plus faible.
Nous retrouvons en Figure 3.3 ces différentes divisions pour l'utilisation d'une carte de capture Napatech. Y sont représentés, la couche d'adaptation permettant la lecture en *(1)*, en *(2)* la dissection protocolaire écrivant dans le *bufffer* circulaire en *(3)* le résultat de son analyse. En *(4)* et *(5)*, nous trouvons des *threads* exploitant ces données.

Acquisition

Cet étage joue le rôle d'interface générique entre le flux réseau et le reste du programme. Il a été conçu pour fournir des flux filtrés, dans un format unique et au travers d'une interface générique, aux fonctions d'analyse protocolaire. Cette démarche rend la dissection et la manipulation des paquets indépendants de la carte

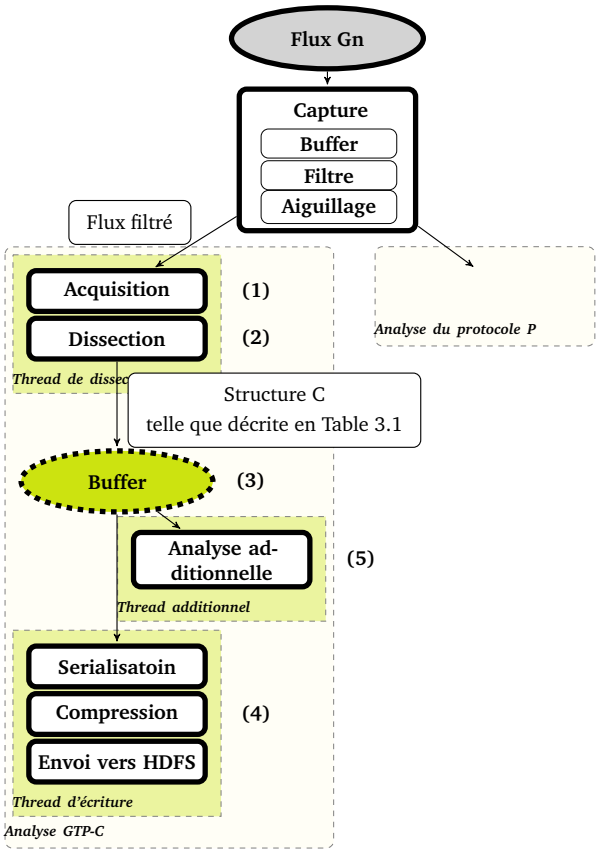


Figure 3.3 : Architecture logicielle

et de la méthode de capture.
Pour autoriser les développements hors ligne ou pour jouer des tests unitaires, cette couche permet de lire des flux *pcap* à partir de fichiers ou d'une carte réseau. Elle gère également les interfaces Netmap et l'API Napatech.

Analyse protocolaire

L'analyse protocolaire transforme le paquet IP en une structure C exploitable pour des traitements tels que l'écriture pour l'échange avec la plateforme d'analyse, ou pour le maintien d'une table de sessions des terminaux connectés.
Dans le cas du GTP-C, sont extraites du paquet de signalisation les informations présentes en Table 3.1.

Ces structures sont stockées dans un *buffer* circulaire afin d'être lues et traitées par d'autres *threads*.

Champ	Description
Date	Horodatage du paquet en μsec
Adresse IP du GGSN/SGW	Terminaisons du tunnel GTP
Adresse IP du SGSN/EnodeB	
ID utilisateur anonyme	ID aléatoire
Tunnel ID	Identifiant du tunnel transportant le trafic de l'utilisateur
Adresse IP du terminal	Adresse IPv4 ou IPv6 du mobile
Type de message	Create / Update / Delete PDP Context requête ou réponse
Techno d'accès radio	Bearer (GPRS, EDGE, EUTRAN ...)
Cause	Cause du refus
TAC	Modèle et fabricant de l'équipement
Numéro de séquence	Numéro de séquence GTP-C
QoS	Débit maximum d'envoi et de réception de données Débit garanti en envoi et réception Latence Priorité du trafic

Table 3.1 : Résumé GTP-C

Traitements supplémentaires et mises en forme

Ici la structure décrite en table 3.1 est mise en forme pour être écrite dans des fichiers ou envoyée à un processus capable de lire des flux de données. Elle peut aussi être exploitée par une machine à états suivant les sessions établies pour un traitement en temps réel.
Les *threads* réalisant ces traitements ont leur propre queue et ne font que dépiler les messages du *buffer* circulaire. Ceci afin de ralentir au minimum le *thread* d'écriture.

3.3 Stockage et analyse

Les choix faits pour la construction de cet élément de notre plateforme sont conditionnés *i*) par la nature des données que nous collectons, *ii*) par ce que voulons en extraire, *iii*) par les volumes que nous souhaitons stocker et *iv*) par les temps de traitement que nous supportons. Ces paramètres ont une forte influence sur l'ensemble du système.
La caractéristique principale de cette plateforme, aussi bien au niveau des sondes que de l'infrastructure d'analyse, est son caractère expérimental et exploratoire. Comme nous l'avons vu plus haut, la conception des sondes est modulaire, elle

permet d'ajouter facilement un nouveau protocole dans son processus d'analyse, ou un nouveau traitement aux protocoles déjà supportés. La structure d'analyse doit répondre à la même contrainte, supporter les changements sans demander en échange de trop grands efforts d'intégration. L'importance d'un champ protocolaire a pu être sous-estimée lors de la conception du système, et la modification de la structure de données émise par la sonde, provoquée par l'ajout de cette information, ne doit pas avoir d'impacts importants sur les outils d'analyse. Le stockage ne doit pas subir de modifications conséquentes et l'analyse doit pouvoir être faite de la même manière sur les données ayant été générées avant et après l'altération.

Un autre point est que nous ne pouvons pas présager précisément du traitement que nous allons faire des données collectées. Allons-nous uniquement nous baser sur l'individu pour reconstituer ses trajets, ou traiterons-nous uniquement les données en nous basant sur les données géospatiales ? Peut-être allons-nous utiliser toutes ces clefs, l'individu, le temps et l'identifiant géographique. Dans la mesure où nous pouvons utiliser n'importe quel champ comme clef de requête, il est impossible de hiérarchiser nos données pour en améliorer le traitement.

De surcroît les volumes que nous produisons et stockons sont importants (plusieurs centaines de gigaoctets par jour), et un traitement sur ce type de matériau demande des techniques particulières. Ces données peuvent être traitées par lots de fichiers ou grâce à une base de données répartissant les informations par clef d'identification. Ces deux méthodes n'ont ni les mêmes contraintes ni les mêmes performances. La première ne nécessite pas d'autres contraintes qu'avoir un système de stockage de fichiers réparti et un outil capable de les lire, les traiter et d'en centraliser le résultat. Par contre, une recherche dans ce cas impliquera la lecture de l'intégralité des fichiers considérés. De ce fait, le temps de traitement dépend du volume et du nombre de ces fichiers. La deuxième technique, se basant sur des clefs, connaît la répartition des données, sait où elles se trouvent. Elle est par conséquent très rapide. La contrepartie est qu'elle demande d'identifier une clef sur laquelle réaliser les requêtes, et cette clef sera difficilement modifiable, ce qui limite son caractère exploratoire. Nous sommes donc en présence d'une solution apportant souplesse et exploration, mais ne permettant pas d'obtenir des résultats en temps réel, et d'une autre, rapide, mais rigide dans la gestion et le stockage des informations.

Un autre critère entre en jeu dans le choix technique de la mise en œuvre de cette plateforme. Il s'agit du format de la donnée insérée. Là où d'autres plateformes se basent sur des données produites par des équipements sur lesquels ils n'ont pas prise, et donc formatées selon les besoins du constructeur et non ceux du chercheur, nous produisons nous-mêmes les données. Le choix du format des données produites a des impacts à plusieurs niveaux. Le format influence le temps entre la production et la mise à disposition, la consommation de ressources de stockage et de ressources réseau - pour le transport de ces informations. Le format est important, car il influence les performances de lectures et donc de traitement. Il doit aussi être choisi avec soin pour ne pas compliquer son exploitation, en demandant des

développements spécifiques pour permettre sa lecture.
Enfin, les outils d'analyse doivent permettre au chercheur d'analyser facilement les données produites, de mettre à sa disposition des bibliothèques et des outils évolués.

3.3.1 Limitations des bases de données

Alors que l'utilisation des bases de données semble tout indiquée pour accueillir les données produites et en permettre l'exploitation, nous leur avons préféré l'approche *BigData*.
Les systèmes de bases de données se présentent comme des ensembles contenant un système de stockage et une interface permettant à l'utilisateur d'effectuer des opérations sur celui-ci. Ces opérations sont, entre autres, les insertions, les mises à jour et les suppressions. Mais aussi des requêtes, des transformations (concaténations, extraction de sous-chaînes de caractères) et des opérations mathématiques (maxima, sommes, comptages).
Le sujet des bases de données a près de 50 ans. La structure du stockage et le formatage des données sont des domaines où les performances sont bonnes. Dans leur majorité, ces systèmes fournissent à l'utilisateur le moyen d'indexer ses informations dans le but d'y accéder rapidement. Les Systèmes de Gestion de Base de Données (SGBD) tiennent à jour, dans une structure dédiée, la correspondance entre la valeur d'un champ et l'endroit (ou les endroits) où il se trouve dans la base de données, ou plus exactement dans les fichiers qui la composent. Ces index peuvent aussi servir de clefs pour la répartition des données dans le cas d'un partitionnement horizontal (*sharding*). Mais ils ne sont efficaces qu'à certaines conditions, tous les types de champ ne peuvent pas servir d'index (les champs *BLOB* et *TEXT* sous *MySQL* par exemple), et les champs indexés se doivent d'être courts.
L'indexation n'est pas un problème (bien au contraire). L'obstacle à l'utilisation de ces plateformes dans notre cas est la rigidité des déclarations des structures de stockage et les conséquences qu'entraînent les erreurs de l'utilisateur dans leurs déclarations. Dans le cas des bases de données relationnelles, où les données sont stockées sous forme de tables, celles-ci sont fortement structurées. Chaque colonne est typée et sa longueur est limitée. Ainsi, si une colonne est déclarée sous MySQL comme un champ *Byte* et que la donnée qu'on souhaite y écrire a récemment changé (elle n'est plus comprise entre 0 et 255) il faudra modifier la structure de la table et le type de ce champ pour tous les enregistrements qu'elle contient. La déclaration d'un index n'est pas sans impact non plus. Elle nécessitera de construire ou de reconstruire la table de correspondance. Ces modifications ne sont pas problématiques pour les petites tables, mais dès que celles-ci grandissent le temps de traitement augmente. De plus les tables sont verrouillées et la charge du serveur peut être élevée pendant ces travaux.

Nous enregistrons aux alentours de 1 000 000 000 de messages par jour. Et nous ne pouvons pas nous permettre de réaliser des opérations sur des tables ne stockant ne serait-ce qu'une journée de ces données. Nous ne nous interdisons pas l'utilisation d'une SGBD dans le futur, mais nous pensons qu'elle n'est pas adaptée à cette phase exploratoire.

3.3.2 Stockage

Parmi les systèmes de fichiers distribués permettant un traitement distribué lui aussi (par exemple : lustre [114], ceph [132], swift [97], HDFS [18], GlusterFs [103]), nous avons décidé d'utiliser Hadoop File System (ou HDFS). Il fait partie d'Hadoop, il est donc intégré aux outils sans avoir besoin d'installer de logiciels supplémentaires. Il est *rack aware*, donc capable de distribuer les données en fonction de l'emplacement physique des serveurs au sein d'un datacenter. Les données sont répliquées sur plusieurs serveurs et le système fournit différentes interfaces pour accéder aux fichiers : API (java, C, python), outil shell et une interface HTTP.

3.3.3 Structure des données

La production et la consommation des données sont réalisées par les deux éléments d'une plateforme dont nous sommes responsables, ce qui nous laisse toute latitude pour choisir un format d'échange répondant à nos contraintes.
Le premier critère de ce choix est de disposer d'un format permettant aisément au système lisant la donnée, de la représenter telle qu'elle a été écrite. Les systèmes de sérialisation de données répondent à ce critère en nommant et en typant les valeurs qu'ils stockent, ils en facilitent ainsi la lecture et l'interprétation. Parmi les différents formats nous pouvons citer l' Extensible Markup Language [128] (XML), le JavaScript Object Notation [76], ou le *YAML Aint Markup Language* [140] (YAML).
Une contrainte inhérente à l'utilisation d'Hadoop, et de MapReduce en particulier, entre ici en compte. Afin de paralléliser le traitement des fichiers, HDFS les découpe en blocs. Chaque processus de traitement lira les blocs résultants ligne par ligne. Or certains types de fichiers ne contiennent qu'un document, se décomposant en sous-éléments, mais analysable uniquement s'il est lu dans son intégralité. C'est le cas du XML. Ces fichiers sont par conséquent insécables et supportent difficilement un traitement en parallèle. Le JSON a une logique différente, il permet d'avoir un enregistrement par ligne, facilitant ainsi le traitement concurrent des données.
Il existe quelques formats binaires de sérialisation. Là où le format classique stocke la donnée numérique dans sa représentation texte, le format binaire stocke les entiers tels quels (tout en en indiquant l'endianness pour l'interopérabilité). Nous prenons ici l'exemple du Binary JSON [24] (BSON) dans le but de le comparer

au JSON. Nous souhaitons stocker les valeurs de $valeur_a$ et celle de $valeur_b$, nous avons dans la représentation JSON :

```
{valeur_a': 1134343184553836202', valeur_b': 1000800281415265069}
```

Une fois sauvées dans un fichier, nous avons :

```
00000000  7b 22 76 61 6c 65 75 72  5f 62 22 3a 20 31 30 30  |{"valeur_b": 100|
00000010  30 38 30 30 32 38 31 34  31 35 32 36 35 30 36 39  |0800281415265069|
00000020  2c 20 22 76 61 6c 65 75  72 5f 61 22 3a 20 31 31  |, "valeur_a": 11|
00000030  33 34 33 34 33 31 38 34  35 35 33 38 33 36 32 30  |3434318455383620|
00000040  32 7d                                             |2}|
00000042
```

Alors que les mêmes données en BSON ont pour représentation :

```
00000000  29 00 00 00 12 76 61 6c  65 75 72 5f 62 00 2d 2b  |)....valeur_b.-+|
00000010  de b0 8d 8e e3 0d 12 76  61 6c 65 75 72 5f 61 00  |.......valeur_a.|
00000020  aa 7e b7 c8 1d ff bd 0f  00                       |.~.......|
00000029
```

Les représentations en base 16, sont :

— 0xDE38E8DB0DE2B2D pour 1 000 800 281 415 265 069 (valeur_b)

— 0xFBDFF1DC8B7B7EAA pour 1 134 343 184 553 836 202 (valeur_a)

Nous retrouvons ces deux valeurs en rouge et en *little endian* dans l'enregistrement binaire.

On observe une différence assez évidente en terme de taille de données, on voit dans la partie gauche que le *JSON* stocke la donnée en texte alors que pour le *BSON* seul le nom des champs apparaît. L'intérêt supplémentaire du BSON est la vitesse de désérialisation, les valeurs binaires sont le plus proches possible des types de la machine.
Il est à noter que l'avantage en termes d'espace que fournit le *BSON* par rapport au *JSON* est surtout valable pour les ensembles où les valeurs numériques sont nombreuses. Nous avons fait des mesures à partir d'un enregistrement produit par nos sondes, contenant 9 556 749 messages GTP-C. Chaque enregistrement contient 19 champs parmi lesquels 10 sont numériques. Le fichier formaté en *JSON* a une taille de 3.9 gigaoctets contre 3.5 gigaoctets pour le *BSON*, soit un gain de 10 %.

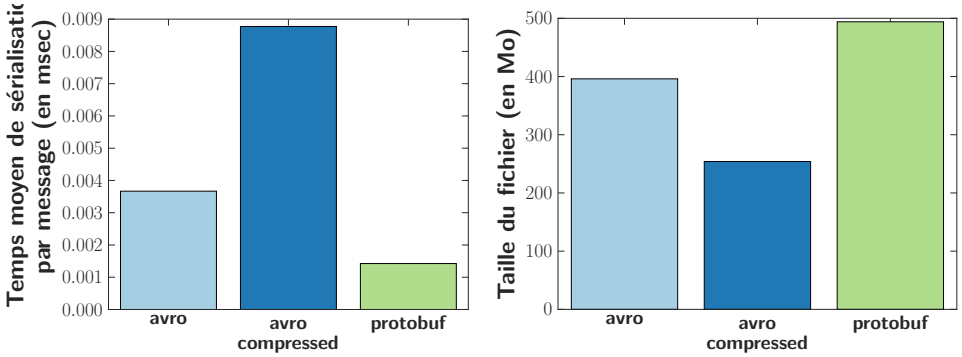
Ces résultats peuvent être améliorés en définissant le nom des champs en entête des fichiers ou dans un fichier externe, pour ne pas avoir à les répéter à chaque enregistrement. Une autre revient à compresser le fichier, ce qui implique d'utiliser en lecture un système capable de le décompresser à la volée. Parmi les formats supportant ces optimisations, nous citerons Protocol Buffer (Protobuf) [52] et Avro [16] (en Avro les données précédentes tiennent dans 1.2 gigaoctets). Il s'agit des deux formats avec lesquels nous avons travaillé et dont nous avons mesuré les performances.

Protobuf est un système de sérialisation binaire développé par Google. Un outil, appelé Protoc, prend en paramètre un schéma représentant la structure des informations à traiter et génère un ensemble de fonctions permettant au développeur de lire et d'écrire des données dans le format en question. Les champs peuvent être ajoutés ou enlevés du schéma et peuvent être définis comme optionnels. De plus, un framework nommé ElephantBird [131], développé par Twitter, permet de lire et d'écrire des fichiers Protobuf compressés. Comme nous l'avons vu plus haut, HDFS découpe les fichiers en blocs de 128 méga-octets (la valeur par défaut aujourd'hui), et ces blocs sont analysés par différentes tâches concurrentes et accélère ainsi le temps de traitement des fichiers. Par conséquent, pour que le traitement en parallèle soit possible, le fichier doit être sécable, et il est préférable que le format de compression soit supporté nativement par Hadoop. C'est pourquoi le choix de la bibliothèque de compression est important. Certains formats de compression sont composés de blocs de tailles variables bornés par des valeurs spéciales intercalées dans la donnée (par exemple chaque bloc gzip se termine par le code 256, et le suivant commence après ce code). Les fichiers compressés dans ces formats sont par conséquent indivisibles. C'est le cas pour le *gzip* [40]. *LZO* [45] possède lui aussi une taille de bloc variable, mais il peut construire un fichier d'index, indiquant la position de chacun de ces blocs à l'intérieur du fichier compressé. D'autres formats se basent sur des blocs à tailles fixes, c'est le cas pour bzip2 [116]. ElephantBird utilise *LZO*.

Avro est développé par l'Apache Foundation et suit le même principe que Protobuf ; un schéma décrivant les données et leurs types et un outil générant des fonctions dans un langage choisi. Le lecteur n'a pas besoin de posséder le schéma pour lire le fichier, l'API Avro écrit celui-ci dans le header du fichier. De plus Avro permet de compresser les données à la volée, là où ElephantBird compresse le fichier une fois que celui-ci est fermé. Les algorithmes de compression disponibles sont Deflate [30] et Snappy [51]. Avro et ses algorithmes de compression sont nativement supportés par Hadoop et Spark.

Nous avons comparé les performances de ces deux frameworks sur 15 minutes de trafic réel anonymisé, composé de 11 131 639 messages GTP-C. Travailler sur ces

données nous permet d'avoir un jeu de données aléatoires et conforme aux flux de production. Nous avons utilisé Avro version 1.7.5 et Protobuf 2.6.0. En Fig. 3.4a, nous comparons le temps moyen pris pour générer un message pour chacun des formats, Fig. 3.4b montre la taille finale des fichiers résultats.



(a) Temps moyen de sérialisation par message **(b)** Taille des fichiers résultants

Figure 3.4 : Comparatif des performances de différents systèmes de sérialisation (réalisé sur 11 131 639 messages GTP-C anonymisés)

Comme nous pouvons le voir en Figure 3.4a, les performances de sérialisation de Protobuf sont bien au-delà de celles d'Avro. Le rapport entre les temps de sérialisation est d'environ 3 avec Avro et de 8 avec la version compressée (basée sur *deflate* [30], disponible dans l'API).
En revanche, Avro est plus intéressant en termes d'occupation disque pour notre test. Il occupe 20 % de moins d'espace disque que Prototbuf dans sa version classique et 50 % en activant la compression. Il faut ajouter que pour que la donnée sérialisée par Protobuf soit exploitable (par ElephantBird), elle doit être compressée dans le format *LZO*. Cette étape de compression, qui n'est pas intégrée dans le processus de sérialisation, complexifie la génération de ces fichiers, et accroît le temps total passé à leur génération.

La complexité d'utilisation et de mise en place de ces formats n'est pas représentée dans ces courbes. Nos sondes sont développées en C, et de ce point de vue Avro a notre préférence, il fournit une API dans ce langage, alors que Protobuf ne fournit une API native qu'en C++. Ce qui ne le rend pas impossible à utiliser, mais rompt l'unité du code et complique sa maintenance. La structure des fichiers eux-mêmes diffère. Avro écrit le schéma dans l'entête de chacun des fichiers, ce qui permet de les interpréter facilement, et de gérer sans problème les modifications d'une structure de données. Alors que pour Protobuf, les fichiers ne peuvent être interprétés que si le schéma est fourni à l'outil de lecture. Finalement, le point qui nous a décidés à choisir Avro est son support natif dans les outils d'analyse que nous avons choisi d'utiliser. Avro est supporté par défaut par Hadoop et les outils de son écosystème et par Spark. Contrairement à Protobuf qui au mieux nécessite l'utilisation d'ElephantBird et au pire n'est pas du tout supporté (par Spark notamment).

3.3.4 Outils d'analyse

Le concept MapReduce [29] a été retenu pour traiter les données collectées. Les outils basés sur ce concept répondent à nos attentes pour l'analyse exploratoire. Son principe a été décrit précédemment (en 2.4.1), nous illustrerons ici son fonctionnement en décrivant une application liée à notre usage.

MapReduce

L'illustration de ce mode de traitement d'information présentée ici se base sur les informations de signalisation GTP-C collectées par nos sondes et disponibles dans des fichiers de type Avro. L'exemple présenté se base sur ces fichiers pour en extraire le nombre de terminaux distincts connectés par cellule et par période de 5 minutes. Le processus de traitement est illustré en Figure 3.5 et nous allons le détailler ci-dessous.

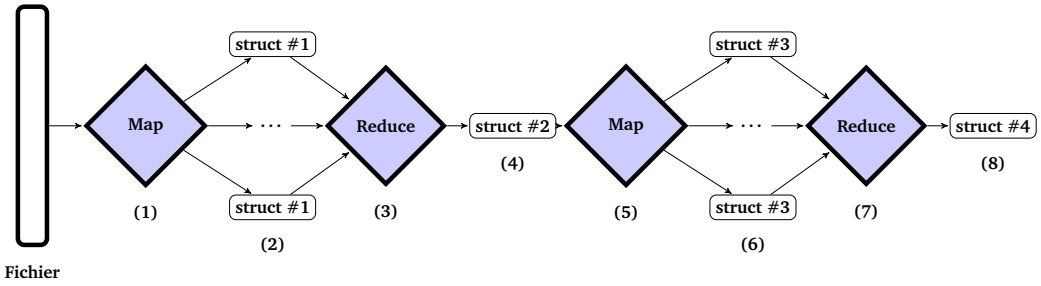


Figure 3.5 : MapReduce.

La source de données présentée aux processus Map est soit une partie d'un fichier divisé en blocs de taille fixe (dans le cas où le fichier est sécable) soit l'intégralité du fichier dans le cas contraire. Le traitement de cette source se fait ligne par ligne, et la division des fichiers par bloc permet de paralléliser cette étape en la déléguant à des processus différents. Nous avons représenté un seul processus Map en Figure 3.5 (1). Nous sommes donc dans le cas où le fichier est insécable ou dans le cas où sa taille est inférieure à la taille d'un bloc.
Pour chaque ligne du fichier, nous allons, à cette étape, créer un couple *clef-valeur* nous permettant de représenter le taux de stationnement par cellule. De ce fait, la clef sera composée de l'identifiant de la période (l'horodatage du message GTP-C arrondi à *300* secondes), concaténé à l'identifiant de la cellule. La valeur est l'identifiant anonyme du terminal client. Le résultat temporaire (*struct #1* en Figure 3.5 (2)) est :

```
(1470217200-00101314000456, 004402597654321)
```

Ce couple est présent autant de fois qu'il y a de messages dans notre fichier. Le regroupement de ces structures par clef est assuré par le processus Reduce représenté en Figure 3.5 (3) La sortie est toujours un ensemble de couples *clef-valeur*, mais ici la valeur est la liste des mobiles ayant été vus dans la cellule à la période considérée (*struct #2* en Figure 3.5 (4)) :

```
(1470217200-00101314000456, [004402597654321, 001016002468013, ...])
```

Ces listes peuvent comporter des doublons, des terminaux passant d'une cellule à l'autre dans le cas d'un effet *ping-pong* par exemple. Une autre tâche, en Figure 3.5 (5) se chargera d'éliminer ces doublons, pour produire *struct #3* (Figure 3.5 (6)) ne comportant que les mobiles distincts sur la cellule et la période.
Finalement un dernier processus Reduce (en Figure 3.5 (7)) retournera la longueur de la liste de chacun des enregistrements dans *struct #4* (Figure 3.5 (8)) :

```
[(1470217200-00101314000456, 63), (1470217500-00101314000321, 22), ...]
```

Cet exemple n'utilise pas les fonctions évoluées disponibles dans les frameworks existants, mais a permis de présenter les structures d'échange et l'enchaînement des traitements possibles.

Spark

Comme nous l'avons vu en 2.4.1, Spark [14] est un ensemble d'outils permettant, comme hadoop MapReduce, de lancer des traitements en parallèle sur des ensembles de données.

Spark propose plusieurs approches de traitement. La première est une interface de type MapReduce enrichie de fonctions permettant par exemple de regrouper les données par clef, de les trier ou encore de les agréger. L'autre approche est de proposer l'accès à une représentation structurée des données via une interface SQL classique ou des fonctions qu'on pourrait rapprocher de celles présentes dans le monde *nosql*. Ces fonctions sont disponibles dans les langages Java, Scala, Python et R.
Spark dispose d'un mode interactif, en Scala ou en Python, permettant de lancer des requêtes ou des traitements au travers d'une invite de commande. Ce mode permet de tester facilement des requêtes et des transformations et ainsi d'accélérer les développements.
Spark fournit également une interface permettant une analyse pseudo temps réel, *spark streaming*. Cette fonctionnalité met en tampon les données lues pendant un

temps paramétrable et y applique des fonctions identiques à celles qui ont pu être développées pour une analyse par lots. Il existe aussi une bibliothèque de *machine learning*, *MLlib* et de fonctions réalisant des analyses en parallèle des graphes, *GraphX*.

Les différents modes d'exécution de Spark permettent au développeur de faire tourner des analyses sur son poste afin de les valider, mais aussi de lancer des tâches sur plusieurs centaines de nœuds afin de profiter de leurs puissances de calcul. C'est l'architecture en mode *cluster* que nous aborderons. Nous présenterons ici les différents éléments des architectures de ces modes, et les outils de distribution des travaux et de gestion des ressources. Ces derniers points peuvent être pris en charge par Spark lui-même, ou par *Mesos* ou *Yarn*. Le fonctionnement de ces derniers sera décrit ci-dessous.

Dans les trois modes, le principe est identique et tel qu'exposé Figure 3.6 ([14] et [112]). Un processus, *SparkContext*, est créé sur l'une de machines du *cluster*. Ce processus peut être lancé, entre autres, par une console Spark ou un exécutable Scala. Le *SparkContext* se connectera à l'entité gérant les ressources du *cluster*, à celle distribuant les tâches et aux nœuds sur lesquels celles-ci sont lancées. Il copiera aussi sur ces nœuds le code à exécuter, ainsi que les bibliothèques dont il dépend. Si l'on détaille le mode autonome représenté en Figure 3.6, en *(a)*, le *SparkContext*

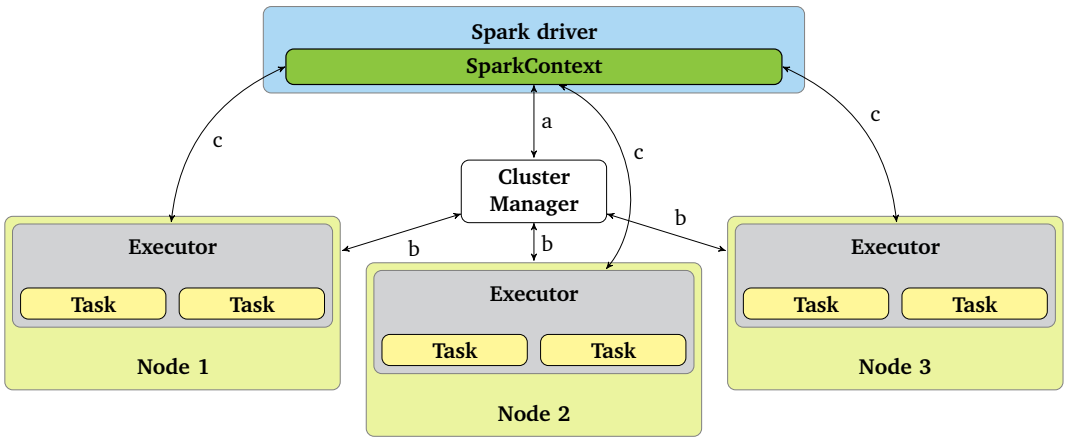


Figure 3.6 : Grappe Spark.

s'adresse au *Cluster Manager* afin d'acquérir les ressources permettant de lancer son traitement sur la grappe. En *(b)*, les environnements d'exécution sont créés sur les nœuds, et finalement en *(c)* le *code* du traitement et les bibliothèques sont copiées sur les *executors* et les tâches peuvent être lancées.

La gestion de ressources, qui va se baser sur la charge des machines et des ressources réservées, peut être réalisée par d'autres logiciels que Spark lui-même. Uti-

liser ces gestionnaires externes permet une plus grande souplesse. Précédemment les exécutables Spark étaient installés sur tous les nœuds, et seul le code d'analyse des données était copié sur les machines. Avec ces solutions, l'exécutable Spark est copié à chaque tâche (ce qui n'augmente que de façon négligeable le temps d'exécution global), ce qui permet de faire tourner sur les nœuds différents logiciels et différentes versions de Spark.

Mesos [13] est un logiciel permettant de réserver de la ressource processeur, mémoire, disque pour des applications sur les nœuds d'une grappe, et de déployer et exécuter ces applications à la volée sur ces mêmes nœuds.
En Figure 3.7 est représentée une architecture basée sur *Mesos*. Nous voyons qu'en *(a)*, le *SparkContext* s'adresse au *Mesos Master* et c'est lui qui gèrera l'allocation de ressources et la distribution des bibliothèques à travers la grappe.

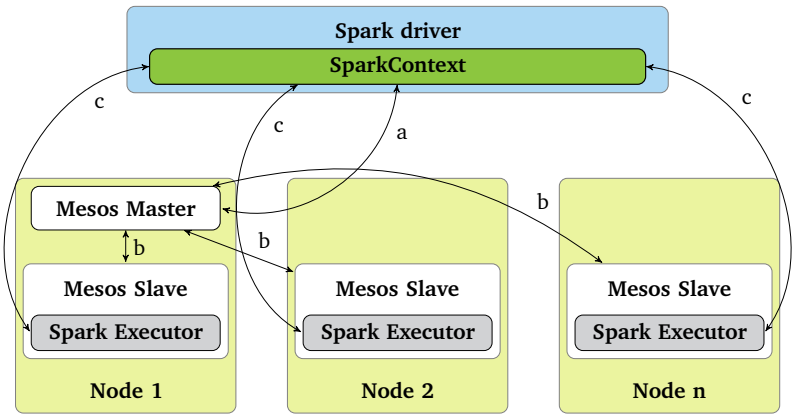


Figure 3.7 : Grappe Spark + Mesos.

Mesos est composé d'un maître centralisant les demandes des clients et distribuant les travaux sur les différents esclaves *(b)*, en fonction de leur nombre et leur charge. Les esclaves créent localement des environnements cloisonnés dans lesquels sont copiés les exécuteurs Spark de même que le code de traitement et les bibliothèques nécessaires *(c)*. Enfin le *SparkContext* lance sur les exécuteurs les différentes tâches.

Un autre ordonnanceur se nomme *Yarn* [11], le principe est proche de celui de *Mesos*. Il s'agit d'un logiciel gérant les ressources présentes dans un *cluster* et le déploiement d'applications. La terminologie change, les éléments gérant globalement les ressources sont les *Yarn ResourceManager*, les processus créant les environnements sur les nœuds sont les *Yarn Node-Manager*. Son avantage est son intégration avec Hadoop (son nom complet est *Apache Hadoop YARN*). Il intègre par exemple la gestion de l'authentification Kerberos [79], permettant ainsi de limiter l'accès aux ressources de traitement, et aux données d'HDFS.
En Figure 3.8, la grappe complète est reproduite, telle que celle que nous avons

mise en place. Sont présents les éléments permettant d'assurer une haute disponibilité, le *cluster* HDFS et un exemple de source extérieure auquel Spark se connecte.

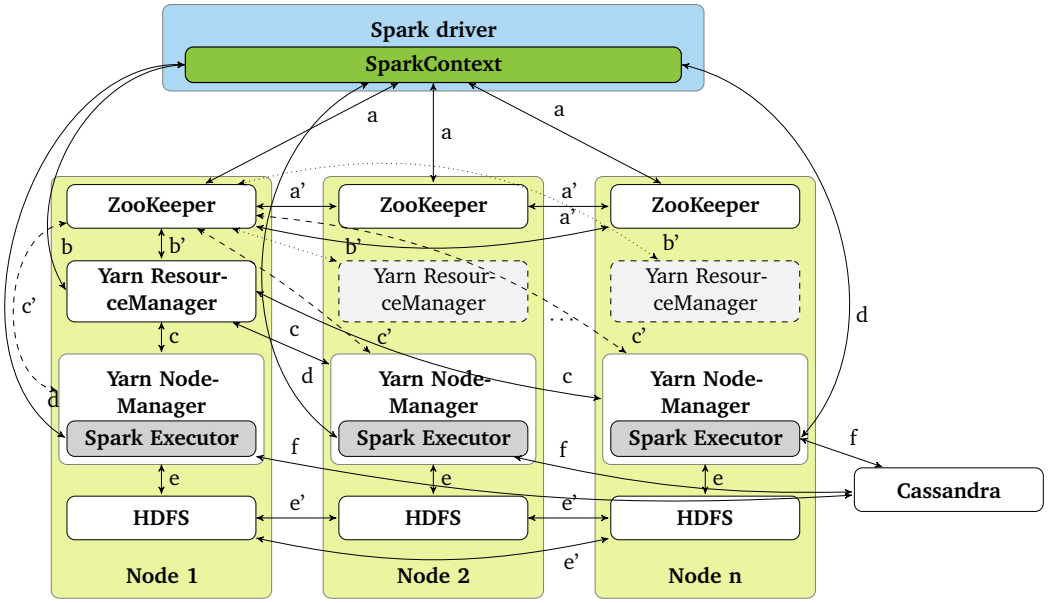


Figure 3.8 : Grappe haute disponibilité Spark + Yarn + HDFS + Cassandra.

La haute disponibilité est assurée par *Zookeeper* et par la mise en place de *Yarn ResourceManager* en mode *actif/passif*. *Zookeeper* est un processus stockant des données dans une mémoire partagée entre différents nœuds d'une grappe *(a')* et gérant l'élection de maîtres d'un *cluster*. Les *Yarn ResourceManager* y souscrivent, élisent leur leader *(b')* (auquel les clients se connecteront pour lancer les taches). De même les *Yarn Node-Manager (c')* et le *SparkContext (a)* se connecteront à *Zookeeper* pour connaître le *Yarn ResourceManager* maître. Ainsi si le maître tombe, l'un des nœuds en attente prendra sa place et *Zookeeper* le signalera aux éléments connectés.
Après avoir récupéré l'adresse du *Yarn ResourceManager* maître auprès de *Zookeeper*, le *SparkContext* s'y connecte pour demander des ressources *(b)*, *Yarn ResourceManager* s'adresse aux *Yarn Node-Manager (c)* pour que ceux-ci créent les environnements. *SparkContext* y copie l'exécutable Spark, le code de traitement et les fichiers associés. Nous avons représenté ici les interactions avec le système de fichier HDFS *(e)* sur lequel les exécuteurs peuvent lire et écrire. Les exécuteurs peuvent aussi communiquer avec des sources de données externes, telles Cassandra *(f)*. Les nœuds HDFS communiquent entre eux *(e')* pour distribuer les données et assurer leur réplication.

Les deux solutions élaborées - avec *Mesos* et *Yarn* - sont robustes et faciles d'utilisation. Elles permettent l'utilisation parallèle de différentes versions de *Spark* et un déploiement aisé des tâches sur les nœuds. L'avantage de *Yarn* - et la raison pour laquelle nous l'utilisons - est son support de Kerberos [79]. Les données que

nous collectons étant sensibles, nous souhaitons y restreindre l'accès et la solution d'authentification adoptée par *Hadoop* est également Kerberos.

3.4 Conclusion

Nous avons construit une solution complète allant de la capture réseau de flux GTP-C à la mise à disposition d'outils de traitement capables de produire, à partir des données extraites, des informations de mobilité humaine pour l'ensemble du territoire français. La conception de cette plateforme est le fruit de réflexions menées, dans les détails, autant sur ses différents composants que sur la chaîne dans sa globalité. Elle est également le fruit d'une expertise dans les domaines des réseaux et des systèmes informatiques.

Les contraintes exercées sur les sondes, en particulier les débits en entrée, ont demandé de revoir la séparation entre les différentes couches composant un système d'exploitation classique, répartissant ses différents rôles entre espace utilisateur et noyau. Les paquets sont copiés directement de la carte réseau à l'espace utilisateur, économisant ainsi de couteuses opérations d'allocation opérées par le *kernel*. Les paquets sont filtrés et la charge est répartie depuis la carte de capture, soulageant de ce fait l'espace utilisateur.

L'infrastructure hébergeant les données a dû également être pensée en fonction des débits. Les sondes de capture génèrent plus de 4 gigaoctets par heure, environ 700 gigaoctets par semaine. De tels volumes sont difficiles à ingérer, à traiter et à requêter sur un système temps réel et classique de base de données. Nous avons préféré aux SGBD un système basé sur Hadoop et Spark, facilitant le stockage et le traitement de gros volumes et fournissant des outils évolués pour l'analyse de données (des bibliothèques d'analyse de graphes et de *machine learning*).

Enfin, nous avons choisi avec soin le format de fichier permettant de communiquer les données entre les sondes et la plateforme de traitement. Notre choix s'est porté sur Avro, un format permettant de lire la donnée telle qu'elle a été écrite, sans traitement ni transformation et économisant les ressources réseaux, disque et *entrée/sortie*, tout en supportant la compression. Avro étant libre, il permet d'échanger nos données avec d'autres systèmes.

Nous avons donc construit une sonde et une plateforme d'analyse GTP-C, supportant d'importants débits, capable de fonctionner chez n'importe quel opérateur, capable de mener des analyses exploratoires et produisant des données exploitables sur d'autres systèmes que le nôtre.

4

Caractérisation et transformation des données issues des flux GTP-C

Ce chapitre a le rôle charnière de vérifier, à partir des données et de l'infrastructure qui ont été présentées, les hypothèses émises dans les chapitres précédents, de décrire les traitements à réaliser sur la matière collectée afin de la préparer pour les travaux à suivre.
Nous n'avons pas trouvé dans la littérature d'éléments nous permettant de confirmer la validité des données de signalisation GTP dans le cadre de l'étude de la mobilité. À notre connaissance, aucune étude n'a cherché à mesurer la densité spatio-temporelle de l'information transportée par ce protocole. C'est pourquoi nous vérifierons ces éléments à partir des données et des procédés exposés dans les chapitres précédents.
Ces mêmes données de signalisation ont besoin d'être mises en forme pour représenter les déplacements d'individus. Ces déplacements seront regroupés en trajectoires anonymes décrivant l'enchaînement de cellules auquel un mobile s'est connecté entre deux longs stationnements.
Les caractéristiques inhérentes au réseau mobile, la taille de ses cellules, le médium d'accès radio et son instabilité en limite de couverture, ainsi que le choix de connexion à une cellule (qui n'est pas fait qu'en fonction de la force du signal reçu, mais aussi de critères dépendants de la charge du réseau), font que la reconstitution brutale d'une trajectoire à partir de vecteur ordonné temporellement des connexions d'un client puisse être bruitée. Il est donc nécessaire de filtrer ces informations pour reproduire les trajectoires au plus proche de la vérité terrain, et de disposer dans la suite de nos travaux de données exploitables.

Nous caractériserons dans ce chapitre la fréquence des messages GTP-C, celle des sessions de données et le temps séparant ces dernières. Nous nous intéresserons également à la distance des sauts entre chaque cellule à laquelle l'individu s'est connecté.
Dans une deuxième partie nous présenterons la manière dont nous construisons les trajectoires, et comment nous en éliminons les aberrations que nous y trouvons.

4.1 Caractérisation

La particularité de nos travaux est de n'être basés que sur le protocole GTP, de ne le corréler ni à des journaux d'appels ni à de la signalisation provenant du domaine voix. Comme nous l'avons vu, les évènements déclenchant un message sont nombreux et le GTPv2 (pour le LTE) a enrichi ce message en créant de nouveaux types en fonction de l'évènement, donnant à leur analyse plus de sens. Le LTE, avec la voix sur LTE (voLTE) a d'autres influences sur la signalisation. La première est que lorsque la voLTE sera mise en place (dans les mois à venir) la connexion de données (GTP-U) sera indispensable au canal voix (les appels se feront en voix sur IP). Mais d'ici là, lorsqu'un client est connecté en LTE et qu'il émet ou qu'il reçoit un appel, il est obligé de passer en UMTS ou en GPRS, provoquant ainsi une augmentation du nombre de messages de signalisation et donc des informations de géolocalisation.
Pour ces raisons, et contrairement à Xu et al. [139], nous pensons que les mises à jour GTP-C sont assez fréquentes pour que les données extraites reflètent la mobilité humaine.
Nous étudierons dans cette partie les temps inter arrivées des différents évènements et la distance des sauts entre ces deux évènements.

Notre jeu de données consiste en 24 heures de données GTP-C collectées sur l'ensemble de la France et rassemblant $49\,029\,361$ sessions.

4.1.1 Fréquence des messages GTP

La densité temporelle des informations géographiques est un paramètre important dans notre domaine d'étude. Bien que le but de nos travaux ne soit pas de connaître la position géographique de manière précise à tout instant de l'ensemble du parc d'utilisateur, il nécessite d'en avoir une connaissance assez précise pour être à même d'observer les déplacements de groupes.
L'estimation de la charge des différents réseaux de transports, et en particulier celle des transports en commun, nécessite de connaître régulièrement la localisation géographique des individus afin d'en déduire leur proximité et d'estimer le nombre d'individus dans un train, ou encore d'être capable d'estimer les changements de modes de déplacement au cours d'un trajet.
Nous avons donc mesuré les intervalles entre les messages de mise à jour pour estimer la plage de temps pendant laquelle nous pourrions observer les mouvements de population, nous avons aussi mesuré les temps de sessions et les temps d'inter-session.

Distribution des inter arrivées

La mesure des temps inter arrivées nécessite de traiter les données stockées dans le cluster Hadoop pour reconstruire les sessions et en extraire les informations temporelles.
La première étape consiste à extraire les données utiles à notre traitement. Nous allons, via une tâche MapReduce ou une requête SQL dans Spark, extraire pour la période étudiée un vecteur $messages$, constitué tel que :

$$messages = [(u_0, t_0, w_0, m_0, s_0, r_0), \ldots, (u_n, t_n, w_n, m_n, s_n, r_n)] \tag{4.1}$$

où :

- u_n est l'identifiant aléatoire du terminal mobile
- t_n est l'horodatage en microseconde de l'évènement
- w_n est l'identifiant de la cellule
- m_n est le type de message GTP
- s_n est le numéro de séquence du message GTP, permettant d'identifier un échange GTP
- r_n est la radio access technology (RAT) utilisée

Les sessions seront construites à partir de ce vecteur, ordonné dans le temps et regroupé par terminal. Les sessions seront identifiées grâce au champ m_n. Une session commence par un *create PDP context response* ($m_n = 17$ en GTP), ou un *create session request* ($m_n = 33$ en GTPv2) ; elle finit par par un *delete PDP context response* ($m_n = 21$ en GTP) ou un *delete session response* ($m_n = 36$ en GTPv2).
L'assemblage de ces données et leur découpage en sessions sont décrits par l'Algorithme 1.

La première étape est de regrouper les messages par identifiant de terminal mobile. Les messages sont regroupés selon cette clef en 1 :

$$\begin{aligned} messageList = [&(u_0, [(t_i, w_i, m_i, s_i, r_i), (t_j, w_j, m_j, s_j, r_j), \ldots]), \\ &\ldots, \\ &(u_n, [t_x, w_x, m_x, s_x, r_x), (t_y, w_y, m_y, s_y, r_y), \ldots])] \end{aligned} \tag{4.2}$$

Nous avons donc une liste de structures *clef-valeur* dont la clef est l'identifiant du terminal. Pour chaque structure de cette liste, la *valeur* est extraite et ordonnée temporellement en 2.
Si le message fait partie d'une session active, nous ajoutons le champ t_n à un vecteur (en 3) :

$$session_n = \langle (t_0, \ldots, t_j) \rangle \tag{4.3}$$

Algorithme 1 : Reconstitution des sessions GTP

Data : $messages$: Liste de messages GTP-C
Result : $sessions$: Liste des sessions
$sessions \longleftarrow []$
$connected \longleftarrow False$
1 $messageList \longleftarrow messages.groupByKey(u_i)$
foreach $mobileTimeLine$ *in* $messageList$ **do**
  $session \longleftarrow []$
2 $timeLine \longleftarrow mobileTimeLine.value.sortByKey(t_i)$
  foreach $event$ *in* $TimeLine$ **do**
    if $event.m$ = *17* || $event.m$ = *33* **then**
      $connected \longleftarrow True$
    if $connected = True$ **then**
3      $session.append(event.t)$
    if $event.m$ = *21* || $event.m$ = *36* **then**
      $connected \longleftarrow False$
      if $session.length$ = *0* **then**
4        $sessions.append(session)$
        $session \longleftarrow []$

Finalement ces sessions sont agrégées dans une structure globale en 4 :

$$sessions = [session_0, \ldots, session_n] \tag{4.4}$$

Les intervalles inter arrivées sont calculés en itérant sur toutes les structures $session$.

En Figure 4.1 nous observons la distribution des intervalles inter arrivés observés sur *24* heures et sur toute la France, à partir des $49\,029\,361$ sessions reconstituées.

Nous remarquons sur cette figure des valeurs particulièrement basses, 50% des valeurs se trouvent en deçà de 11 secondes, et 75% sous 44 secondes. Dans le but de vérifier que ces résultats ne sont pas faussés par la surreprésentation de quelques terminaux causant un grand nombre de messages dans un contexte particulier (oscillant entre deux cellules par exemple), nous avons mesuré les moyennes et les médianes des inter arrivées par utilisateur, cela sur 24 heures et sur toute la France. Les résultats sont visibles en Figure 4.2.

Nous voyons sur cette figure que l'intervalle moyen par utilisateur monte à 782 secondes pour 75% des moyennes d'intervalles par utilisateur. De même, 75 % des intervalles médians sont situés en dessous de 160 secondes.
Ces valeurs sont assez basses pour que nous estimions à ce stade que cette source de données est suffisamment dense temporellement pour répondre à nos besoins. De

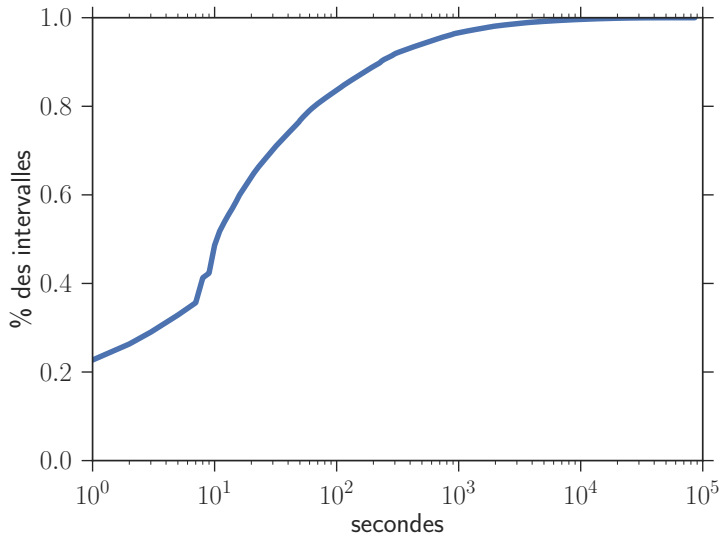


Figure 4.1 : Densité du temps moyen inter-arrivée des messages GTP-C par utilisateur

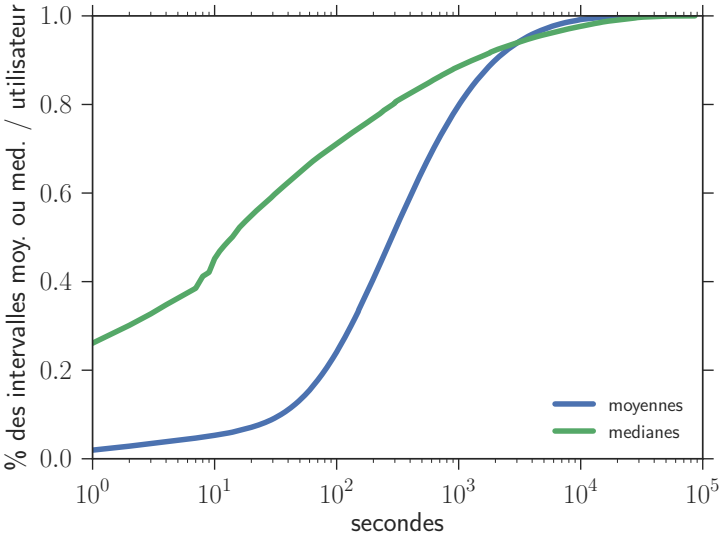


Figure 4.2 : Fonction de répartition du temps moyen inter arrivées des messages GTP-C par utilisateur

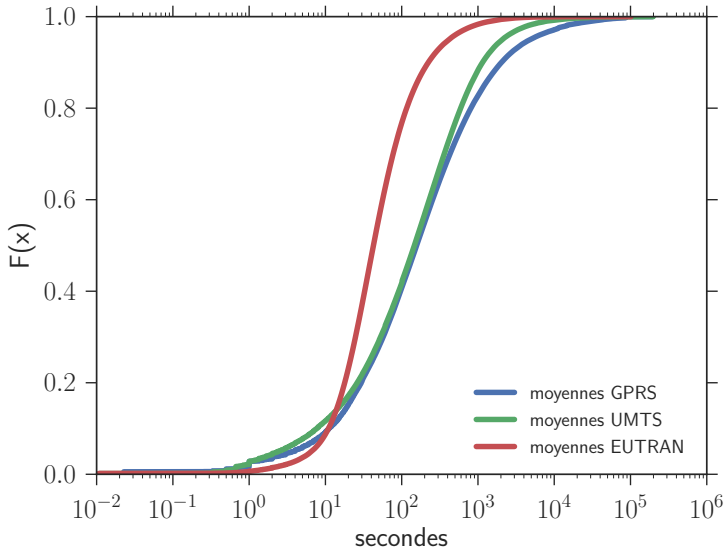
plus la différence entre les distributions brutes et moyennées par utilisateur nous laisse à penser que dans certains cas la fréquence des mises à jour peut être très élevée. Dans le but d'en identifier la cause, nous étudierons les fréquences de messages par technologie d'accès radio (RAT) et par région française.

Distribution des intervalles d'inter arrivées moyens et médians par utilisateur et par RAT Nous avons voulu mesurer l'influence de la technologie utilisée sur le nombre de messages générés. Comme nous l'avons vu en 1.3.2, les différentes technologies d'accès radio (RAT) ne génèrent pas les mêmes messages de signalisation (GTP ou GTPv2), et ne les génèrent pas à la même fréquence. Pour ces mesures nous avons reconstitué les sessions pour lesquelles les clients sont restés connectés en utilisant le même type d'accès du début à la fin. L'inconvénient de ces mesures est que le critère de filtrage utilisé restreint grandement le nombre de sessions étudiées. Nous avons 68% de sessions mêlant les RAT, contre 10% uniquement en LTE, 14% en UMTS et 8% en GPRS.
Les résultats de ces mesures sont représentés en Figure 4.3.

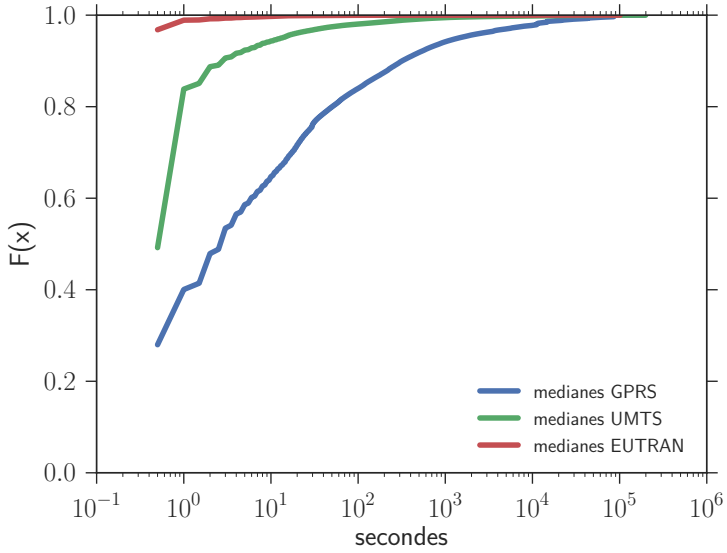
Nous y remarquons, comme attendu, que le LTE génère beaucoup plus de messages (50% : 42 secondes, 75% : 93 secondes) que l'UMTS (50% : 146 secondes, 75% : 458 secondes) et le GPRS (50% : 156 secondes, 577 secondes). Ceci s'explique, comme nous l'avons vu en 1.2.2, par une nature plus verbeuse du protocole GTPv2. La fréquence des mises à jour plus élevée de l'UMTS s'explique par les *Direct Tunnel* produisant plus de messages qu'une session GPRS entre un SGSN et un GGSN.

Distribution des intervalles d'inter arrivées moyens et médians par utilisateur et par département français Un autre paramètre pouvant entrer en compte dans la fréquence des mises à jour GTP-C est la densité de population, et par la même, la densité de cellules associées.
Pour étudier cela, nous avons choisi arbitrairement trois départements français parmi les plus densément peuplés et trois parmi les moins densément peuplés. Le choix de Paris est particulier, car il s'agit en même temps d'une ville et d'un département dont la densité de population est la plus élevée du pays.
Nous avons ici mesuré sur 24 heures la distribution des intervalles inter arrivées moyens et médians par utilisateur.
Les distributions sont tracées en Figure 4.4 et résumées dans le tableau 4.1.

Les valeurs des intervalles augmentent inversement à la densité de population. Exception faite pour la Creuse, un département peu peuplé. Une des causes de ces performances étonnantes pourrait venir des problèmes de couvertures notés dans le département [39]. Une connexion instable forçant le mobile à passer d'une cellule à l'autre produira un grand volume de messages de signalisation.

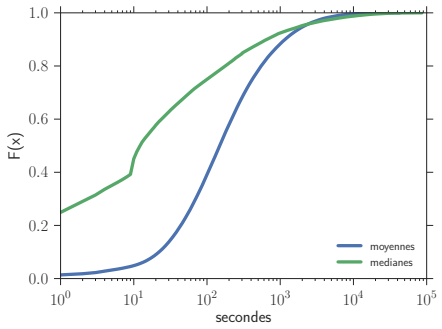


(a) Moyennes

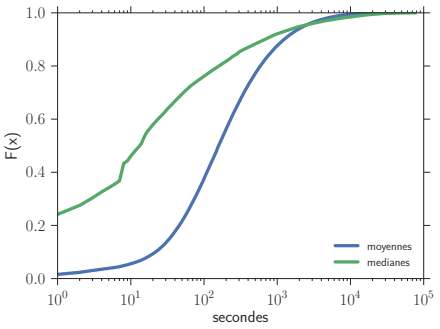


(b) Médianes

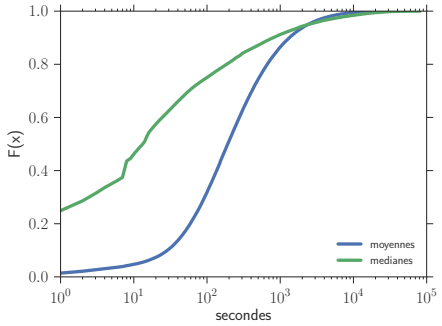
Figure 4.3 : Densité de probabilité du temps inter arrivée des messages GTP-C par utilisateur et par technologie radio



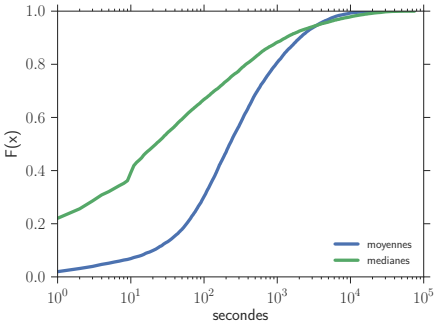
(a) Paris (21 154 hab./km2)



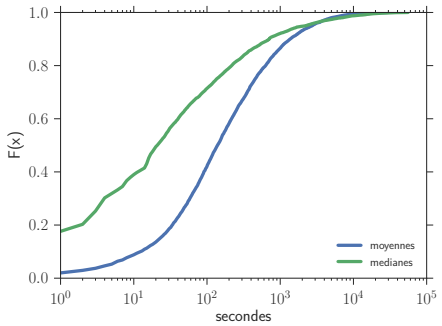
(b) Rhône (548 hab./km2)



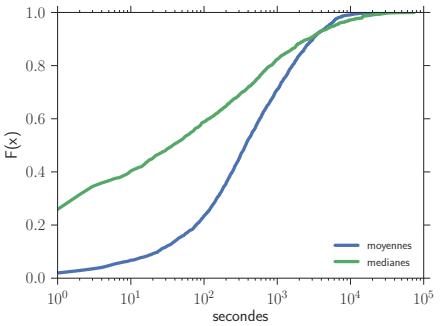
(c) Bouches-du-Rhône (392 hab./km2)



(d) Haute-Marne (29 hab./km2)



(e) Creuse (22 hab./km2)



(f) Lozère (15 hab./km2)

Figure 4.4 : Densité de probabilité du temps inter arrivées des messages GTP-C par utilisateur et par département

Département	Densité	Centiles					
		25		50		75	
		Med.	Moy.	Med.	Moy.	Med.	Moy.
National	98,8 h./km2	1s	105s	14s	284s	160s	782s
Paris	21 153,9 h./km2	2s	57s	13s	150s	101s	416s
Rhône	547,8 h./km2	2s	60s	14s	159s	88s	447s
Bouches-du-Rhône	391,8 h./km2	2s	76s	14s	192s	100s	516s
Haute-Marne	29,2 h./km2	2s	78s	23s	225s	229s	703s
Creuse	21,7 h./km2	3s	46s	21s	140s	136	455s
Lozère	14,8 h./km2	1s	114s	38s	369s	534s	1218s

Table 4.1 : Distribution du temps inter arrivées moyen et médian par utilisateur et par département

Encore une fois, ces résultats sont satisfaisants, en zone densément peuplée 75 % des sessions sont mises à jour en moyenne en moins de 10 minutes.

4.1.2 Transitions entre cellules

La densité temporelle a été traitée dans ce qui précède. Nous mesurerons ici la densité spatiale des informations que nous collectons. Pour ce faire, nous avons construit un graphe *G(V,E)*, où *V* est l'ensemble de nœuds composé par les cellules du réseau mobile, et les arcs *E* représentent les transitions des équipements d'une cellule à l'autre, pondéré par le nombre de terminaux uniques ayant emprunté ce chemin. Ce graphe est directionnel, il comporte 447 867 nœuds et 24 752 802 arcs. La distribution de ses degrés entrants est égale aux degrés sortants et est représenté en Figure 4.5.

Nous pouvons reconnaître sur cette courbe une loi de puissance, soulignant le caractère hétérogène du réseau et des mouvements d'individus se connectant de proche en proche.
En figure 4.6 nous représentons la distribution de la longueur des arcs du graphe exprimée en mètre. La longueur de ces arcs est calculée à partir d'un référentiel que l'opérateur fournit. La clef de chacune des entrées est formée par les identifiants de l'opérateur et du code de la cellule. Et parmi les valeurs attachées à cette clef, nous trouvons l'adresse postale et les coordonnées GPS de la cellule. Les distances sont calculées grâce à la formule de Vincenty pour chaque arc du graphe.
La distribution ainsi calculée est représentée en Figure 4.6.

Nous avons ici des arcs de longueurs assez importantes, 40 % des sauts font plus de 10 km. Pour éviter l'influence d'arcs atypiques sur nos analyses, connectant des

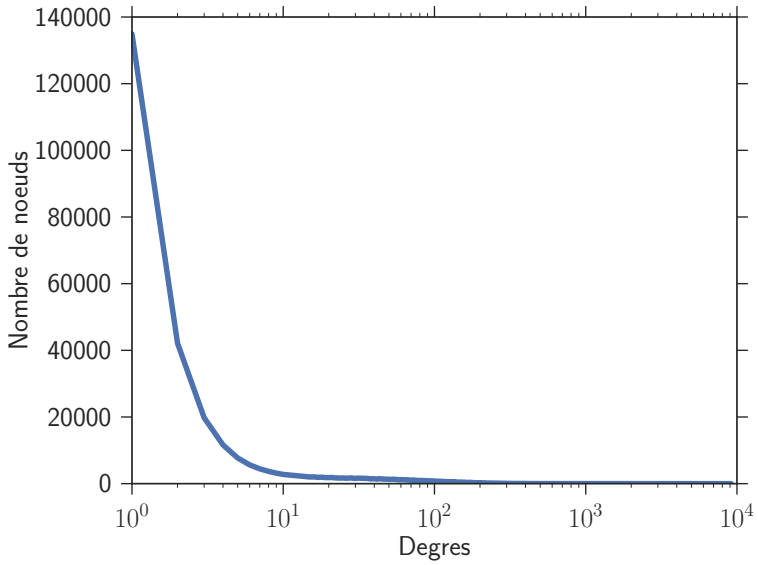


Figure 4.5 : Distribution des degrés du graphe de transitions

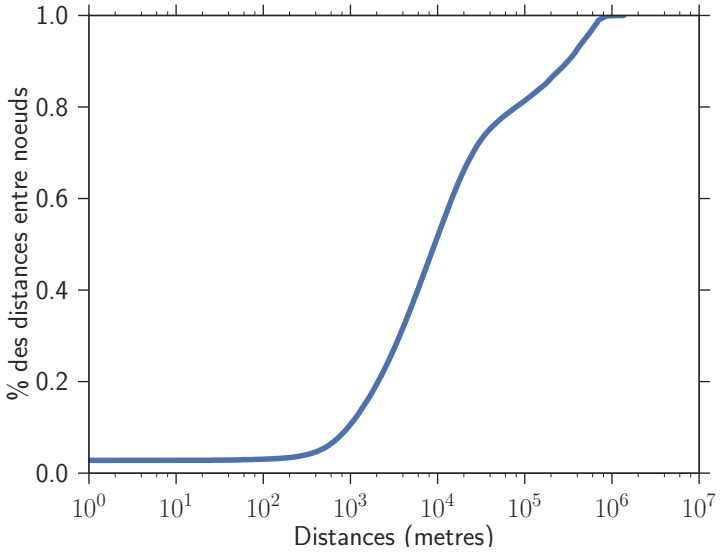


Figure 4.6 : Distribution des distances des arcs

nœuds éloignés (les aéroports, les tunnels, les cellules en bordure de voies ferrées), nous allons calculer les moyennes et les médianes, par nœud des longueurs des arcs. Les distributions de ces valeurs sont tracées en Figure 4.7. Y sont présentées les médianes et les moyennes, mais aussi leurs pendants pondérés. Nous avons utilisé le poids des arcs dans nos calculs pour faire ressortir les chemins les plus empruntés.

Nous pouvons voir que le voisinage des nœuds est assez éloigné. La moitié des nœuds est éloignée en moyenne de ses voisins de plus de 54 km (ce qui conforte notre mesure précédente). Mais en pondérant cette valeur, la distance moyenne

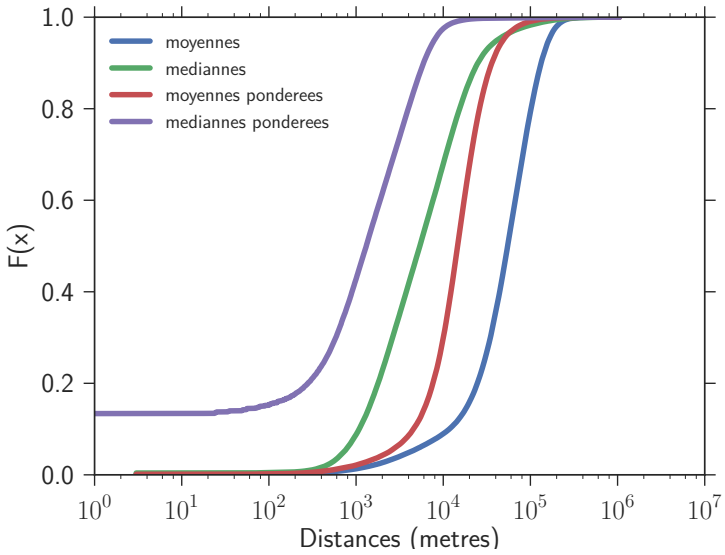


Figure 4.7 : Distribution des distances des arcs

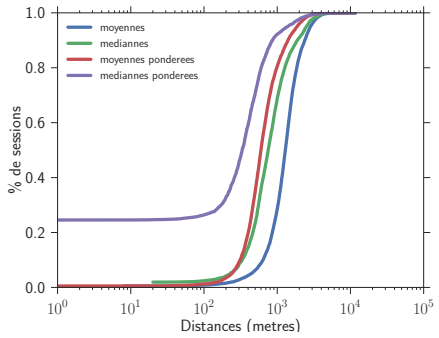
chute à 14 km. Ce qui semble indiquer que, bien que connectées à des nœuds éloignés, les transitions les plus nombreuses se font vers les nœuds les plus proches.

Ces transitions sont réalisées sur l'ensemble du territoire. Nous allons réduire nos observations à l'échelle du département pour observer l'influence de la densité de cellule et de population sur ces valeurs.
Les départements sont les mêmes que ceux choisis pour les intervalles inter arrivées. En Figure 4.8 les distributions sont tracées, et le tableau 4.2 résume les valeurs observées.

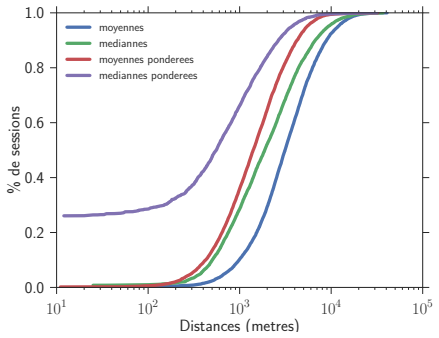
Département	**Densité**	**Centiles**					
		25		50		75	
		Med.	Moy.	Med.	Moy.	Med.	Moy.
National	98,8 h./km2	0,4km	8,8km	1.3km	14.6km	3.3km	23.1km
Paris	21 154 h./km2	0,04km	0,4km	0,3km	0,6km	0,5km	0,9km
Rhône	548 h./km2	0km	0,7km	0,5km	1,4km	1,4km	2,6km
Bouches-du-Rhône	392 h./km2	0,09km	1km	0,6km	1,8km	1,6km	3,2km
Haute-Marne	29 h./km2	1,1km	3,8km	4,3km	5,7km	6,7km	8,2km
Creuse	22 h./km2	1,9km	3,7km	4,9km	5,4km	6,9km	8km
Lozère	15 h./km2	2,1km	4,4km	5,5km	7,2km	9km	10,7km

Table 4.2 : Distribution des distances moyennes et médianes entre cellules du graphe pondérées par le poids des arcs ; par département

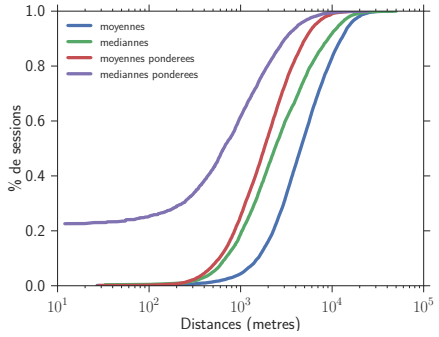
Ici, la longueur moyenne et médiane des arcs suit la densité de population. Il est visible en Table 4.2 que les distances croissent en même temps que la densité décroît, sans exception (la Creuse suit la même règle).



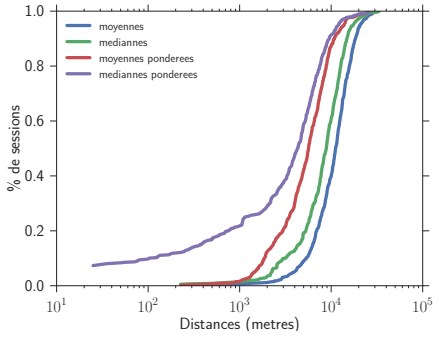
(a) Paris (21 154 hab./km2)



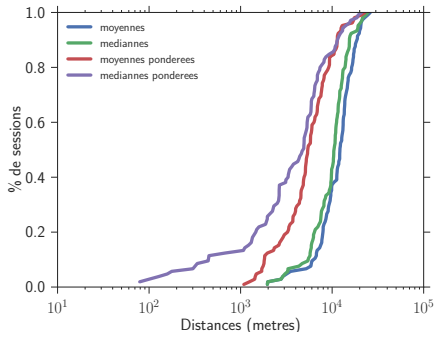
(b) Rhône (548 hab./km2)



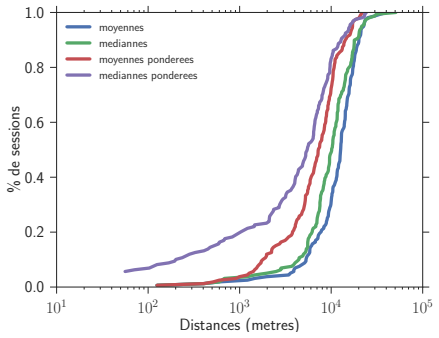
(c) Bouches-du-Rhône (392 hab./km2)



(d) Haute-Marne (29 hab./km2)



(e) Creuse (22 hab./km2)



(f) Lozère (15 hab./km2)

Figure 4.8 : Densité de probabilité de la distance moyenne et de la distance médiane entre cellules du graphe.

Les résultats présentés dans cette partie nous satisfont. Le contexte PDP de 75 % du parc de terminaux est mis en moyenne à jour plus d'une fois par 15 minutes, à l'échelle nationale. Si l'on observe ces fréquences de mises à jour au niveau urbain, elles augmentent à plus d'une fois toutes les 10 minutes.
De même pour la distance séparant les nœuds du graphe de transition. La distance moyenne entre eux est très intéressante en milieu urbain (sous le kilomètre à Paris), et reste certainement exploitable à l'extérieur des villes.
Donc le GTP est une base d'analyse que nous utiliserons pour l'analyse de la mobilité.

4.2 Construction de trajectoires

La mesure du taux d'occupation d'un axe de communication, l'observation de la périodicité dans les déplacements de populations ou encore l'étude de la propagation d'un virus biologique ; toutes ces tâches ont besoin d'une source de données stable et prétraitée, prête à être exploitée. La donnée brute ne peut pas être utilisée telle quelle. Le temps de traitement pour en extraire les données à chaque tâche serait trop long, et, dans le cas de figure le moins favorable, les procédés mis en place seraient incompatibles et redéfinis à chaque fois et ceci par chacun des chercheurs. C'est pourquoi nous avons décidé d'utiliser une structure simple représentant les trajectoires des individus de façon totalement anonyme.

Nos données sont issues de la signalisation GTP (le flux GTP-C). Une fois celui-ci disséqué, sérialisé et stocké sur en HDFS, notre plateforme traite ces informations afin de construire une structure capable de représenter les mouvements de population.
Nous en extrayons la séquence *messages* présentée en équation 4.1 à laquelle est appliquée la transformation $\mathcal{T}$ dans le but d'en extraire des trajectoires $[g_1, \dots g_n]$. Une trajectoire donnée g_i est un vecteur d'identifiants géographiques horodatés $g_i = \langle (t_0^i, w_0^i), \dots, (t_n^i, w_n^i) \rangle$ où t_0^i est la date de l'évènement, et w_0^i est un identifiant pointant vers la localisation géographique de la cellule. Et une trajectoire s'arrête si l'équipement reste attaché à la même cellule pendant plus de 30 minutes (c'est-à-dire si $t_{i+1}^i - t_i^i > 30\ mins$ et $w_{i+1}^i = w_i^i$).

$$\begin{aligned} \mathcal{T}(\mathcal{S}) &\rightarrow [g_1, \dots g_n] \\ where\ g_i &= \langle (t_1^i, w_1^i), \dots, (t_n^i, w_n^i) \rangle \end{aligned} \tag{4.5}$$

Ainsi le résultat ne contient plus aucun identifiant utilisateur, il s'agit d'une liste de paires composée par des identifiants géographiques et des dates.

Nous avons cherché à valider notre approche en utilisant des trajectoires construites à partir de données capturées pour la période du 4 au 11 août 2014.

Pendant cette période, nous avons mesuré le nombre de trajets que nous avons généré par terminal ainsi que l'intervalle inter arrivées de ces trajets. Le résultat de ces mesures est présenté en Figure 4.9. En abscisse est représenté le nombre de trajectoires par semaine et par terminal. En ordonnée, l'intervalle entre ces mêmes trajectoires. La couleur indique le nombre de fois où plusieurs utilisateurs faisant le même nombre de trajets par semaine ont eu le même intervalle entre deux de ces trajets.

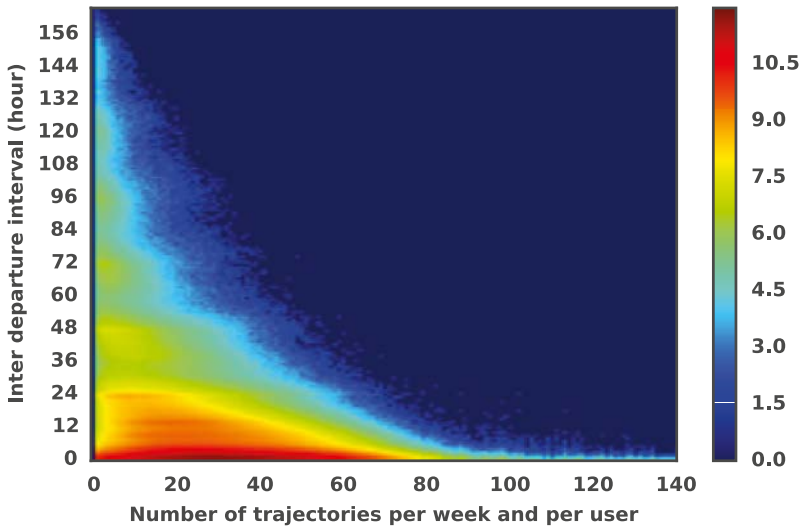


Figure 4.9 : Distribution de l'intervalle d'inter arrivées en fonction du nombre de trajets par semaine

Nous remarquons que le point chaud se trouve autour de 30 trajets par semaine, séparés par très peu de temps. Nous remarquons également pour des valeurs de 4 à 40 trajectoires par semaine, des zones chaudes, aux alentours d'intervalles de temps de 8h, 12h et 24h. Marquant de manière précise la périodicité dans les déplacements.

4.3 Filtrages

Les trajectoires, nous venons de le voir, sont exploitables facilement et donnent rapidement des résultats pour les aspects temporels. Les observations spatiales sont moins immédiates. La donnée est bruitée, elle doit être filtrée.
À titre d'exemple, voici en Figure 4.10 des traces me concernant pour la journée du

5 décembre 2015 de 14h19 à 16h07. Chaque segment représenté est une transition d'une cellule à une autre. Il est numéroté par ordre d'apparition dans la chronologie. De ce fait, chaque segment numéroté plusieurs fois est une transition faite plusieurs fois.

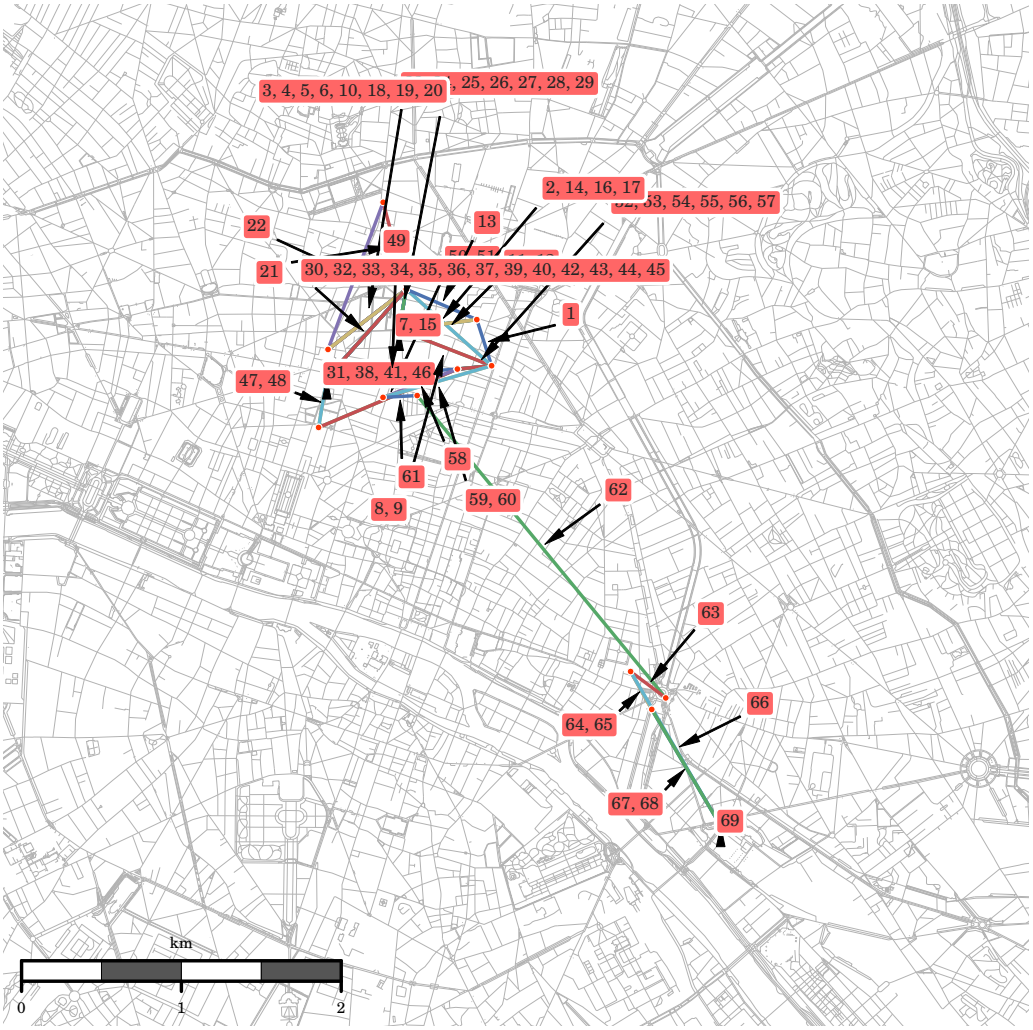


Figure 4.10 : Trajet du 5/12/2015 de 14h19 à 16h07

On remarque que ce trajet est fortement bruité, et qu'il est indispensable de filtrer les transitions superflues.

4.3.1 Filtrage protocolaire

Notre première approche a été de ne pas filtrer les évènements GTP-C écrits par les sondes. Ainsi, si les sondes manquent des paquets (une réponse à un update PDP context par exemple), nous serions malgré tout capables de connaître la dernière position du mobile (indiquée dans le message de type *request*).

Cette approche naïve ne prend pas en compte la possibilité de paquets orphelins, un équipement mal configuré ou bogué peut continuer à répondre à une requête antérieure sans cesse ou à envoyer en boucle une requête alors qu'il a déjà eu une réponse. Ces paquets peuvent provoquer de grandes erreurs dans notre processus de construction de trajectoires. Nous voulons un système robuste, c'est pourquoi nous avons décidé d'implémenter un filtre GTP-C à état. Nous ne considérons que les réponses GTP-C si nous avons précédemment reçu une requête. Le numéro de séquence GTP est l'élément identifiant un échange.

4.3.2 Rayon de giration

La connexion à une cellule étant décidée par le réseau cellulaire, le mobile ne décide pas de se connecter à un équipement plutôt qu'à un autre en fonction de la qualité du signal, il reçoit l'ordre du réseau lui indiquant à laquelle se connecter, en fonction de sa charge. Par conséquent, même si un mobile est statique et que la réception du signal est stable, il peut passer d'une cellule à l'autre, en fonction de la charge du réseau. De notre point de vue, passer d'une cellule à une autre est considéré comme un mouvement physique et donc comme une trajectoire. Nous devons donc détecter et filtrer les petits mouvements parasites.

Afin de nettoyer nos données de ces faux positifs, nous utilisons le rayon de giration [47] (eq. 4.6) des trajectoires comme un seuil de filtrage.

$$r_g^a(t) = \sqrt{\frac{1}{n_c^a(t)} \sum_{i=1}^{n_c^a} (\overrightarrow{r_i^a} - \overrightarrow{r_{cm}^a})^2} \tag{4.6}$$

Le rayon de giration est la racine carrée de la moyenne des distances au carré des différents points d'une trajectoire vers le barycentre de ces points. Nous avons décidé de ne conserver que les trajets dont cette valeur est supérieure à 1 km. En deçà de cette valeur, l'utilisateur est considéré statique.

4.3.3 Recursive look-ahead filter

Une autre source de points aberrants dans les trajectoires peut venir de la géographie de la zone étudiée. Une antenne placée sur le sommet d'une colline peut être captée sur une distance pouvant dépasser 10 km. Pour éliminer ces points, qui ne nous apprennent rien sur la position du mobile, nous utilisons le filtre Recursive Look-Ahead [57] (Algorithme 2).

Algorithme 2 : Recursive Look-Ahead Filter

Data : L : Liste horodatée de coordonnées géographiques
Result : L : Liste horodatée de coordonnées géographiques (filtrée)

```
L ⟵ L.sortByKey(t_i)
for i ⟵ 0 to L.size do
    if i > 0 and i < L.size - 1 then
        v ⟵ distance(L[i − 1].pos, L[i].pos)/(L[i].time − L[i − 1].time)
        if v > V_supersonic then
            d_1 ⟵ distance(L[i + 1].pos, L[i].pos)
            d_2 ⟵ distance(L[i + 1].pos, L[i − 1].pos)
            if d1 > d2 then
                L.remove(i)
            else
                L.remove(i − 1)
return(L)
```

Ce filtre supprime des trajectoires les cellules auxquelles un utilisateur ne peut pas se connecter dans un intervalle de temps donné. Il se base sur l'estimation de la vitesse de déplacement du mobile observée. Cette vitesse est déduite de la distance entre deux antennes et de l'intervalle de temps entre l'entrée dans la zone de couverture d'une de ces antennes et l'entrée dans la zone de couverture de l'autre. Si cette vitesse est supérieure à une variable notée $V_{supersonic}$, le point est supprimé. L'algorithme ici est modifié. Nous supprimons également le second point si la vitesse est égale à 0, nous conservons ainsi des trajectoires les plus courtes possible.

En Figure 4.11, le trajet du 5 décembre 2015 est représenté filtré.

Ce trajet non filtré est composé de 70 segments. Il est réduit à 30 par ce filtre et ceci sans supprimer d'informations.

4.3.4 Détection et suppression des oscillations

Ce filtre a été conçu par Wu et al. [135], contrairement au *Look Ahead Filter*, il ne prend pas la vitesse en argument, mais il détecte les oscillations (les aller-retour entre plusieurs cellules dans un court laps de temps) et supprime les points les plus éloignés et les moins visités.
L'algorithme se décompose en une phase de détection des oscillations. Sur une fenêtre glissante de *a* secondes les cellules différentes vues sont comptées, si ce nombre dépasse le seuil *b* et si le nombre de cellules distinctes contenues dans cet ensemble est supérieur à la valeur *c*, alors la séquence est considérée oscillante.
La deuxième étape consiste à itérer sur les cellules avant et après la séquence selon deux critères : les cellules rencontrées doivent être présentes dans la séquence

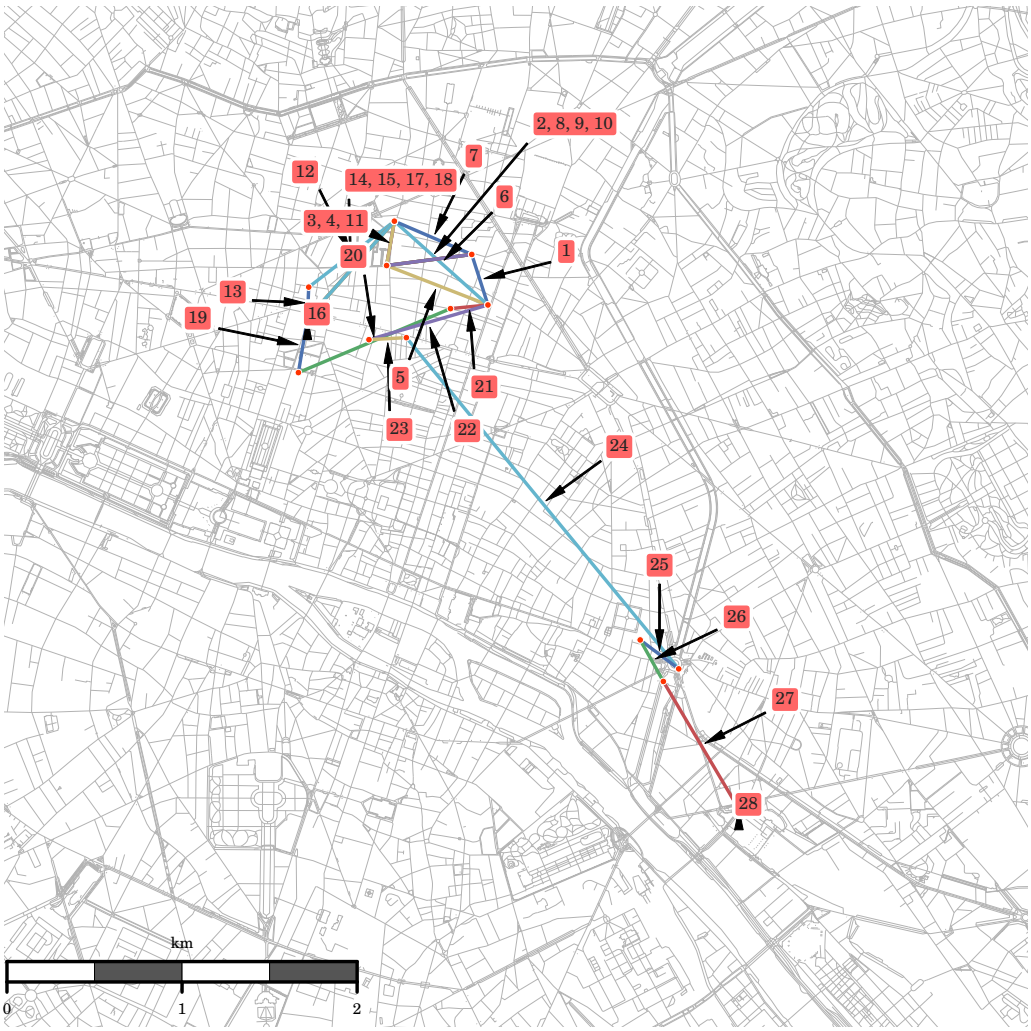


Figure 4.11 : Trajet du 5/12/2015 de 14h19 à 16h07, filtré par le Look Ahead Fitler

oscillante, et la différence temporelle entre l'évènement et la séquence ne doit pas dépasser la valeur d.
L'oscillation isolée, les cellules y sont notées et filtrées selon l'algorithme 3. Celui-ci calcule un *score* en fonction de la distance des cellules entre elles et du nombre de fois où la cellule est vue pendant l'oscillation. La cellule qu'on voit le plus souvent et qui est la plus proche des autres en moyenne obtient le score le plus élevé. Et c'est la seule qu'on conserve, les autres sont supprimées.

Ce filtre, appliqué au trajet du 5 décembre 2015, est tracé en 4.12. Le nombre de segments passe ainsi de 70 à 33. Soit trois segments de plus par rapport au filtre précédent. Nous observons que les oscillations en fin de parcours ne sont pas supprimées.

Algorithme 3 : Oscillations : Notation et suppressions des cellules

Data : C : Liste de coordonnées géographiques de cellules dans une période oscillante

Result : $Cell$: cellule issue du filtrage

$Score \longleftarrow \{\}$

$Freq \longleftarrow \{\}$

$Dist \longleftarrow \{\}$

foreach *cell in* C **do**

  $Freq[cell] += 1$

foreach *distinct(cell) in* C **do**

  $Dist[cell] = avgDistance(cell, C - \{cell\})$

  $Score[cell] = Freq[cell]/Dist[cell]$

$Score = Score.sortByVal(reverse = True)$

return($Score[0].key$)

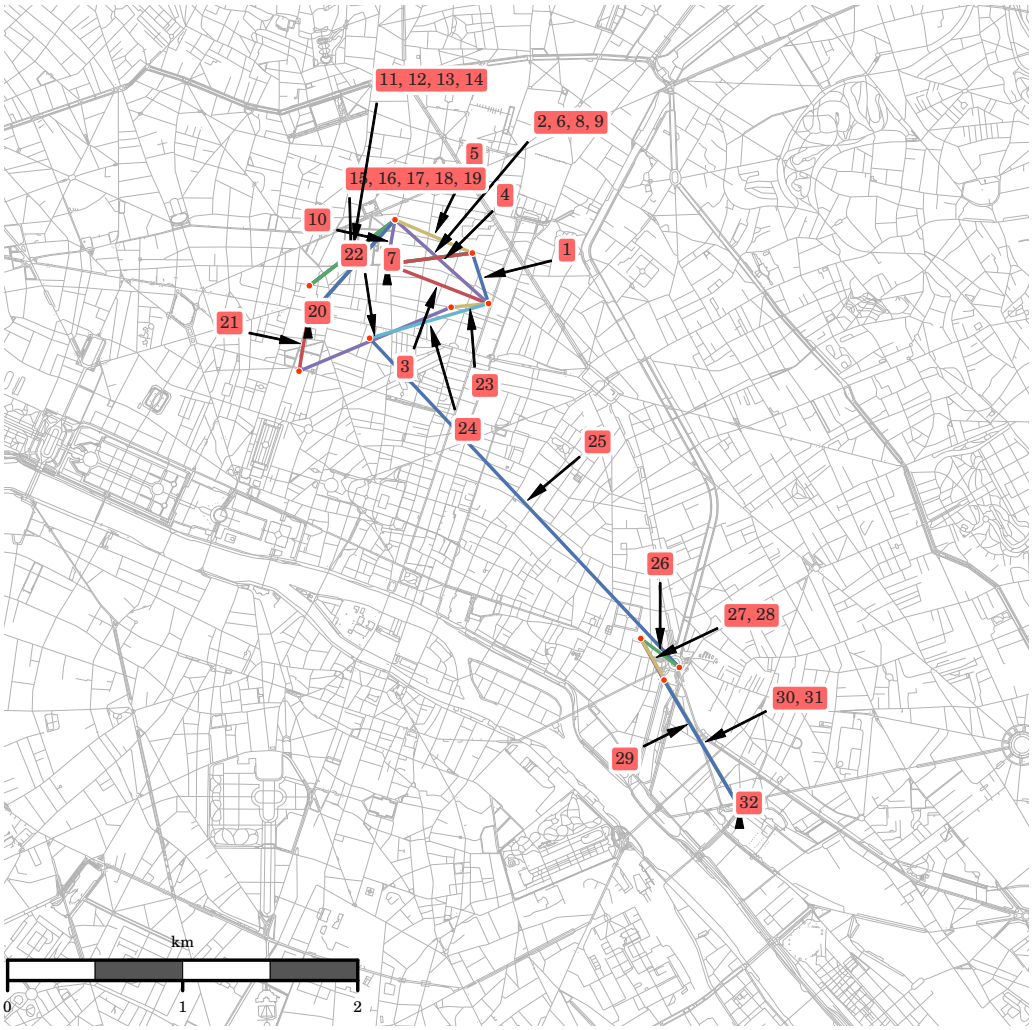


Figure 4.12 : Trajet du 5/12/2015 de 14h19 à 16h07, filtré par la détection et suppression d'oscillations

4.3.5 Performances des Filtres

Nous venons de voir que les performances en milieu urbain semblent être meilleures pour le *Look Ahead Filter*. Le résultat comporte moins de segments et ne dénature pas la trajectoire.
Nous avons appliqué ces deux filtres sur un trajet en train effectué entre 19h26 et 19h50 dans le sud de Paris (en Figure 4.13).

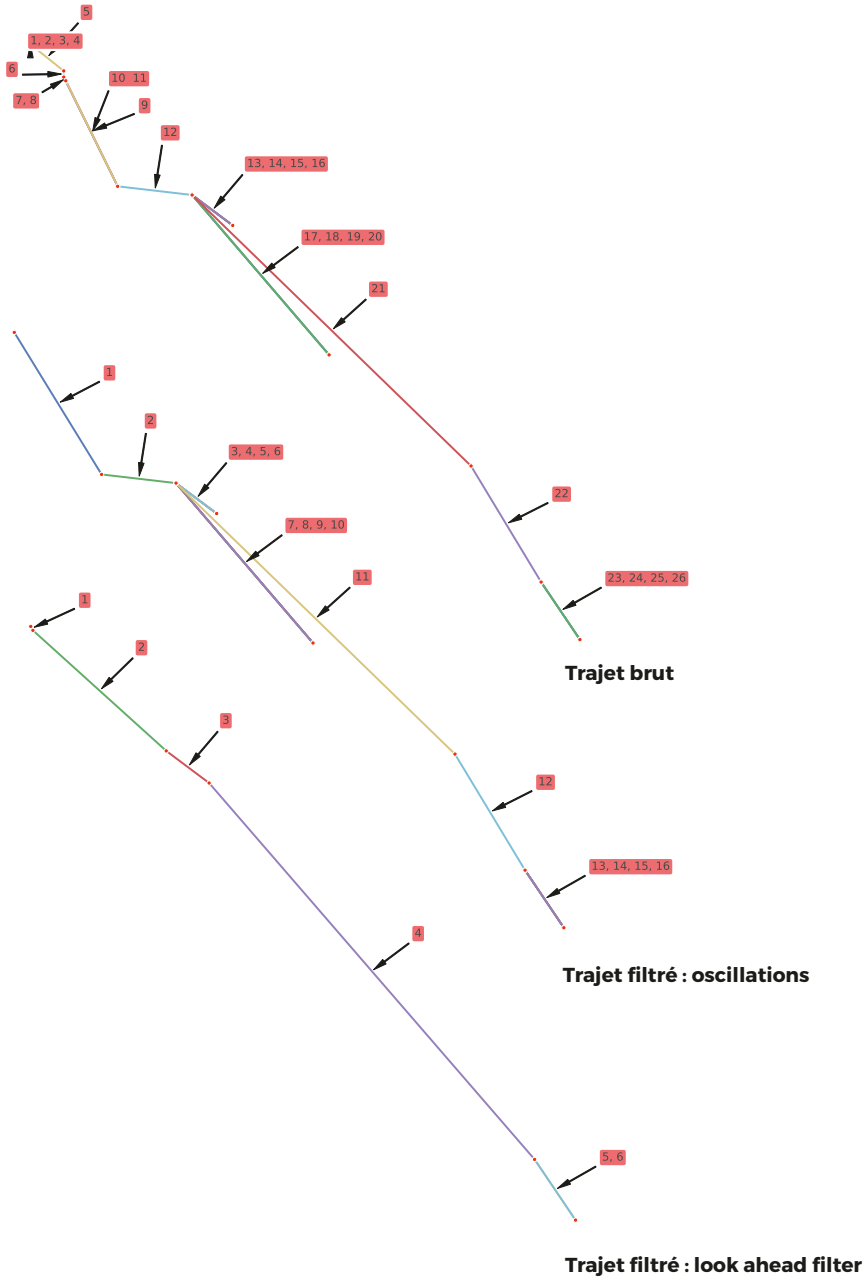
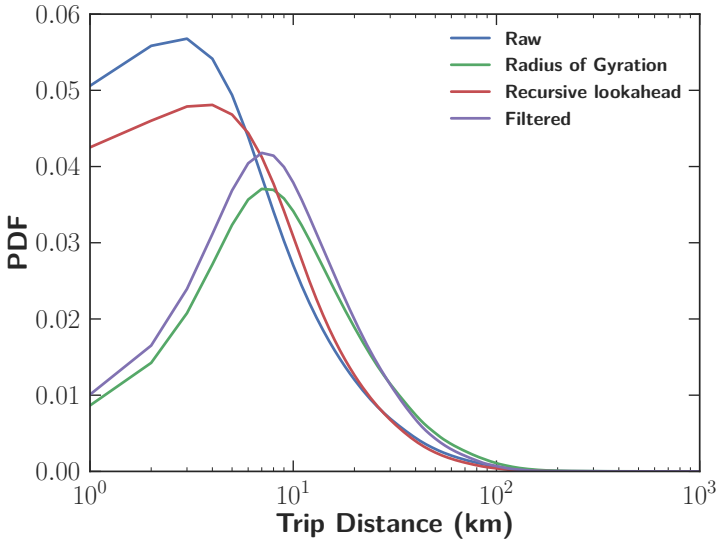


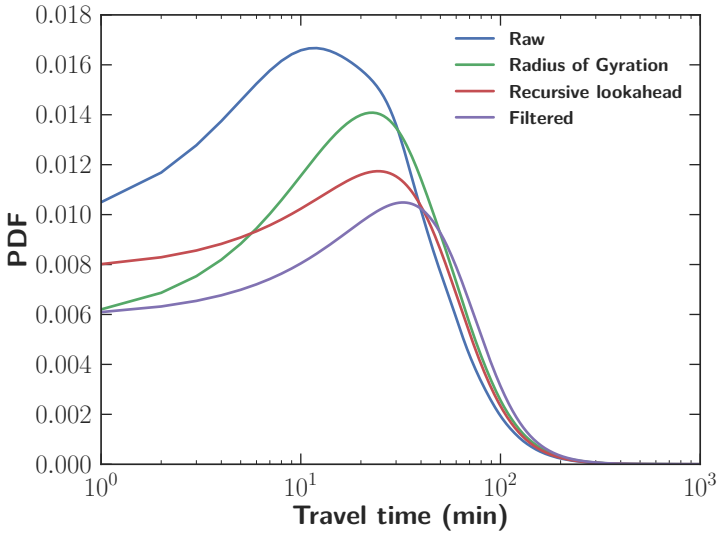
Figure 4.13 : Trajet en train de 19h26 à 19h50

Encore une fois les performances du *Look Ahead Filter* semblent meilleures que celle du filtre basé sur la détection d'oscillations. Le trajet brut est constitué de 27 segments filtrés par la détection d'oscillations, il est ramené à 17 et à 7 avec le second. Un grand nombre d'oscillations disparaît grâce au *Look Ahead Filter*, mais les transitions sont ici légèrement modifiées, la trajectoire est plus rectiligne.
À ce stade, les performances de ce filtre nous satisfont, nous présentons en Figure 4.14c les caractéristiques des trajectoires observées sur le réseau sur 24 heures. Ces caractéristiques portent sur la distance séparant l'origine de la destination de ces parcours, leur temps et le rapport de la distance OD sur la somme de la longueur des segments de la trajectoire. Y sont représentés les effets du filtrage par le rayon de giration et celui de la suppression des aberrations.

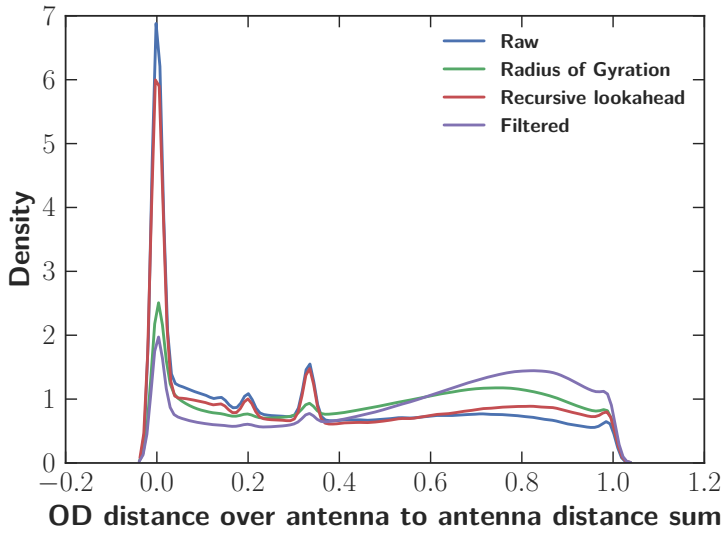
Ces courbes nous permettent d'apprécier les effets des différents filtres, et nous donnent quelques informations sur la nature des trajectoires.
Le filtrage sur le rayon de giration, comme attendu, supprime les trajets de courtes distances, et par conséquent ceux dont la durée est courte. Nous le voyons en Figure 4.14c *(a)* et *(b)*, le sommet des courbes filtrées se déplace vers des valeurs plus importantes que pour les données brutes. En *(c)*, ce filtrage réduit le nombre de trajectoires dont le ratio est proche de 0. Ce qui est lié de manière évidente à la nature du filtre (qui supprime les trajets dont la distance moyenne entre les cellules le composant se trouve sous un seuil). Un de ses buts est la suppression des faux positifs (dans le cas où un terminal est statique et que sa connexion oscille entre plusieurs cellules), la distance OD sera ici celle séparant les deux cellules, et la somme des segments sera fonction du nombre d'oscillations. Le coefficient obtenu en divisant la distance OD par la somme des segments peut ainsi tendre vers 0 très rapidement.
Nous pouvons remarquer l'effet du *Look Ahead Filter* sur les courtes distances et les trajets brefs (en *(a)* et *(b)*). Cet effet semble montrer que de nombreuses oscillations sont présentes dans les petites trajectoires. L'effet de ce filtre sur les distances globales n'a pas le même impact que le précédent, mais il fait chuter la densité des trajets pour d'autres raisons. Son impact ne se fait pas que sur les courts trajets, mais sur leur ensemble. Là où uniquement les courts trajets étaient supprimés précédemment, ce filtre lisse toutes les trajectoires. On voit tout même en *(c)* une représentation plus importante des trajets longs.
Finalement, la combinaison des deux filtres est appliquée aux données, elle a l'effet de modifier la représentation en déplaçant les densités vers des trajets plus longs en termes de distances et de temps, et de garder des trajets plus rectilignes, en *(c)* nous voyons la représentation des rapports proche de 1 augmenter.



(a) Distribution de la distance Origine-Destination.



(b) Distribution du temps de parcours.



(c) Distribution du rapport de la distance OD sur la somme des longueurs des segments d'un trajet.

Figure 4.14 : Caractéristiques des trajectoires filtrées.

4.4 Conclusion

Dans le but d'estimer la précision que peuvent apporter les données GTP-C dans le cadre de l'étude de la mobilité, nous avons mesuré dans ce chapitre la distribution des intervalles d'inter arrivées des messages de mise à jour des contextes PDP et la distribution des distances des transitions faites par les utilisateurs entre chaque cellule.
Nous avons mesuré que les intervalles de mise à jour croissent inversement à la densité de population des départements considérés. Nous passons d'un intervalle moyen de mise à jour par utilisateur inférieur à 500 secondes pour 75 % des départements peuplés à une valeur moyenne inférieure à 1212 secondes pour 75 % des équipements connectés en Creuse. De même, la distribution de sauts entre cellules croît en même temps que la densité de population décroît. Cette dernière tendance peut s'expliquer par le fait que les investissements d'infrastructure des opérateurs se font en fonction de la population à couvrir : ainsi, plus la densité de population est basse, plus les cellules sont grandes. La taille moyenne de ces sauts est particulièrement intéressante à Paris où elle est inférieure à 1 km pour 75 % des cellules. Elle semble difficilement exploitable par contre sur les trajets nationaux où elle atteint 23 km pour le même pourcentage de la population.

De futurs travaux permettraient d'identifier la cause des fréquences de mise à jour très élevées dans la distribution globale (que nous remarquons en Figure 4.1) qui disparaissent lorsque nous traitons ces distributions en les regroupant par utilisateur. De plus, les valeurs observées ont besoin d'être qualifiées à l'aide de mesures faites sur le terrain. Il semble évident que les données urbaines (pour Paris) sont suffisamment précises pour servir de base aux analyses de la mobilité humaine. Mais qu'en est-il des données de la Haute-Marne ? CT-Mapper, présenté dans le chapitre suivant, est une approche permettant de répondre à ces questions.

Nous proposons également dans ce chapitre une structure simple permettant de décrire le trajet d'un terminal à partir des messages GTP-C. Cette structure, que nous avons simplement nommée *trajectoire*, est un vecteur de tuples constitués d'un identifiant de cellule et de la date à laquelle le terminal s'y est connecté. Cette structure sera utilisée par CT-Mapper.
Une exploitation simple de la trajectoire (présentée en 4.2) valide les traces GTP-C et la construction des trajectoires. Elles permettent d'observer la périodicité des déplacements des utilisateurs avec des fréquences de voyages situés aux alentours d'une fois toutes les 8, 12 et 24 heures.
Cependant, ces trajectoires contiennent aussi des faux positifs, déclenchés par exemple par des déplacements à l'intérieur de bâtiments, par des procédures d'équilibrage de charge ou encore par des instabilités radio. Placés sur une carte, nous pouvons

aussi remarquer que ces trajets sont bruités et qu'il est nécessaire de les lisser et de les nettoyer.
Nous avons donc implémenté et testé deux types de filtres. Le premier type supprime les trajets courts et est basé sur le rayon de giration. L'autre catégorie supprime les aberrations apparues pendant une trajectoire en supprimant les cellules éloignées sur lesquelles le client ne se connecte qu'un bref instant et les séquences pendant lesquelles un client se connecte sur un grand nombre de cellules pendant un court laps de temps.
Ces filtres appliqués à nos données semblent efficaces, de futurs travaux permettraient de comparer, dans notre cas, les distances entre nos trajets filtrés et des trajets réels.

5 CT-Mapper

5.1 Introduction

L'analyse macroscopique des flux dans de grandes aires métropolitaines est une tâche ardue. Ceci est particulièrement vrai lorsque de multiples entités sont en charge des différents réseaux de communication (route, train, métro ...). Dû au manque des sources d'information unifiées au travers des systèmes de transport, il est souvent difficile pour les services d'urbanisme ou d'aménagements du territoire d'avoir une vue exhaustive des motifs de mobilité humaine. Dans ce contexte, les données extraites du réseau de téléphonie mobile sont récemment devenues une source d'information intéressante concernant le comportement en mobilité. Grâce à l'omniprésence des téléphones portables dans nos vies, l'analyse des données issues de ceux-ci devient un champ d'investigation prometteur dans la compréhension des déplacements humains multimodaux [104, 34, 118] permettant d'identifier le trajet journalier d'un mobile et ainsi d'enregistrer les habitudes de transport (train, métro, bus, etc ...) au sein de grandes métropoles. Les approches traditionnelles dans l'étude de la mobilité utilisent le GPS afin de percevoir finement les données spatiales avec une précision inférieure ou égale à 50 mètres. Aussi, elles assurent une granularité fine dans la collection des trajectoires (cf. Figure 5.1b). Malheureusement la collecte des données GPS a deux inconvénients, sa consommation énergétique et la population restreinte sur lesquels les études sont menées (des conducteurs de taxi [77] ou un groupe de conducteurs de voitures [44]). La collecte de données GPS n'est donc pas adaptée à une étude à l'échelle de la population de métropoles. Par contre, les données issues des opérateurs de réseaux mobiles ne souffrent pas de ces biais et sont devenues récemment une source d'information concernant la mobilité. Les journaux d'appels (Call Data Records ou CDRs) ont été utilisés comme une source valable concernant la mobilité de populations importantes [118, 33, 8, 22].

La géolocalisation de téléphones mobiles basée uniquement sur la station de base sur laquelle celui-ci est connecté fournit une précision assez faible, de l'ordre de plusieurs centaines de mètres en milieu urbain dense à quelques kilomètres en milieu rural [118]. À partir de la *trajectoire de mobilité cellulaire* (une séquence faite d'identifiants de stations de base) et de coordonnées des stations de base, comme

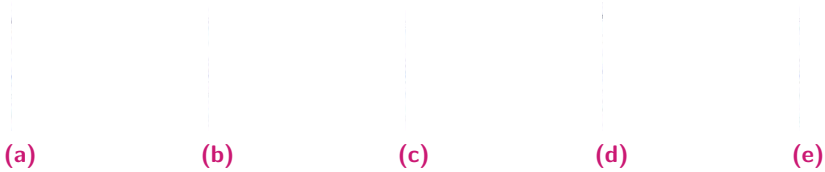


Figure 5.1 : Le trajet d'un utilisateur de l'aéroport CDR au centre de Paris : en Figure 5.1a Le trajet routier consiste en une séquence de routes que l'utilisateur emprunte ; en Figure 5.1b La trajectoire GPS est échantillonnée toutes les minutes ; en Figure 5.1c La trajectoire cellulaire (complète) enregistre chacune des cellules sur lesquelles le client se connecte ; en Figure 5.1d La trajectoire CDR liste les endroits depuis lesquels l'utilisateur a émis ou reçu des appels ; en Figure 5.1e La trajectoire cellulaire éparse, échantillonnée toutes les 15 minutes.

montré en Figure 5.1c, il peut être difficile d'observer la route ou la ligne de métro que l'utilisateur emprunte (cf Figure 5.1a).

Pour construire des *trajectoires de mobilité cellulaire*, les travaux précédents [33, 118] extrayaient généralement les trajectoires des journaux d'appels (Call Detailes Records ou CDR), où le CDR concernant un utilisateur liste les identifiants de cellules depuis lesquelles des appels ont été émis ainsi que l'horodatage de ces appels. Pour comprendre la mobilité humaine, ces travaux ont été limités principalement à l'agrégation sur une longue période des trajectoires des utilisateurs afin de déterminer les lieux fréquemment visités ainsi que l'heure de la visite (par exemple, le lieu dans lequel l'utilisateur ou l'utilisatrice passe habituellement entre 7 et 9h les jours ouvrés). En tant que tels, les techniques proposées ne sont pas applicables pour une estimation précise de trajets sur route ou à travers les réseaux de transports en commun.

De plus, les évènements présents dans les CDR sont déclenchés lorsque le client passe un appel, rendant ainsi les déplacements entre deux appels consécutifs invisibles à l'expérience. Ceci est amplifié par le fait que la période entre deux appels peut être longue (cf Figure 5.1d). Donc, même s'ils ont servi de matériau pour de nombreux travaux, les journaux d'appel ne sont pas une bonne source pour la construction de trajectoires géospatiales. En considérant la rareté des évènements des CDR, nous utilisons dans notre travail une nouvelle technique passive de capture, nous permettant d'extraire efficacement la position de stations de base auxquelles le téléphone portable est connecté. Cette technique analyse le canal de signalisation du réseau de données mobile et en extrait les données signifiantes. Cette technique de capture de la mobilité passe à l'échelle et fournit une plus grande précision temporelle que celle basée sur les CDR.

Les données que nous utilisons pour cette étude nous ont été fournies par les participants à l'expérience, qui ont eux-mêmes demandé à l'opérateur de leur fournir

les données concernant leurs connexions. En considération des *questions concernant la vie privée* [89], l'opérateur localise chaque utilisateur mobile périodiquement (toutes les 15 minutes pour cette étude). Comparé à la trajectoire réelle (Figure 5.1a), le trajet issu des données mobiles (Figure 5.1e) mesure partiellement la mobilité de l'utilisateur et ceci avec des coordonnées très peu précises. L'objectif de notre travail est de projeter les données issues du réseau mobile sur le réseau de transport multimodal afin d'obtenir la séquence des nœuds du réseau à travers lesquels passe l'utilisateur. Par exemple, à partir des cellules en Figure 5.1e et du réseau de transports de l'Île-de-France en Figure 5.2, notre but est de récupérer la séquence des nœuds du trajet réel de la Fig 5.1a.

L'approche commune pour cartographier les trajectoires cellulaires sur le réseau de transport métropolitain (principalement routier) s'opère en deux étapes. Premièrement, la collecte d'un important volume de trajectoires cellulaires et l'établissement manuel de correspondance entre séquences d'intersections - une intersection étant un nœud du graphe associé au croisement entre deux routes. L'étape suivante est d'entraîner un *modèle de mobilité supervisé* (par exemple le Modèle de Markov Caché) en utilisant les séquences d'intersections, afin de construire un modèle probabilistique liant les séquences des deux mondes. Après cet entraînement, pour une nouvelle trajectoire cellulaire, le modèle supervisé prédit, comme un résultat, une séquence d'intersections ayant la probabilité maximale de générer la séquence d'identifiants de stations de base. Toutefois, collecter le volume de trajectoires cellulaires permettant de couvrir l'intégralité du réseau de transports et chacune des stations de bases du réseau mobile est un but difficilement atteignable. C'est pourquoi nous proposons de résoudre le problème de projection de trajectoires en utilisant un *modèle de mobilité non supervisé* ne nécessitant de cataloguer aucune trajectoire.

Étant donnés les exemples cités ci-dessus ainsi que les buts de recherche, les éléments clefs du modèle de mobilité non supervisé sont les suivants :

Graphe multimodal

Pour une séquence donnée d'identifiants de cellules dans un trajet cellulaire, rechercher la séquence d'intersections routières/ferroviaires que l'utilisateur traverse depuis une ***base de données*** *stockant le* ***réseau multimodal de transport***. Le graphe des transports contenant et connectant les nœuds des différents modes de transports (rail, métro, autoroute . . .) est appelé *réseau de transport multimodal* [80]. Dans ce réseau, chaque nœud est soit une jonction de routes soit une gare de transport ferré (station de métro, gare de tramway, RER) et chaque arc est une connexion entre ces intersections (par exemple, le trajet entre une station de métro et un arrêt de bus).

Il est évident qu'il est non trivial d'extraire un trajet précis de l'utilisateur à travers le réseau multimodal en se basant sur une séquence d'identifiants de stations de base.

Une trajectoire cellulaire peut venir de nombreux systèmes de transport situés autour des stations de base correspondantes et distribués dans différentes couches (sous-terrain, extérieur). Pour circonvenir ce problème, il est nécessaire de construire une base de données contenant l'ensemble du graphe intermodal de transport, permettant de récupérer avec précision les intersections aux alentours des stations de base étudiées.

Pour construire ce graphe nous utilisons les données *open data* fournies par OpenStreetMap (OSM) et par l'Institut Géographique National (IGN). Nous en extrayons les données concernant l'Île-de-France, cette région est caractérisée par une grande diversité dans le type de transports (tram, RER, métro, train, bus) qui ont chacun des caractéristiques spécifiques. Donc, construire un graphe de transport multimodal dans le but d'étudier la mobilité des individus requiert une bonne compréhension de la complexité du graphe multimodal. Le graphe se base, dans ce travail, sur le concept de liens 'inter-couche' qui connectent deux nœuds permettant à l'utilisateur de changer de mode de transport.

Projection

Pour un trajet cellulaire observé, trouver la ***séquence d'intersections la plus probable*** *au sein du graphe de transport multimodal*. Il est difficile de chercher la séquence d'intersections la plus probable à partir d'un ensemble d'intersections pour la raison suivante :

Alors que le modèle classique de mobilité basé sur le Modèle de Markov Caché (MMC) supervisé exploite les statistiques des trajectoires cellulaires labellisées (c'est-à-dire la probabilité d'émission / la probabilité de transition) et qu'il est habituellement utilisé pour estimer la probabilité, nous proposons un MMC non supervisé qui ne nécessite pas de traitement humain ayant pour but de labéliser les données. Nous proposons une méthode calculant la probabilité en utilisant les *propriétés topologiques du réseau de transport*. Ainsi, les paramètres du MMC sont automatiquement issus d'une connaissance a priori des propriétés du graphe.

En résumé les principales contributions de ce travail sont :

— d'étudier la projection des trajets cellulaires sur le graphe multimodal de transport, afin d'obtenir la mobilité précise de l'utilisateur. À notre connaissance, il s'agit du premier travail traitant ce problème. En particulier, plutôt que de projeter les trajectoires cellulaires en utilisant un algorithme supervisé basé sur les données de mobilité labélisées, nous proposons un algorithme de projection non supervisé basé sur la topologie du réseau de transport, permettant d'éliminer la fastidieuse tâche de labéliser.

— de proposer un algorithme, nommé *CT-Mapper*, non supervisé de construction de trajectoire et qui projette les données de mobilité issues du réseau cellulaire sur le graphe de transport multimodal. La base de données du graphe de transport multimodal a été construite à partir de différentes ressources géospatiales. L'algorithme de projection est modélisé par un MMC où les observations correspondent aux trajectoires cellulaires des utilisateurs et les états cachés sont associées aux nœuds du graphe multicouches. La probabilité de transition et le score d'émission ont été modélisés à partir des propriétés topologiques du réseau de transport et de la distribution spatiale des stations de base. L'algorithme de Viterbi permet de réduire la complexité lors de la recherche de la meilleure correspondance et nous permet ainsi de déployer notre algorithme à grande échelle sur d'importants jeux de données.

— de collecter avec l'aide d'un opérateur français les trajectoires réelles d'un groupe d'utilisateurs en Île-de-France, puis d'évaluer notre algorithme à partir de ces données. À travers l'évaluation approfondie avec des trajectoires cellulaires couvrant plus de 2500 nœuds d'intersection, 3 couches physiques et 1000 stations de métro, nous montrons que notre algorithme a, sur la zone considérée, de bonnes performances et renvoie des données avec une précision élevée, malgré la faible densité des données cellulaires. Notre algorithme atteint une précision supérieure de 20% à celle obtenue via une approche classique qui exploite pour estimer les paramètres du Modèle de Markov Caché non supervisé la complexité et la topologie du réseau multicouche sans tenir compte des propriétés des arcs du graphe.

Dans ce chapitre nous donnerons un aperçu du système proposé, puis présenterons les détails de notre estimation non supervisée des paramètres de notre MMC et expliquerons comment les deux principales distributions de probabilité utilisées pour la projection ont été dérivées. Finalement nous évaluerons les performances de notre algorithme.

5.2 Présentation du système CT-Mapper

Nous formulerons ici la problématique de recherche de CT-Mapper et présenterons les données collectées que nous utiliserons pour la projection. Nous analyserons ensuite la complexité de calcul liée au problème de projection sur les différentes données et conclurons en présentant la structure de CT-Mapper.

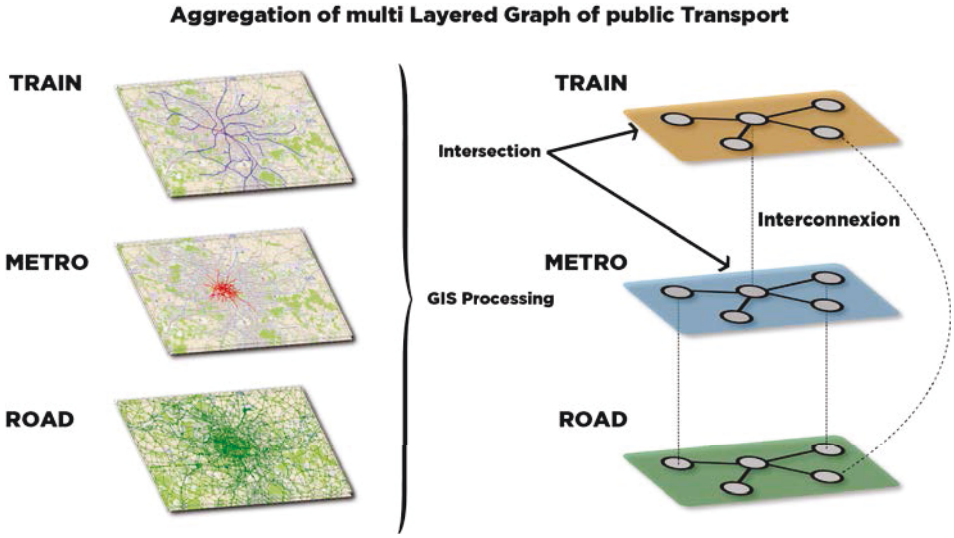


Figure 5.2 : Représentation multicouches de différents réseaux de transport.

5.2.1 Exposé du problème

Nous formulerons ci-dessous le problème en définissant les différents concepts clefs de notre approche.

Définition 1. ***Graphe multicouches de transport***

Un tel graphe est défini comme $\mathbf{G} = (V, E, L, \Psi)$ où V, E représentent respectivement les nœuds et les arcs et L est l'ensemble des couches possibles. Dans notre étude, nous nous concentrerons sur 3 couches : routes, train et métro.

— *Fonction* Ψ indique la couche de chacun des nœuds $\Psi : V \rightarrow L$ in **G**.

— *La couche de transport* $G^l = (V^l, E^l)$ est un sous-ensemble de **G**, où $V^l = \{v | v \in V, \Psi(v) = l\}$ et $E^l = \{< v_i, v_j > \in E, \Psi(v_i) = \Psi(v_j) = l\}$. Chaque noeud v_i est caractérisé par ses coordonnées géographiques ($v_i =< lat, lon >_i$)

— Un ensemble d'*arcs inter-couche* $E^{cl} \subset E$ définit les arcs liant des nœuds n'appartenant pas à la même couche : $E^{cl} = \{< v_i, v_j > \in \mathbf{E} | \Psi(v_i) \neq \Psi(v_j)\}$

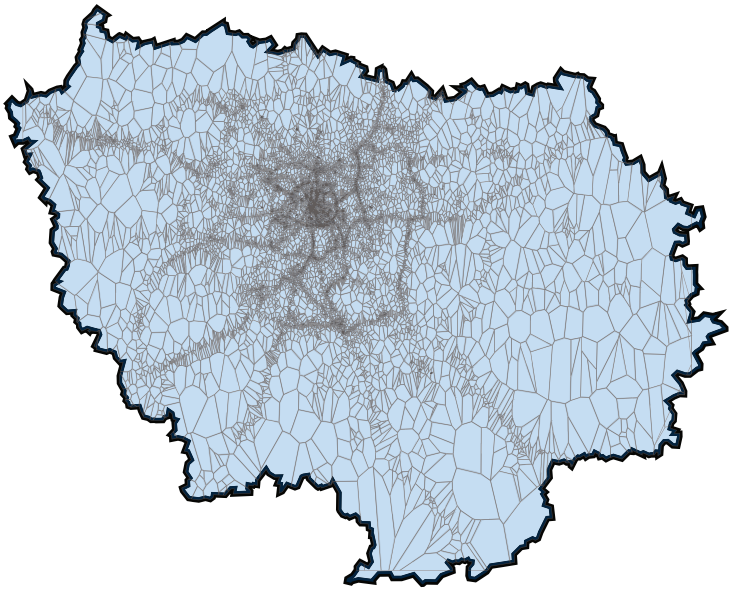


Figure 5.3 : Tessallation Voronoi des cellules d' Ile-de-France

— Le graphe de transport multicouches est caractérisé par sa *matrice d'adjacence* $W_{ij} \in \mathbb{R}^{|V|\times|V|}$ *où* $|V|$ *désigne le cardinal de* V. La Figure 5.2 illustre comment les différentes couches de transport ont été agrégées pour construire un graphe de transport multimodal.

Définition 2. ***Réseau cellulaire***

Dans ce travail nous considérons le réseau cellulaire comme un ensemble de stations de base $C = \{c_0, c_1, \dots c_P\}$, où chaque station de base $c_p =< lat, lon, r^{max} >_p$ est définie par ses coordonnées géographiques (latitude et longitude) et par le rayon maximal (r^{max}) de la cellule de Voronoi dans laquelle se trouve la station de base. Le graphe de Voronoi est construit à partir de l'ensemble C. Il est également à noter que la position de chaque cellule ne correspond à aucune intersection dans le réseau de transport, c'est à dire : $\forall v_i \in V$, $\forall c_p \in C$, nous avons $< lat, lon >_p \neq < lat, lon >_i$

Définition 3. ***Trajet cellulaire épars***

Nous définissons un *trajet cellulaire épars* tel qu'une séquence horodatée attribuée à un utilisateur et constituée de coordonnées $O = o_0 \rightarrow o_1 \cdots \rightarrow o_M$ où chaque coordonnée horodatée $o_t =< c(t) >$ se rapporte à la station de base à laquelle l'utilisateur était connecté à l'instant t.

Définition 4. ***Problème de projections de trajectoire***

Pour un graphe de transports donné **G**, une station de base C et un trajet cellulaire épars O, notre problème est de *trouver une séquence d'intersections* $v_0 \rightarrow v_1 \cdots \rightarrow v_q$ *par laquelle l'utilisateur passe réellement dans le réseau de transport*.

5.2.2 Collecte et jeux de données

Nous utilisons dans cette étude trois types de données : les données du réseau de transport multimodal, les trajets cellulaires épars et les données issues du module GPS du *smartphone* des utilisateurs. Les données du réseau de transport permettent de construire le graphe multicouches et le modèle de mobilité sur lequel se basera l'algorithme de projection. Les trajectoires cellulaires sont utilisées pour les tests alors que les trajectoires GPS sont utilisées comme une vérité terrain et non comme un jeu de données d'apprentissage pour les paramètres du MMC.

Données des trajets cellulaires épars

Nous utilisons dans ce travail un nouveau type de données que nous nommerons Trajet Cellulaires Épars. Un ensemble de techniques, décrit dans le Chapitre 3, est utilisé pour capturer les messages du GPRS Tunneling Protocol (GTP) depuis le réseau de données cellulaire. L'inspection du flux de signalisation de ce protocole (control plane ou GTP-C) nous permet d'obtenir les informations de géolocalisation des utilisateurs à une fréquence plus élevée qu'on pourrait le faire en exploitant des CDR. Le GTP est le protocole de gestion de tunnels utilisé pour transporter des données au travers du réseau mobile (de la 2G à la 4G). Lorsqu'un smartphone active sa connexion à internet (de nos jours, dès qu'il est allumé), un message GTP traverse le réseau demandant l'accès pour le client. Ce message contient, entre autres, l'identité du téléphone et l'identifiant de la cellule à laquelle le client est connecté. Une fois la session établie, des messages de mise à jour sont envoyés, informant du changement de cellule, du passage en (ou de la sortie de) veille radio du téléphone et du changement de technologie radio. Finalement, lorsque le mobile perd le signal ou lorsqu'il est éteint, le réseau met fin à sa connexion en envoyant un message de clôture. Avec les équipements récents, de nombreuses applications y sont installées et font un usage régulier du réseau de données (au travers des notifications en *push*, des la consultation du courrier électronique) ; il est attendu que le tunnel GTP reste actif et que s'il passe en veille radio, il en sort suffisamment régulièrement pour que nous soyons capables de suivre de manière fréquente le changement de cellule des utilisateurs. Alors que les données GTP sont différentes des CDR, elles souffrent des

mêmes problèmes : coordonnées dupliquées et effet de ping-pong. Elles ont besoin d'être nettoyées afin d'éliminer les bruits des échantillons utiles [65].

Données GPS des trajets

Afin d'évaluer la précision de l'algorithme que nous proposons, des données issues du GPS nous serviront de données de terrain. Un groupe de participants a dû installer l'application pour smartphones "Moves" [90] afin d'enregistrer leurs coordonnées GPS. Les coordonnées GPS fournies par "Moves" ont été analysées pour obtenir le trajet réel des participants.

5.2.3 Complexité calculatoire concernant le problème de projection dans les jeux de données collectés

	Nombre de		**Moy.**		
	Nœud	Arc	Degré	Long. d'Arc	Référence
Métro	303	356	2.35	0.757	OSM
Train	241	244	2.025	3.07	OSM
Tram	146	140	1.918	0.71	OSM
Route	14798	22276	3.01	1.34	IGN

Table 5.1 : Les différents réseaux de transport et leurs propriétés.

Le réseau de transport sous-jacent utilisé pour cette étude est le réseau multimodal de transport d'Île-de-France. Il est modélisé par des couches distinctes du graphe, correspondant chacune à un mode de transport différent, les interconnectant en un multiplex **G**. Pour construire ce multiplex, diverses sources géospatiales ont été utilisées. Le réseau routier, provenant de l'Institut Géographique National (IGN) [63] et le réseau ferré (train et métro), extrait d'OpenStreetMap (OSM) [98], sont agrégés. Chaque nœud **G** peut être un croisement routier, une gare ferroviaire ou encore une station de métro. Un élément clef du réseau multimodal de transport proposé est de modéliser les transitions entre chaque mode de transport pendant un voyage. Les transitions intermodales modélisées sont assurées par l'ajout de nœuds de transition *inter-couche* au sein du graphe.

Bien qu'une telle représentation multicouches du réseau de transport nous permette de modéliser et définir des trajets en utilisant différents modes de transport, il augmente également la complexité du réseau obtenu. Le tableau 5.1 illustre les diffé-

rences topologiques entre chaque couche du graphe multicouches **G**. Par exemple, la longueur moyenne entre deux intersections est assez hétérogène au travers des différentes couches de transport. Pour évaluer quantitativement la complexité du réseau, nous utilisons une mesure d'entropie pour caractériser la facilité ou la difficulté de naviguer dans ce réseau en utilisant "the search information" [119, 110].

L'équation 5.1 définit la probabilité pour une marche aléatoire, débutant au nœud s de degré k_s, d'atteindre le nœud t. Par conséquent, dans l'équation 5.2, nous définissons l'entropie de recherche dans le graphe comme la moyenne sur tous les couples de nœuds possibles (s,t) dans le graphe G, du log en base 2 de l'inverse de la somme des probabilité des tous les plus courts chemins allant du nœud s au nœud t dans G. On désigne par $\{SP_{st}\}$, l'ensemble des plus courts chemins allant de s à t. Il en résulte, par le calcul de cette entropie moyenne de recherche sur tous les plus courts chemins possible de G, que nous pouvons exprimer la complexité relative (S_{avg}) de recherche d'un chemin donné dans un graphe donné G.

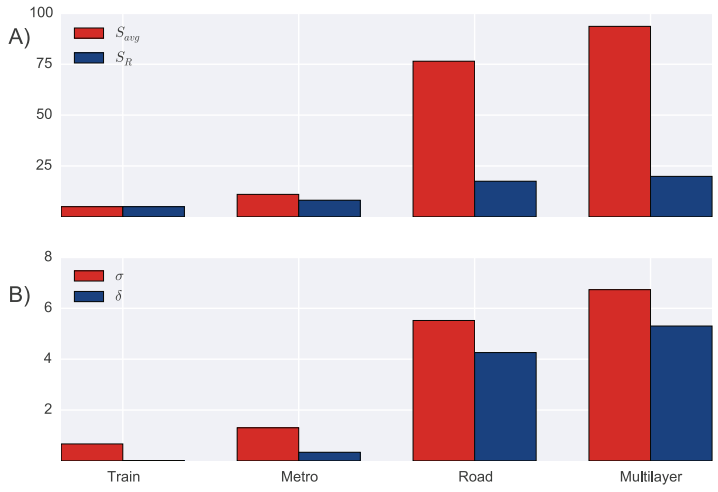


Figure 5.4 : Entropie du graphe : (A) valeur absolue de l'entropie moyenne du graphe où S_{avg} est l'entropie du graphe réel et S_R est l'entropie du graphe aléatoire ayant des caractéristiques similaires et (B) est l'entropie moyenne du graphe des chemins dans les sous-graphes métro, train route.

$$P[SP_{st}] = \frac{1}{k_s} \prod_{j \in SP_{st}} \frac{1}{k_j - 1} \tag{5.1}$$

$$S_{avg} = \frac{1}{N(N-1)} \sum_{s=1}^{N} \sum_{t=1}^{N} - \log_2 \sum_{\{SP_{st}\}} P[SP_{st}] \tag{5.2}$$

En Figure 5.4a, nous représentons l'entropie moyenne de chaque couche du graphe mulitmodal du réseau de transport d'Île-de-France, ainsi que l'entropie moyenne du réseau multicouches interconnecté. Nous observons que l'entropie moyenne est plus élevée dans le réseau de transport multicouches que dans toutes les couches prises séparément. La Figure 5.4b montre également l'entropie moyenne $sigma$ de recherche d'un chemin relativement à la taille du graphe. Il est montré clairement que la complexité du graphe multicouches est plus grande que chacune des couches prises séparément et ceci peu importe sa taille. Nous définissons $\sigma = S/\log_2(N)$ comme étant l'entropie moyenne de recherche d'un chemin dans le graphe relativement à sa taille et $\delta = (S_{avg} - S_R)/\log_2(N)$ pour comparer un graphe avec son équivalent aléatoire en termes de degré de nœud, quelle que soit la taille du réseau.

En conclusion, la complexité de recherche du bon chemin au sein du graphe multicouches de transport est plus élevée qu'elle ne le serait dans une seule couche du graphe prise en particulier. Ceci a deux causes : (i) lorsque différentes couches sont combinées pour former un graphe multicouches, le nombre de chemins dégénérés (dont la longueur est identique) augmente, de même que la complexité générale et (ii) lorsque nous construisons le réseau de transport multicouches, nous ajoutons de nombreuses interconnexions entre les couches, augmentant ainsi le degré des nœuds se trouvant aux carrefours des couches. Il est également nécessaire de noter la différence importante de complexité entre différentes couches (train, métro, route). L'agrégation de couches accroît le nombre de chemins dégénérés.

Ces effets combinés accroissent la complexité de la recherche d'un parcours donné dans un réseau de transport multicouches et par conséquent augmentent la difficulté de trouver la bonne projection des trajectoires cellulaires éparses sur le graphe. De plus, notre algorithme devra faire face au défi de projeter les trajets sur des couches d'une complexité élevée.

5.2.4 Structure en conception globale

Étant donné le réseau de transport multimodal **G** et le réseau cellulaire C, nous définissons un algorithme produisant le parcours, ou la séquence d'intersections, le ou la plus probable associé(e) au trajet cellulaire épars O. Afin d'inférer la séquence d'intersections la plus fidèle à partir d'un trajet cellulaire épars défini, nous proposons un algorithme de projection non supervisé en deux étapes : la **première**, l'algorithme recherche une séquence d'intersections, que nous appellerons la *séquence squelette*, où deux intersections consécutives ne sont pas nécessairement adjacentes (comme en Figure 5.5c). Dans cet objectif nous avons développé un algorithme basé sur le Modèle de Markov Caché (MMC) s'accommodant de la rareté des obser-

Figure 5.5 : Une illustration de différentes phases de l'algorithme de projection. La ligne bleue dans la Figure 5.5a est la trajectoire GPS réelle d'un utilisateur donné et sa séquence de 5 cellules échantillonnées toutes les 15 minutes. En Figure 5.5b, nous représentons la trajectoire cellulaire (les positions des cellules sur lesquelles le mobile a été détecté pendant son déplacement). En Figures 5.5c, 5.5d et 5.5e, nous montrons les étapes de l'algorithme dont le résultat final est présenté en Figure 5.5e.

vations (une observation toutes les 15 minutes). Les états cachés dans le MMC sont les intersections routières et ferroviaires du réseau de transport et sont représentés par des nœuds dans le graphe multimodal. La probabilité de transition dans notre modèle prend en compte la rareté des observations en permettant des transitions entre des nœuds non adjacents comme expliqué en 5.3. Pour chaque observation, un ensemble d'états cachés sont sélectionnés parmi les états candidats dans le but de réduire la complexité de recherche dans le graphe. Pour une séquence donnée d'observation cellulaire éparse, notre MMC produit la séquence la plus probable au sein du réseau multiplex (pour seulement 3 ou 4 points observés).

Ensuite, dans une **seconde phase** (décrite en Figure 5.5d), l'algorithme parcourt le squelette et produit une séquence d'intersections adjacentes en complétant les séquences (cf. Figure 5.5e). Il est à noter que la séquence squelette recherchée dans la première étape est de même longueur que le trajet cellulaire épars O, alors que la séquence créée lors de la seconde étape est plus longue que O. Il est évident que pendant une période d'observation de 15 minutes, un utilisateur traversera plus d'une intersection (lors d'une correspondance dans le métro, passer d'une ligne à l'autre prend environ 3 minutes).

Recherche de la séquence squelette. Pour une trajectoire cellulaire éparse donnée $o_0 \rightarrow o_1, ... \rightarrow o_M$, cette étape retourne la séquence squelette d'intersections $v_0 \rightarrow v_1, ... \rightarrow v_M$. Nous modélisons le problème de recherche par un modèle de Markov caché, dans lequel les intersections sont les états cachés. Nous disposons :

— de M observations $O_{t_0}, \dots O_{t_M}$;

— de la densité de probabilité d'une observation O_{t_k} sachant que nous nous trouvons à l'intersection v (notée $P(O_{t_k}|v)$) et telle que définie au paragraphe 5.3.2 ;

— et finalement de la densité de probabilité ou probabilité de transition de l'état v à l'état w notée $Tr(v,w)$ et telle que définie au paragraphe 5.3.1

Sous ces conditions l'aglorithme de Viterbi se décline comme suit :

Phase d'initialisation :

— Recherche de l'ensemble $V_{t_0} = \{v \in V; P(O_{t_0}|v) \neq 0\}$ (en effet, il est inutile de considérer les intersections telle que $P(O_{t_0}|v) = 0$)

— Initialisation de la probabilité associée aux intersections candidates :

$$Pr_{t_0}(v) = P(Ot_0|v) \ (\forall v \in Vt0) \tag{5.3}$$

Phase récursive : Recherche des intersections candidates :

$$\text{pour } t_k, 1 \leq k \leq M, \text{ on définit } V_{t_k} = \{\forall v \in V; P(O_{t_k}|v) \neq 0\} \tag{5.4}$$

Recherche du noeud père : **pour chaque état** w de V_{t_k}, on détermine le nœud père de w dans la séquence en recherchant l'intersection v de $V_{t_{k-1}}$ qui réalise le maximum de :

$$Pr_{t_{k-1}}(v) \times Tr(v,w) \tag{5.5}$$

Notons $v_{t_{k-1}}$ cette intersection où le maximum est réalisé, en d'autres termes

$$v_{t_{k-1}}(w) = \arg \max_{v \in V_{t_{k-1}}} (Pr_{t_{k-1}}(v) \times Tr(v,w)) \tag{5.6}$$

Nous enregistrons ce résultat dans un tableau (dénommé tableau des nœuds pères ou Par) à deux entrées dont les lignes sont les temps t_k et les colonnes sont indicées par des éléments de v faisant partie de V_{t_k} pour $1 \leq k \leq M$.

$$Par(t_k, w) = v_{t_{k-1}}(w) \tag{5.7}$$

Mise à jour de la probabilité qu'un utilisateur soit à l'intersection w au temps t_k, sachant les observations $O_{t_0} \rightarrow \ldots O_{t_k}$ par l'équation récursive 5.3 :

$$\begin{aligned} Pr_{t_k}(w) &= P(O_{t_k}|w) \times \max_{v \in V_{t_{k-1}}} (Pr_{t_{k-1}}(v) \times Tr(v,w)) \\ &= P(O_{t_k}|w) \times Pr_{t_{k-1}}(Par(t_k,w)) \times Tr(Par(t_k,w),w) \end{aligned} \tag{5.8}$$

Phase finale Une fois l'étape récursive au temps t_M réalisée, nous disposons de $Pr_{t_M}(w)$ pour tout $w \in V_{t_M}$. Il reste à choisir parmi ces intersections w celle qui a la plus haute valeur de $Pr_{t_M}(w)$, en d'autres termes on note :

$$\overset{*}{v}_{t_M} = \arg \max_{v \in V_{t_M}} Pr_{t_M}(v) \tag{5.9}$$

On obtient la séquence d'intersections la plus probable en parcourant le tableau des noeuds pères par une itération en marche arrière selon l'équation :

$$\overset{*}{v}_{t_{k-1}} = Par(t_k, \overset{*}{v}_{t_k}) \text{ pour } k = M, \ldots, 1 \tag{5.10}$$

cette séquence d'intersections la plus probable $\overset{*}{v}_0 \rightarrow \overset{*}{v}_1, ... \rightarrow \overset{*}{v}_M$ produit le parcours le plus probable pour le trajet cellulaire épars $o_0 \rightarrow o_1, ... \rightarrow o_M$. Comme il n'est pas certain que $\overset{*}{v}_i, \overset{*}{v}_{i+1}$ soient des intersections adjacentes dans le multiplex **G**, la séquence $\overset{*}{v}_0 \rightarrow \overset{*}{v}_1, ... \rightarrow \overset{*}{v}_M$ var servir d'entrée à la prochaine étape pour récupérer la séquence d'intersections adjacentes pour un trajet cellulaire épars.

Complétion de la séquence d'intersections adjacentes. Pour une séquence squelette donnée $\overset{*}{v}_0 \rightarrow \overset{*}{v}_1 ... \rightarrow \overset{*}{v}_M$ et pour chaque couple consécutif d'intersections $\overset{*}{v}_i, \overset{*}{v}_{i+1}$ qui ne sont pas adjacentes dans le multiplex **G**, l'algorithme recherche la séquence optimale $v_{i_1} \rightarrow v_{i_2} ... \rightarrow v_{i_k}$ et insère les séquences nouvellement trouvées entre les deux intersections $\overset{*}{v}_i, \overset{*}{v}_{i+1}$:

$$\overset{*}{v}_i \rightarrow \underbrace{v_{i_1} \rightarrow v_{i_2} ... \rightarrow v_{i_k}}_{\Uparrow \text{ Parcours reconstitué}} \rightarrow \overset{*}{v}_{i+1} \tag{5.11}$$

établissant ainsi la séquence complète d'intersections adjacentes. Il est à noter que tous les nœuds consécutifs obtenus dans la nouvelle sous-séquence sont adjacents dans le multiplex **G**. Dans la partie suivante, nous introduirons le calcul des probabilités utilisé dans notre structure.

5.3 Algorithmes principaux

Nous avons précédemment décrit l'algorithme général de projection sur le réseau de transport multimodal, les deux principales distributions de probabilité utilisées dans l'algorithme de projection sont les transitions MMC et les scores estimés de manière non supervisée. Cette partie explique en détail comment ces deux scores sont définis et évalués/estimés.

5.3.1 Probabilité de transition

La probabilité de transition $Tr(v_i, v_j)$ dans notre algorithme de projection définit la probabilité qu'un individu se déplace d'un état caché v_i à $t-1$ vers un état caché v_j à t. La probabilité de transition est déduite du réseau sous-jacent, à savoir le réseau multicouches de transport dont chaque couche de transport possède ses caractéristiques propres. Le tableau 5.1 nous montre quelques propriétés topologiques du graphe, telles que la distribution du degré des nœuds et la distribution de longueur physique des arcs dans différentes couches du réseau multimodal de transport.

Il est important de souligner que reposer sur les propriétés topologiques des couches du réseau sans tenir compte de leurs différences mène à un algorithme de projection biaisé, dans lequel les observations tendent à être liées à une couche de transport spécifique. De plus, prendre en compte la rareté des observations cellulaires favorise la possibilité de transitions entre des intersections non adjacentes. Nous proposons une probabilité de transition de passer d'une intersection v_i à une intersection v_j dépendant de deux facteurs :

1. Le type d'arc et la vitesse sur chaque arc : chaque arc physique du graphe multicouches **G** appartient à une couche. De plus, seule la couche route contient différents types d'arcs (autoroute, route nationale, etc . . .). Nous définissons une matrice W dans laquelle chaque élément représente un poids entre deux nœuds s'il existe une interconnexion entre eux. Le poids de chaque lien est défini comme l'inverse de la vitesse moyenne de l'arc correspondant. Le tableau 5.2 représente le poids en fonction de la vitesse moyenne sur les arcs du graphe **G**.

$$W_{ij} = \begin{cases} w_{ij} & \text{si } v_i, v_j \text{ sont adjacents dans } \mathbf{G} \\ 0 & \text{sinon.} \end{cases} \tag{5.12}$$

$\mathbf{w_{ij}}$	**Condition**
$1/10$	$\Psi(v_i) \neq \Psi(v_j)$
$1/100$	$\Psi(v_i) = \Psi(v_j) =$ train
$1/80$	$\Psi(v_i) = \Psi(v_j) =$ metro
$1/90$	$\Psi(v_i) = \Psi(v_j) =$ route (autoroute)
$1/60$	$\Psi(v_i) = \Psi(v_j) =$ route (nationale)
$1/40$	$\Psi(v_i) = \Psi(v_j) =$ route (départementale)
$1/30$	$\Psi(v_i) = \Psi(v_j) =$ route (locale)

Table 5.2 : Classification et pondération des arcs du graphe de transport mutlicouches **G**.

2. Longueur d'arc : tenir compte de la longueur d'arc dans la probabilité de transition implique indirectement une probabilité plus grande pour les nœuds les plus proches plus que pour les plus éloignés.

   La probabilité de transition entres deux intersections v_i et v_j est définie comme l'inverse du coût du plus court chemin $SP_{v_iv_j}$, entre v_i et v_j :

$$Tr(v_i, v_j) \propto \left(\sum_{\forall\ (mn)\ \in\ SPv_iv_j} w_{mn} \times d(v_m, v_n) \right)^{-1} \tag{5.13}$$

   où (mn) est un arc entre v_m et v_n appartenant à $SP_{v_iv_j}$, le plus court chemin entre v_i et v_j dans le graphe **G**. Le coût du plus court chemin de $SP_{v_iv_j}$ est la somme des distances de chacun des arcs (mn) appartenant à $SP_{v_iv_j}$, pondéré par w_{mn}. Où $d(v_m, v_n)$ désigne la distance euclidienne entre les nœuds v_m et v_n.

Dans de précédentes études, la quantification des probabilités de transition était basée sur les propriétés topologiques du réseau sous-jacent qui était principalement constitué par le réseau routier. Dans [94, 127], le réseau de transport est représenté comme un ensemble de segments routiers dont les transitions ont lieu entre segments adjacents. Les auteurs de [127, 59] considèrent équiprobables les transitions entre nœuds d'un même segment routier ou entre des nœuds entre des segments routiers adjacents à une intersection. La probabilité de transition dans [126] est basée sur la distance Manhattan entre les cellules des grilles d'un réseau routier. L'objectif de notre proposition de modèle de probabilité de transition est de minimiser le biais de l'algorithme de projection pour les couches ayant des propriétés topologiques différentes.

5.3.2 Probabilité d'émission

Avec le Modèle de Markov Caché, à chaque étape t, il existe une observation o_t qui est identifiée à la cellule sous-jacente $c_t =< lon, lat, r_t^{max} >_t$. Le score d'émission est le reflet de la notion qu'il est plus probable qu'un point particulier observé soit observé à une intersection voisine plutôt qu'à une intersection éloignée [127]. Dans les études dans lesquelles les données GPS sont utilisées en tant qu'observations [127, 94, 46], le score de la probabilité d'émission est modélisé par une distribution normale qui est une fonction de la distance euclidienne entre le point observé et l'état caché, tenant compte de la déviation estimée des erreurs des capteurs.

Dans ce travail, les coordonnées géographiques des stations de base nous servent d'observation. Étant donné qu'il n'y a pas de capteurs pour estimer l'erreur dans le réseau cellulaire, nous avons construit le diagramme de Voronoï des stations de base du réseau dans la zone géographique considérée. Dans le diagramme de Voronoï

(a) Distribution de la durée des trajets. **(b)** Distribution de la distance des trajets.

Figure 5.6 : Distribution de la durée et de la distance des trajets.

chaque cellule C_i est caractérisée par un rayon r_i qui est la distance maximale entre la station de base et les sommets de la cellule. Notre score d'émission est défini tel que décroissant en fonction de la distance entre la station de base et le nœud caché (ou intersection) :

$$Pr(o_t|v_j) \propto \begin{cases} 1.0 & \text{si : } d_{tj} \leq r_t^{max} \\ \left(\frac{r_t^{max}}{d_{tj}}\right)^{\beta} & \text{si : } r_t^{max} \leq d_{tj} \leq \tau \\ 0 & \text{sinon.} \end{cases} \tag{5.14}$$

où $d_{tj} = d(o_t, v_j)$ est la distance euclidienne entre o_t et l'intersection v_j, et τ est le seuil correspondant à la distance maximale de réception pour la station de base. τ impose la contrainte que seules les intersections présentes dans le rayon τ de la station de base sont des états potentiels (nœuds).

5.4 Évaluation

5.4.1 Jeu de données pour l'évaluation

Afin d'évaluer l'algorithme proposé, des données GPS sont utilisées comme vérité terrain. Nous recueillons les trajets cellulaires et les coordonnées GPS correspondantes pour dix participants volontaires pendant un mois (août à septembre 2014). Les données GPS proviennent de l'application "Moves" [90] qui a été installée sur les smartphones des participants. Les données collectées sont la géolocalisation du

Figure 5.7 : Distribution de la distance entre cellules.

téléphone pendant les déplacements ainsi que son activité classée en quatre catégories : 'marche', 'course', 'vélo', 'transport'. Fondées sur ce jeu de données, un ensemble de taches de prétraitements ont été réalisées dans le but d'extraire les trajectoires projetées sur le réseau de transport.

De plus, les trajets dont la longueur est plus courte que 5 kilomètres ont été éliminés de notre base de données. Étant donnée la basse fréquence d'échantillonnage des positions cellulaires de l'expérience (une mesure toutes les 15 minutes), il n'est pas réaliste de chercher à reconstruire un mouvement sur une période plus courte que la période d'échantillonnage. L'effet de ce filtre sur la distribution du jeu de données peut être observé en Figure 5.6a et Figure 5.6b.

La précision spatiale nécessaire à faire la distinction entre une mobilité réelle et du bruit dépend de la distance entre deux stationnements. Afin de filtrer les mouvements sans importance, nous supprimons tous les trajets sous un seuil x_{th} tel que $P_r(X < x_{th}) = q$ où $P_r(X)$ est la distribution des distances entre les stations de base voisines. Pour $q = 0.97$, comme la figure 5.7 le montre, ces distances entre voisines sont inférieures à 5 kilomètres.

En conclusion, nous avons construit un jeu de données de 80 trajets cellulaires (constitués chacun par une séquence de stations de base) avec les coordonnées GPS correspondantes projetées sur le graphe multicouches **G**. Le réseau multicouches de transport contient environ $16\,000$ nœuds et $26\,000$ arcs. Les trajets d'utilisateurs couvrent une distance totale de 2200 kilomètres. Le nombre moyen de points observés dans chaque trajet cellulaire est de 5.55 stations de base et la longueur moyenne d'un trajet est de 26.5 kilomètres.

5.4.2 Évaluation des résultats et comparaisons

Figure 5.8 : Évaluation du résultat.

Efficacité de l'algorithme de projection

Afin d'évaluer notre algorithme, le jeu de données qualifié présenté ci-dessus a été utilisé pour des tests et des évaluations. Nous utilisons *CT-Mapper* pour projeter les trajets cellulaires et pour comparer les résultats avec la vérité terrain. Il est important de noter que cette comparaison est faite entre deux séquences n'ayant pas nécessairement la même longueur. Nous utilisons la *distance d'édition*, ou *distance de Levenshtein* entre deux séquences, consistant en le nombre minimal d'éditions (insertion, suppression ou substitution) requises pour transformer un trajet en un autre. Nous évaluons les deux étapes de l'algorithme en calculant le score de similarité basé sur l'édition pour les deux squelettes et la séquence projetée complète. Pour avoir un aperçu complet, nous calculons également le rappel/recall moyen et la précision des résultats des trajets du jeu de données.
Ici, le rappel est la fraction des nœuds corrects retrouvés pas l'algorithme pour toutes les trajectoires. De même, la précision est la fraction des nœuds réels de la trajectoire retrouvés.

Du fait du grand bruit spatial des observations cellulaires, nous avons utilisé une erreur fixe, RMSE (Root Mean Square Error) pour identifier les points corrects parmi les coordonnées estimées.

Par exemple, une erreur de 0.1 kilomètre indique que pour chaque nœud dans la séquence résultante, le nœud est considéré comme bon s'il est dans un rayon de 0.1 kilomètre autour de sa position réelle. Nous calculons les quatre résultats de précision mentionnés (précision, rappel, squelette et séquence complète score de similarité) pour une plage fixée de RMSE permises sur les résultats de projection obtenus. Les scores de similarité sont le complément des scores des distances d'édition.

La Figure 5.8 représente le résultat de cette évaluation. Comme la Figure 5.8 le montre, avec une RMSE permise à 200 mètres, plus de 50 % des squelettes et des trajectoires complètes peuvent être construits. Ceci est remarquable au vu de la rareté des informations présentes dans la position des stations de base vis-à-vis du trajet réel de l'utilisateur (une moyenne de 5.5 observations par trajet dans le jeu de données alors que le trajet moyen est de 26.5 km). Il est important de mentionner que l'échantillonnage des données cellulaires est fait toutes les 15 minutes, ainsi avec une fréquence plus élevée nous espérons avoir de meilleures performances avec *CT-Mapper*. Le score moyen de similarité pour une RMSE de 1 kilomètre atteint 80 %. De plus, *CT-Mapper* atteint un rappel et une précision aux alentours de 80 % lorsqu une RMSE de 1 km est permise. En plus de la métrique mentionnée ci-dessus, nous évaluons la distance d'édition comme le coût de chaque édition requise. La moyenne des distances d'édition pour tous les trajets du jeu de données est de 0.79 kilomètre.

Comparaison avec la référence

Pour évaluer notre modèle de probabilité de transition basé sur les propriétés du réseau de transport présenté en Équation 5.13, nous considérons un modèle de référence associé à l'hypothèse naïve consistant à appliquer des probabilités égales à toutes les transitions sortantes de chaque nœud (incluant une transition vers le même nœud). Avec un tel modèle, la probabilité de transition entre deux nœuds v_i et v_j est représentés telle que :

$$Tr(v_i, v_j) = \left(k_i * \prod_{n \in Q} k_n \right)^{-1} \tag{5.15}$$

où $Q = SP_{v_i v_j} - \{v_i, v_j\}$ et k_i est le degré de v_i. Cette hypothèse naïve tient compte de tous les arcs du réseau multicouches mis sur un pied d'égalité, indépendamment des propriétés des couches de transport.

Figure 5.9 : En haut à gauche : Precision, en haut à droite : Rappel/Recall, en bas à gauche : le score de similarité basé sur l'édition, en bas à droite : le score similarité basé sur le squelette .

Utilisant ce modèle de probabilité de transition, nous construisons un MMC de la même manière que *CT-Mapper* a été développé. Nous utilisons ce modèle comme un algorithme de référence et l'avons lancé sur le jeu de données et avons comparé les résultats avec *CT-Mapper*. Nous avons calculé les quatre mesures de performance avec l'algorithme de référence. La figure 5.9 compare les performances de ces deux modèles. Comme les figures le montrent, il y a 20 % d'amélioration de performances dans le rappel/recall en utilisant le modèle de probabilité de transition proposé. Aussi la distance d'édition moyenne de l'algorithme de référence est de 1.04 kilomètre, ce qui prouve que les performances de *CT-Mapper* sont significativement meilleures comparativement à l'algorithme de référence. La figure 5.10a montre la distribution de la distance d'édition pour l'algorithme de référence et pour *CT-Mapper*.

Analyse de la multimodalité

Dans l'étape suivante de l'évaluation de notre algorithme de projection, nous étudions la précision de l'algorithme de projection dans la détection de la couche de transport. Comme mentionné en 5.13, la complexité de la projection multimodale augmente significativement en raison d'importantes différences entre les couches de transport. Ce problème est traité par ce qui, dans le modèle de probabilité de transition, cherche à minimiser le biais dans l'algorithme de projection.

(a) Distance d'édition.

(b) Rappel/Recall et précision dans la détection de couche.

Figure 5.10 : Résultats.

Nous calculons le rappel et la précision pour une détection correcte de la couche une fois pour chaque couche. Le rappel et la précision générales pour l'ensemble du réseau sont déterminés comme la moyenne de rappel et précision pour chaque couche, pondérée par le nombre de nœuds.

La Figure 5.10b montre ces mesures comparées avec l'algorithme de référence. Nous avons remarqué que chaque hypothèse tenant compte d'un aspect spécifique des propriétés topologiques du réseau, peut introduire un biais dans le problème de projection. Comme montré en Figure 5.10b, le rappel et la précision globales de la détection de la couche correcte sont améliorées dans *CT-Mapper* comparé à l'algorithme de référence.

5.5 Discussion & conclusion

Dans cette étude nous avons proposé un algorithme de projection non supervisé (*CT-Mapper*) pour projeter des données éparses issues du réseau cellulaire sur un réseau multimodal de transport. Nous avons modélisé et construit le réseau multicouches de transport des couches métros, trains et routes pour l'Île-de-France. Le réseau multicouches de transport contient environ $16\,000$ nœuds et $26\,000$ arcs. Pour étudier la complexité de ce graphe, un modèle de probabilité de transition tenant compte du type de couches de transport et de leurs propriétés topologiques est estimé et utilisé dans un algorithme basé sur un MMC non supervisé. Nous avons conduit l'expérience sur un jeu de données de quatre-vingts trajectoires multimodales réelles issues de dix participants pendant un mois (aout à septembre 2014) pour évaluer notre algorithme. Tenant compte de la rareté de nos observations cellulaires (à la fréquence d'un évènement toutes les 15 minutes), le pourcentage de construction de parcours d'utilisateurs de smartphones atteint est notable. Pour

valider notre modèle de probabilité de transition, qui est mieux adapté à la complexité du réseau multimodal de transport, nous l'avons comparé avec l'algorithme de référence qui ne tient pas compte des propriétés de transport de chaque couche. Le résultat montre jusqu'à 20 % d'amélioration de la précision du premier sur le second. Nous espérons qu'utiliser une matrice de pondération dynamique, qui est compatible avec le modèle de trafic à différentes périodes de la journée, soit une amélioration probable de notre algorithme. Ce problème sera étudié dans de futurs travaux. L'amélioration de la précision des mesures de notre algorithme de projection en minimisant le biais émanant principalement de la multimodalité du réseau de transport est de grande importance et devra être discutée dans de futures contributions. Étudier la possibilité d'utiliser l'algorithme proposé en quasi-temps réel pour la supervision de trafic est une autre direction à envisager pour d'autres contributions.

6 TV-Whitespace

Nous aborderons dans ce chapitre l'utilisation dynamique des bandes de fréquences nommées *white space*. Il s'agit de bandes de fréquences libérées (comme celles de la télévision analogique) ou inutilisées localement.
Analyser les traces de communication des voyageurs journaliers dans de grandes villes et à l'échelle d'un pays fournit des informations très précieuses pour le dimensionnement du réseau cellulaire. Nous avons appris par exemple que le dimensionnement conventionnel du réseau mobile n'est plus optimal si l'on se place à l'échelle de l'heure ou de la journée. Nous observons que la plus forte densité de sites de communications cellulaires varie entre *très dense* (avec même de très hauts rejets d'appels et de connexion) et *très bas* et ceci avec un motif périodique régulier. Cette analyse met en lumière l'intérêt non seulement d'optimisation de la bande de fréquence grâce au *white space* (comme la radio cognitive le fait, cf. [83]) mais aussi de déplacer les ressources d'un lieu à l'autre, en fonction des déplacements de foules. Les architectures *femtocells* et *metrocells* ainsi que les récents standards dans les API permettant de gérer les ressources *TV whitespace* (cf. [147, 61, 53]) ont rendu cette gestion dynamique possible.

Le déploiement de *femtocells* et *metrocells* dans un réseau cellulaire fournit des architectures flexibles plus faciles à contrôler et à adapter aux conditions sans fil. Les techniques venant de la radio cognitive permettant l'allocation dynamique de spectre peuvent être utilisées dans ce contexte et donner un bénéfice stratégique à l'opérateur. *TV-Whitespace* est l'un des premiers standards soumis à enchères à déployer des licences de spectre. Il combine un ensemble de protocoles, des bases de données et des ressources spectrales pour fournir un tel mécanisme.

Nous estimons par conséquent qu'il est nécessaire d'avoir une gestion dynamique de l'endroit *où* la ressource est allouée. Nous proposons ici une architecture *femtocell* originale utilisant des algorithmes d'optimisation s'adaptant aux motifs de mobilité de populations à l'échelle de l'heure. Il est basé sur un système calculant des scores à partir de la charge du réseau, du coût de la bande de fréquences et de la qualité d'expérience (QoE). Il est important de considérer l'expérience utilisateur lors de l'allocation de fréquences dans le réseau de données cellulaire, plutôt que tenir compte uniquement du débit. Il a été montré que la perception utilisateur et sa satisfaction ne sont pas proportionnelles à des paramètres de qualité de services

tels que le débit. C'est pourquoi notre projet prend en compte non seulement les exigences du client, mais aussi les paramètres de qualité d'expérience. Lorsque le score d'une cellule décroît, sa bande de fréquences allouée décroît et inversement, permettant à la bande de fréquences d'être déplacée ailleurs selon les mouvements de population.

La suite de ce chapitre est organisée comme suit : nous présentons un résumé des résultats de notre analyse des traces du réseau cellulaire, puis suit la présentation de l'architecture cellulaire basée sur les *femtocell* et enfin nous décrivons l'algorithme d'optimisation. Nous concluons sur les résultats des simulations.

6.1 Analyse de population vue du réseau cellulaire

Comme nous l'avons vu dans les chapitres précédents, le GTP-C nous donne un aperçu de l'activité et de la mobilité humaine à l'échelle d'un pays.
Un résultat direct de l'analyse des traces nous fournit la charge du réseau cellulaire pendant une période. Par simplicité et pour le passage à l'échelle, nous avons choisi de limiter la granularité de nos intervalles de mesures à la demi-journée. En Figure 6.1, nous pouvons voir qu'à Paris, un nombre important de mobiles utilise le réseau cellulaire en matinée. Les traces montrent le nombre de connexions de données mobiles sur deux cellules différentes. À gauche, il s'agit d'une cellule de haute densité dans un quartier d'affaires qui montre que le weekend est très plat en termes de trafic alors qu'il est très dense pendant les jours de la semaine. L'autre cellule est plus équilibrée, car elle couvre un quartier touristique.

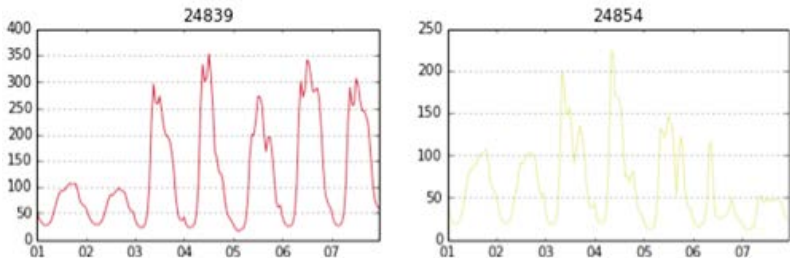


Figure 6.1 : Nombre de terminaux mobiles distincts par heure de la journée, sur une semaine et sur deux cellules dans le cœur de Paris.

Nous notons, grâce à une analyse générale de ces traces collectées sur plus d'une centaine de cellules à Paris et dans sa banlieue, que les zones résidentielles et celles de bureaux ont des pics d'activités à différents moments de la journée. Des cellules couvrant certains nœuds de transport font face à des pics de connexion correspondant à des arrivées massives de personnes (telles que l'arrivée d'un TGV).

6.2 Déploiements de femto/metro cells avec les techniques TV-Whitespace

Notre architecture provient du projet de recherche LCI4D [147]. L'idée principale est de développer une architecture femtocell possédant des aptitudes mesh basée sur un réseau hétérogène et sur des techniques provenant de la radio cognitive pour les pays émergents. L'architecture est basée sur des *femtocells* hybrides utilisant le LTE aussi bien que le 802.11. L'idée est de déléguer l'architecture complète à de petits entrepreneurs locaux et de mutualiser les opérations de gestion et de facturation.

6.2.1 L'architecture LCI4D

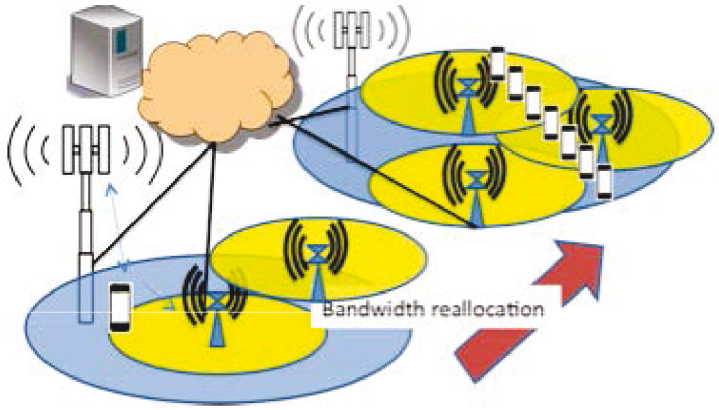


Figure 6.2 : Architecture femtocell, supportant l'offloading et l'équilibrage de charge.

L'architecture générale est décrite en Figure 6.2. Nous pouvons voir qu'elle est composée de trois éléments principaux : un ensemble dense de *femtocells*, un *backhaul mesh* basé sur le protocole propriétaire 802.11N (avec des garanties de qualité de service) et un réseau cellulaire de cœur hébergeant les fonctions de gestion du réseau ainsi que les fonctionnalités AAA.
L'équipement *femtocell* sélectionné est en cours de développement. Il supporte le concept de réseau hétérogène et a accès à de multiples technologies dont le LTE et les 802.11. Il est prévu pour supporter le protocole 802.AF TV-Whitespace. Un équipement femtocell est connecté à ses voisins au travers d'une interface LTE spécifique appelée X2. Elle est également connectée au backbone grâce à un réseau wifi mesh et au cœur de réseau. L'équipement de base a plusieurs versions. Il peut supporter 32, 64 ou 200 utilisateurs simultanés sur des bandes de 5, 10 et 20 Mhz. Il existe de nombreuses combinaisons de topologies, permettant l'overlap de *femtocells* et metrocells augmentant ainsi la couverture et la bande de fréquences. Nous

limitons notre étude à la première version supportant 32 utilisateurs et laissons le scénario d'overlap pour des travaux futurs.

6.2.2 Le Protocole TV-Whitespace

Le système de radio cognitive utilise l'architecture et le protocole de TV-Whitespace (TVWS) comme socle. L'idée vient du groupe de travail IEEE 802.11AF. Il s'agit du premier vrai système d'enchères sur l'allocation dynamique de spectre dans des fréquences soumises à licences. L'IETF a un intérêt dans les bandes TVWS (libérées par la télévision analogique), spécifiquement concernant le partage de spectre dynamique, dans cette bande, à travers une base de données de TVWS. Le partage de spectre peut prendre différentes formes, mais son propos est de permettre à certaines parties d'utiliser des fréquences lorsqu'elles ne sont pas utilisées par l'utilisateur principal dans le cadre d'une licence. Par exemple, si un système de communication n'a pas besoin d'une bande de fréquences à un moment précis, cette ressource peut être libérée dans un but commercial pendant cette période. Avec le partage dynamique, différents utilisateurs, parmi lesquels des entités gouvernementales ou commerciales, peuvent partager des bandes et accroître la disponibilité de cette ressource précieuse. Le partage dynamique du spectre permet aux utilisateurs inscrits d'interroger une base de données pour déterminer, pour un lieu donné, la liste des fréquences pouvant être utilisées tout en protégeant des interférences les entités soumises à licence. La base de données a été certifiée aux USA par la Federal Communications Commision (FCC) et elle est disponible pour les appareils sans-fil approuvés pas le FCC pour la bande TVWS. Les fabricants de matériel, les chercheurs et n'importe quel individu intéressé peuvent maintenant utiliser un système développé par Google pour identifier et utiliser les fréquences TVWS API [147] et [61].

6.3 Algorithme de réallocation de bande de fréquences

Nous voyons dans les traces réelles de télécommunication que la charge varie pendant les heures de la journée. L'idée principale est d'assigner dynamiquement les bandes de fréquences soumises à licence aux points où il est attendu d'avoir des goulots d'étranglement de trafic. La bande de fréquences est gérée avec le mécanisme TV-Whitespace [61]. Étant soumis à licence, il n'est pas gratuit et est provisionné à la demande et par opérateur. Un opérateur peut par conséquent décider de déplacer de la bande de fréquences d'un point à un autre et mettre à jour la base de données sans payer de frais supplémentaires. Si l'opérateur demande une bande de

fréquences additionnelle, ceci aura un coût qu'il faudra calculer. Nous utilisons la procédure de réallocation basée sur une *utility function* qui est la somme de deux scores : un pour le point de vue réseau et l'autre pour le point de vue qualité de service. Un score global (la somme des deux points de vue) dans les différents points est calculé et donne comme résultat la cellule candidate, qui a besoin de bande de fréquences supplémentaire et les autres où il est possible de réduire la taille du spectre. Il peut y avoir différents moyens de déclencher le mécanisme de réallocation. Dans nos simulations, nous déclenchons la réallocation à chaque fois que la charge d'une cellule croît de 10. Le score femtocell représente une valeur qui croît avec les coûts (bande de fréquences) et avec la charge supportée. Donc, lorsque le nombre d'utilisateurs augmente, la charge augmente de concert. Lorsque nous calculons le coût sur l'ensemble du réseau, les cellules avec un petit score seront identifiées et leur bande de fréquences sera réduite afin d'augmenter la capacité des cellules chargées. Les utilisateurs ont un score calculé en fonction de la bande de fréquences allouée. Ce mécanisme est contextuel et dépend de plusieurs paramètres, tels que la catégorie d'équipement et des caractéristiques du contenu (résolution, taille, etc.). Il a été grandement étudié dans différents travaux [61, 31].

Pour l'opérateur, le score est :

$$S_o(i) = [\gamma_{NL}\frac{NL(i)}{NLs} + \gamma_{Cop}\frac{Cop(i)}{Cops} + \gamma_{NC}(1 - \frac{NC(i)}{NCs})] \times \frac{1}{\gamma_{NL} + \gamma_{Cop} + \gamma_{NC}} \quad (6.1)$$

où :

- $Cop(i)$ est le coût pour l'opérateur lorsqu'il acquiert une bande de fréquences additionnelle à un instant i ;
- $NL(i)$ est la charge du réseau à l'instant i ;
- $NC(i)$ est le nombre de canaux *whitespace* utilisés à l'instant i ;
- NLs est la charge maximale pour la femtocell ;
- $Cops$ est le coût total ;
- NCs est le nombre maximal de canaux *whitespace* ;
- γ_{NL}, γ_{Cop} et γ_{NC} sont des constantes, ce qui permet de pondérer l'un des trois critères. Par exemple si l'on veut favoriser les *femtocells* à large capacité, nous pouvons remplacer par NLs lui-même ou par une fonction de NLs. Dans ce qui suit nous prenons γ_{NL}, γ_{Cop} et γ_{NC} tous égaux à 1. Ainsi l'évolution du score de l'opérateur est linéaire avec la charge, les whitespaces disponibles et le coût.

Pour le score client, nous choisissons les paramètres suivants :

$$S_{Cl}(i) = \frac{MOS_{NET}(i)}{S_{max}} \tag{6.2}$$

où :

— $MOS_{NET}(i) = A + B \times e^{-C \times \frac{D_{max}(i)}{D(i)}}$

— $D_{max}(i)$ est le débit maximal offert à un client ;

— $D(i)$ est le débit actuel du client ;

— $Smax$ est la valeur maximale du MOS ;

— A, B et C sont des paramètres contextuels. Ils sont considérés comme constants et dépendent du contexte.

Le débit $D(i)$ par utilisateur décroît logiquement avec l'arrivée de nouveaux utilisateurs dans la cellule. Il est proportionnel à $1/NL(i)$.

L'objectif d'optimisation est d'assigner bande de fréquences aux cellules à hauts scores en tenant compte des exigences de l'opérateur et du point de vue client. C'est pourquoi nous considérons un score pondéré tenant compte des deux points de vue :

$$S_T = \alpha * S_{Cl} + \beta * S_{op} \tag{6.3}$$

Considérant le nombre d'utilisateurs (la "charge"), l'évolution du score total S_T est la somme d'un terme linéaire, d'une exponentielle et d'une constante. Dépendant de la valeur du poids et de la constante, le score total peut avoir différentes formes comme montrées en Figure 6.3.

L'intérêt pour la perception du service par l'utilisateur peut varier d'un opérateur à l'autre. Par facilité, nous avons choisi un score identique et partagé par le réseau et l'utilisateur, ainsi α et β valent $1/2$.

Donc, notre objectif est de déplacer la bande de fréquences entre cellules en fonction de la variation dans le temps de $S_T = 1/2 * S_{Cl} + 1/2 * S_{op}$. Nous appliquons un algorithme de réattribution après l'évaluation qui scrute les cellules à scores bas (comme calculé en Figure 6.4) et réattribue les canaux TV *whitespace* pour améliorer les scores.

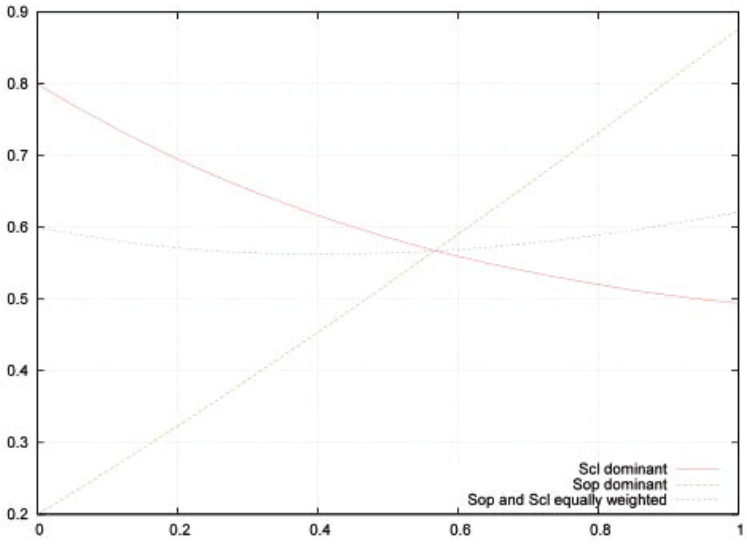


Figure 6.3 : Influence du poids dans la fonction de coût S_T en fonction de la charge.

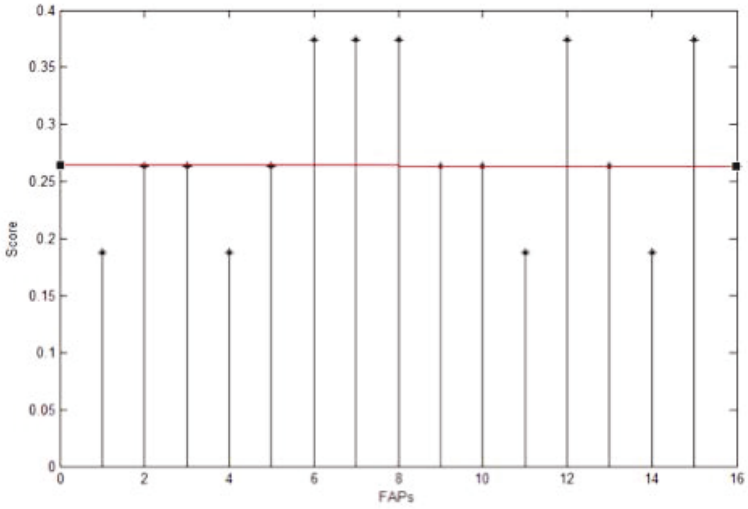


Figure 6.4 : S_T pour différentes cellules(femto access points, or FAPs) dans la ville.

6.4 Simulations et résultats

Nous voulons maintenant voir comment l'algorithme d'optimisation se comporte dans un scénario où les utilisateurs migrent progressivement d'un point à un autre. Nous utilisons le simulateur NS2 avec une topologie de 16 femtocells (femtocell access points, FAP). Le modèle montré en Figure 6.5 représente la topologie requise. Nous simulons le déplacement d'utilisateur de la partie gauche des FAPs vers le seul côté droit du FAP.

Chaque cellule a dix utilisateurs. Les utilisateurs consomment du trafic UDP/IP. La cellule cible va progressivement être saturée. Le choix d'avoir seulement dix utilisateurs dépend de l'équipement. Comme expliqué précédemment, nous prenons

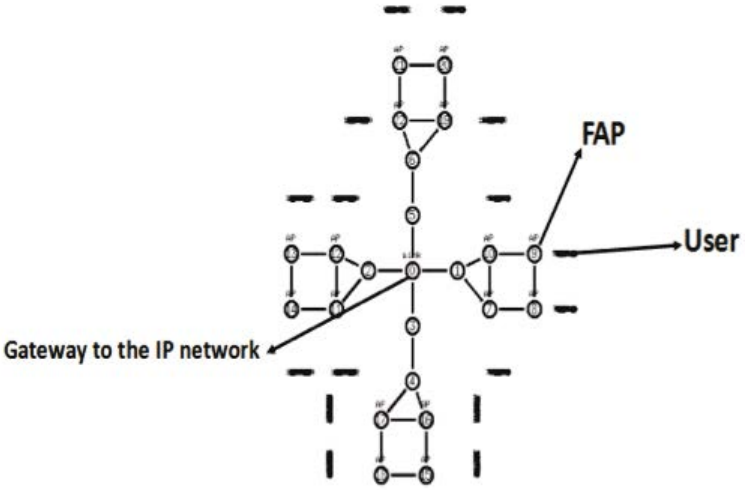


Figure 6.5 : Topologie initiale de simulation FAP.

comme exemple une *femtocell* supportant jusqu'à 32 utilisateurs. Il est par conséquent nécessaire d'être capable de faire face à tous les utilisateurs lors d'un déplacement massif. Le nombre peut bien sûr être changé et étendu en fonction de la topologie et des conditions physiques que nous voulons représenter.

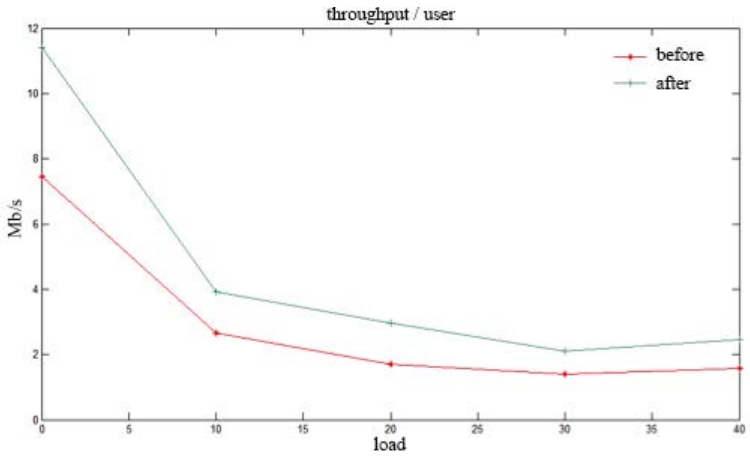


Figure 6.6 : Bande de fréquences allouée par utilisateur après activation de l'optimisation.

La figure 6.6 montre une comparaison entre le débit par utilisateur avec notre méthode basée sur la notation (la légende "after" est à comprendre *après réallocation*) comparé au même débit dans le cas d'une allocation statique. Les résultats de la simulation montrent la différence de débit que la *femtocell* sera capable de donner à chaque utilisateur lorsque le nombre d'utilisateurs augmente comme un pourcentage du nombre initial. Ces résultats confirment le bénéfice de notre méthode de réallocation de bande de fréquences basée sur la notation. De la même manière, le délai moyen avant réallocation et après réallocation TV *whitespace* sera amélioré en conséquence. La figure 6.7 nous donne l'augmentation moyenne du délai par utilisateur sur une cellule souffrant de cette charge supplémentaire.

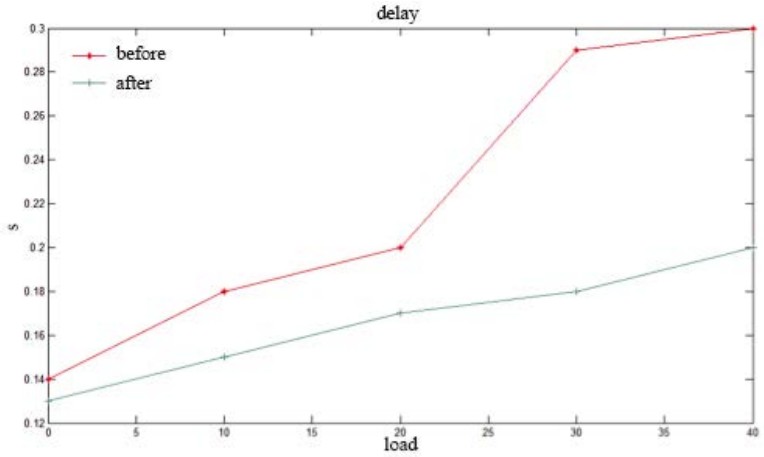


Figure 6.7 : Délais avant et après la procédure TV Whitespace.

Nous voyons comment le fait d'équilibrer la bande de fréquences améliore les paramètres de performance. Ici, nous supposons que les réseaux prennent les actions du protocole *whitespace* à 10%, 20% et 30% de charge supplémentaire. À ces étapes, il attribue une bande de fréquences additionnelle égale à ces charges supplémentaires rencontrées. Donc, lorsque le nombre d'utilisateurs supplémentaire dans la cellule atteint 20%, le système supprime 20% de bande de fréquences à une cellule vide et l'alloue à la cellule chargée.

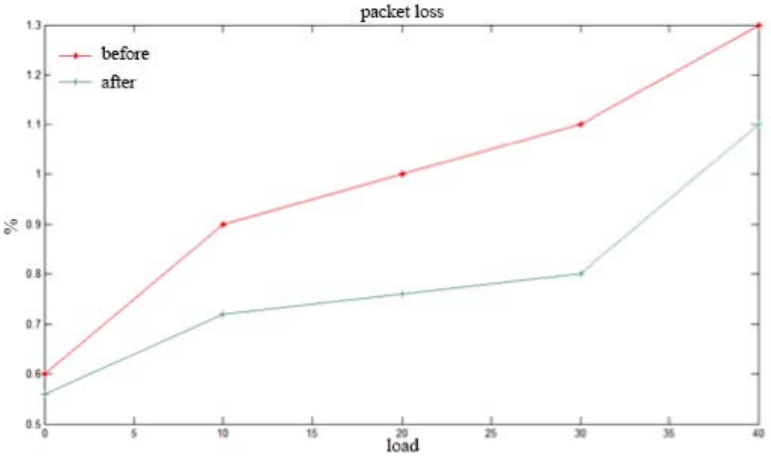


Figure 6.8 : Les pertes rencontrées avec la surcharge des cellules femto avant et après la réallocation TV Whitespace.

6.5 Conclusion

Nous avons présenté dans ce chapitre un algorithme de réallocation dynamique de spectre dans des conditions urbaines denses, en utilisant des techniques de *TV Whitespace* inspirées par les récents travaux de modélisation de la mobilité basés sur les traces issues de réseaux d'opérateurs mobiles et rendus possibles par les récentes

avancées dans le domaine du *big data*. L'algorithme nourrit une architecture hétérogène femtocell originale. Les *femtocells* coopératives partagent dynamiquement la bande de fréquences en fonction de la charge attendue du réseau. Basées sur des traces réelles, extraites d'un réseau de télécommunication pour une aire urbaine et dense, nous montrons comment le trafic et la charge du réseau varient en fonction de l'heure de la journée et comment ces variations ont un impact sur la qualité d'expérience du client, que l'allocation soit dynamique ou statique.

7 Conclusion

7.1 Contributions

Cette thèse propose d'utiliser le protocole GTP-C comme source unique dans l'étude de la mobilité humaine. Profitant des taux d'équipements élevés en matière de téléphones mobiles et de smartphones, un grand nombre d'études se basent sur les données provenant d'opérateurs mobiles, permettant l'accès à des données concernant une population et un territoire importants. Mais aucune étude, à notre connaissance, utilise exclusivement des informations issues du domaine paquet, préférant soit des données du domaine voix soit des données hybrides mêlant différentes sources.

Nous avons tout d'abord présenté l'architecture du réseau mobile et les flux en présence, ce qui nous permet de justifier notre choix d'utiliser ce protocole, en nous basant sur des critères de complexité de capture et d'analyse des différentes alternatives dans le domaine circuit ou hors du cœur de réseau. Nous avons aussi, à partir de ces informations, identifié les points de captures où nous pourrions collecter ces flux.

Puisqu'aucune solution générique n'est capable de capturer des flux avec les débits très élevés que l'on retrouve dans les POP d'opérateurs mobiles, ni de produire des données dans un format modulaire, nous avons développé une plateforme de sondes réseau passives distribuées dans le *core network*, capables de fonctionner sur le réseau de n'importe quel opérateur. Ces sondes ont été conçues pour supporter d'importants débits, traiter différents protocoles et générer des fichiers dans un format exploitable de façon efficace.
Les volumes générés par ces sondes, trop importants pour être injectés par un système de base de données classique, sont stockés et traités par un *cluster* Spark/Hadoop que nous avons mis en place.
Ces deux éléments, modulaires, forment une plateforme globale échangeant les données en utilisant un format ouvert et performant. Les modifications faites sur la structure des données depuis la sonde se répercutent sur la plateforme de traitement sans efforts.

Cette plateforme nous a permis de caractériser le protocole GTP-C dans une optique de travaux sur la mobilité humaine. Contrairement à nos travaux, Metzger et al. [87] avaient caractérisé ces flux en fonction de la charge des équipements de cœur. Comme attendu, il ressort de ces mesures que les intervalles de mises à jour GTP et la distance des sauts entre cellules s'accroissent en même temps que la densité de population de la zone concernée diminue.

Ces mêmes données ont servi de matériau à CT-Mapper dans lequel nous reconstituons des trajets d'utilisateurs de réseau mobile. Il permet à partir des données issues du réseau mobile de placer les points probables de la position de l'individu sur un graphe multimodal des réseaux de transport. Ce placement est fait grâce à un Modèle de Markov Caché et l'algorithme a été validé sur l'Île-de-France dans le cadre d'une expérimentation où les participants ont demandé à l'opérateur de leur fournir leurs traces de connexion. Cette démarche a l'avantage de valider l'algorithme et l'utilisation des traces dans les études de mobilité.

Finalement, l'occupation des cellules, estimée à partir de ces données, a servi de base à un algorithme de dimensionnement dynamique de réseaux mobile. Ce dimensionnement se fait en utilisant une infrastructure dynamique de *femtocells*, achetant aux enchères de la bande de fréquence dans le spectre de la télévision analogique pour assurer son *backhaul*. Le choix de l'achat se fait en fonction du coût de la ressource pondérée par un algorithme de mesure de la qualité d'expérience.

7.2 Perspectives

Dans cette thèse, nous avons validé l'utilisation du protocole GTP-C comme base à l'étude de la mobilité humaine dans une région dense, l'Île-de-France. Dans des travaux futurs, il serait intéressant de tester CT-Mapper dans des régions moins peuplées et de faire varier les fréquences de mises à jour pour arriver à détecter des seuils, permettant ainsi de valider finement les données issues de la caractérisation du réseau.
De même, pour les filtres, il serait intéressant de mesurer l'effet de ces techniques sur la distance entre le trajet vu depuis le réseau et le trajet réel.
Un autre axe de recherche est l'utilisation de ces données dans le but de détecter les anomalies réseau et de proposer des solutions proactives.

Enfin, nous pensons que l'utilisation de ces données, avec l'apparition de la LTE où le domaine circuit disparaît et que la voix passe sur IP dans le plan utilisateur, aura un rôle encore plus important dans l'étude du réseau mobile et de ce qu'il peut nous apprendre de la mobilité humaine dans les années à venir.

Bibliographie

[1] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA) ; S1 Application Protocol (S1AP)*. TS 36.413. 3rd Generation Partnership Project (3GPP), sept. 2008 (cf. p. 8).

[2] 3GPP. *General Packet Radio Service (GPRS) ; Evolved GPRS Tunnelling Protocol (eGTP) for EPS*. TS 29.274. 3rd Generation Partnership Project (3GPP), juil. 2008 (cf. p. 4, 7).

[3] 3GPP. *General Packet Radio Service (GPRS) ; GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface*. TS 29.060. 3rd Generation Partnership Project (3GPP) (cf. p. 4, 7).

[4] 3GPP. *GSM-UMTS Public Land Mobile Network (PLMN) Access Reference Configuration*. TS 24.002. 3rd Generation Partnership Project (3GPP), juin 2007 (cf. p. 5).

[5] 3GPP. *Mobile radio interface Layer 3 specification ; Core network protocols ; Stage 3*. TS 24.008. 3rd Generation Partnership Project (3GPP), sept. 2008 (cf. p. 6, 15).

[6] 3GPP. *UTRAN Iu interface Radio Access Network Application Part (RANAP) signalling*. TS 25.413. 3rd Generation Partnership Project (3GPP), sept. 2008 (cf. p. 6).

[7] A. ABADI, T. RAJABIOUN et P.A. IOANNOU. « Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data ». In : *Intelligent Transportation Systems, IEEE Transactions on* 16.2 (2015), p. 653–662 (cf. p. 27).

[8] Rachit AGARWAL, Vincent GAUTHIER, Monique BECKER, Thouraya TOUKABRIGUNES et Hossam AFIFI. « Large scale model for information dissemination with device to device communication using call details records ». In : *Computer Communications* 59 (2015), p. 1 –11 (cf. p. 77).

[9] Vaneet AGGARWAL, Emir HALEPOVIC, Jeffrey PANG, Shobha VENKATARAMAN et He YAN. « Prometheus : Toward Quality-of-experience Estimation for Mobile Apps from Passive Network Measurements ». In : *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. HotMobile '14. New York, NY, USA : ACM, 2014, 18 :1–18 :6 (cf. p. 21).

[10] A. AGUIAR, F.M.C. NUNES, M.J.F. SILVA, P.A. SILVA et D. ELIAS. « Leveraging Electronic Ticketing to Provide Personalized Navigation in a Public Transport Network ». In : *Intelligent Transportation Systems, IEEE Transactions on* 13.1 (2012), p. 213–220 (cf. p. 29).

[11] APACHE. *Apache Hadoop 2.7.2 – Apache Hadoop YARN*. `http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html` (cf. p. 24, 50).

[12] APACHE. *Apache Hive TM*. https://hive.apache.org/ (cf. p. 24, 25).

[13] APACHE. *Apache Mesos*. http://mesos.apache.org/ (cf. p. 50).

[14] APACHE. *Apache Spark - Lightning-Fast Cluster Computing*. https://spark.apache.org/ (cf. p. 24, 48, 49).

[15] APACHE. *HBase – Apache HBase Home*. http://hbase.apache.org/ (cf. p. 24, 25).

[16] APACHE. *Welcome to Apache Avro !* http://avro.apache.org/ (cf. p. 25, 45).

[17] APACHE. *Welcome to Apache Pig !* https://pig.apache.org/ (cf. p. 24).

[18] APACHE FOUNDATION. « HDFS Architecture Guide ». In : () (cf. p. 43).

[19] ARCEP. « Baromètre du numérique 2015 (27 novembre 2015) ». In : () (cf. p. 17).

[20] Fereshteh ASGARI, Alexis SULTAN, Haoyi XIONG, Vincent GAUTHIER et Mounim EL-YACOUBI. « CT-Mapper : Mapping Sparse Multimodal Cellular Trajectories using a Multilayer Transportation Network ». In : (2016). arXiv : 1604.06577 [cs.SI] (cf. p. 15).

[21] BAY AREA BIKE SHARE. *Introducing Bay Area Bike Share, your new regional transit system*. http://www.bayareabikeshare.com/ (cf. p. 19).

[22] Vincent D BLONDEL, Adeline DECUYPER et Gautier KRINGS. « A survey of results on mobile phone datasets analysis ». In : *EPJ Data Science* 4.1 (déc. 2015), p. 10 (cf. p. 77).

[23] D. BROCKMANN, L. HUFNAGEL et T. GEISEL. « The scaling laws of human travel ». In : *Nature* 439.7075 (2006), p. 462–465 (cf. p. 27).

[24] BSON. *BSON - Binary JSON*. http://bsonspec.org/ (cf. p. 25, 43).

[25] Vernon K C BUMGARDNER et Victor W MAREK. « Scalable hybrid stream and hadoop network analysis system ». In : *Proceedings of the 5th ACM/SPEC international conference on Performance engineering*. ACM, 2014, p. 219–224 (cf. p. 25).

[26] CHOFFNES. *Mobilyzer : Mobile Network Measurements Made Easy*. http://mobilyzer-project.mobi/ (cf. p. 20).

[27] CITY OF NEW YORK NYC. *NYC Taxi & Limousine Commission - Trip Record Data*. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml (cf. p. 19).

[28] Balázs Cs CSÁJI, Arnaud BROWET, Vincent A TRAAG et al. « Exploring the mobility of mobile phone users ». In : *Physica A : Statistical Mechanics and its Applications* 392.6 (2013), p. 1459–1473 (cf. p. 27).

[29] J DEAN et S GHEMAWAT. « MapReduce : simplified data processing on large clusters ». In : *Commun. ACM* (2008) (cf. p. 47).

[30] P. DEUTSCH. *DEFLATE Compressed Data Format Specification version 1.3*. RFC1951. 1996 (cf. p. 45, 46).

[31] Mamadou Tourad DIALLO, Hassnaa MOUSTAFA, Hossam AFIFI et Nicolas MARECHAL. « Context aware quality of experience for audio-visual service groups ». en. In : *IEEE Commun. Lett.* 8.2 (mar. 2013), p. 9–11 (cf. p. 105).

[32] DISCO. *Disco MapReduce*. http://discoproject.org/ (cf. p. 24).

[33] Matthew F. DIXON, Spencer P. AIELLO, Funmi FAPOHUNDA et William GOLDSTEIN. « Detecting Mobility Patterns in Mobile Phone Data from the Ivory Coast ». In : "*Proc. of The main conference on the scientific analysis of mobile phone datasets*". NETMOB'13. 2013 (cf. p. 77, 78).

[34] John DOYLE, Peter HUNG, Damian KELLY, Sean MCLOONE et Ronan FARRELL. « Utilising Mobile Phone Billing Records for Travel Mode Discovery ». In : *22nd IET Irish Signals and Systems Conference, ISSC*. 2011 (cf. p. 28, 77).

[35] DPDK. *DPDK*. `http://dpdk.org/` (cf. p. 23, 36).

[36] EMULEX. *Emulex Endace Visibility Products*. `http://www.emulex.com/visibility/` (cf. p. 23, 37).

[37] ENAIKOON. *OpenCellID - OpenCellID*. `http://opencellid.org/` (cf. p. 20).

[38] Michal FICEK, Tomáš POP, Petr VLÁČIL et al. « Performance Study of Active Tracking in a Cellular Network Using a Modular Signaling Platform ». In : *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. MobiSys '10. New York, NY, USA : ACM, 2010, p. 239–254 (cf. p. 20).

[39] Centre FRANCE. *Téléphonie : zones blanches, le sud de la Creuse espère [Carte]*. `http://www.lamontagne.fr/limousin/actualite/2015/06/12/telephonie-zones-blanches-le-sud-de-la-creuse-espere-carte_11477398.html` (cf. p. 58).

[40] Jean-Loup GAILLY. *The gzip home page*. `http://www.gzip.org/` (cf. p. 45).

[41] GANDI. *Gandi/packet-journey*. `https://github.com/Gandi/packet-journey` (cf. p. 36).

[42] Vincent GAUTHIER, Fereshteh ASGARI, Mounim EL-YACOUBI et Alexis SULTAN. « Procédé d'estimation de trajectoires utilisant des données mobiles ». Brev. 2015 (cf. p. 15).

[43] Fosca GIANNOTTI, Mirco NANNI, Fabio PINELLI et Dino PEDRESCHI. « Trajectory Pattern Mining ». In : *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '07. 2007, p. 330–339 (cf. p. 27).

[44] Fosca GIANNOTTI, Mirco NANNI, Dino PEDRESCHI et al. « Unveiling the Complexity of Human Mobility by Querying and Mining Massive Trajectory Data ». In : *The VLDB Journal* 20.5 (oct. 2011), p. 695–719 (cf. p. 27, 77).

[45] Oberhumer Com GMBH. *oberhumer.com : LZO real-time data compression library*. `http://www.oberhumer.com/opensource/lzo/` (cf. p. 45).

[46] C.Y. GOH, J. DAUWELS, N. MITROVIC et al. « Online map-matching based on Hidden Markov model for real-time traffic sensing applications ». In : *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. 2012, p. 776–781 (cf. p. 28, 92).

[47] Marta C GONZÁLEZ, César A HIDALGO et Albert-László BARABÁSI. « Understanding individual human mobility patterns ». In : *Nature* 453.7196 (2008), p. 779–782 (cf. p. 2, 68).

[48] Marta C. GONZÁLEZ, César A. HIDALGO et Albert-László BARABÁSI. « Understanding individual human mobility patterns ». In : *Nature* 453.7196 (2008), p. 779–782 (cf. p. 27).

[49] GOOGLE. *Android*. https://www.android.com/ (cf. p. 20).

[50] GOOGLE. *Google BigQuery - Fully Managed Big Data Analytics Service*. https://cloud.google.com/bigquery/ (cf. p. 25).

[51] GOOGLE. *google/snappy*. https://github.com/google/snappy (cf. p. 45).

[52] GOOGLE. *Protocol Buffers — Google Developers*. https://developers.google.com/protocol-buffers/ (cf. p. 25, 45).

[53] GOOGLE. *Spectrum Database – Google*. https://www.google.com/get/spectrumdatabase/ (cf. p. 101).

[54] S. HANKS, T. LI, D. FARINACCI et P. TRAINA. *Generic Routing Encapsulation (GRE)*. RFC1701. 1994 (cf. p. 7).

[55] Juan Carlos HERRERA, Saurabh AMIN, Alexandre BAYEN et al. *Dynamic estimation of OD matrices for freeways and arterials*. Institute of Transportation Studies, UC Berkeley, 2007 (cf. p. 18).

[56] Michio HONDA, Felipe HUICI, Giuseppe LETTIERI et Luigi RIZZO. « mSwitch : A Highly-scalable, Modular Software Switch ». In : *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*. SOSR '15. New York, NY, USA : ACM, 2015, 1 :1–1 :13 (cf. p. 36).

[57] HORN, CHRISTOPHER KLAMPFL, STEFAN CIK, MICHAEL REITER, THOMAS. « Detecting Outliers in Cell Phone Data : Correcting Trajectories to Improve Traffic Modeling ». In : *Transportation Research Record : Journal of the Transportation Research Board* (2014) (cf. p. 26, 68).

[58] Congwei HU, Wu CHEN, Yongqi CHEN et Dajie LIU. « Adaptive Kalman Filtering for Vehicle Navigation ». In : *Journal of Global Positioning Systems* 2.1 (2003), p. 42–47 (cf. p. 28).

[59] Britta HUMMEL. « Map Matching for vehicle Guidances ». In : *Dynamic and Mobile GIS : Investigating Changes in Space and Time*. Sous la dir. de Roland BILLEN, Elsa JOAO et David FORREST. CRC Press, 2006 (cf. p. 28, 92).

[60] Timothy HUNTER, Teodor MOLDOVAN, Matei ZAHARIA et al. « Scaling the mobile millennium system in the cloud ». In : *Proceedings of the 2nd ACM Symposium on Cloud Computing - SOCC '11*. 2011, p. 1–8 (cf. p. 24, 28).

[61] IEEE. *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5 : Television White Spaces (TVWS) Operation*. https://standards.ieee.org/findstds/standard/802.11af-2013.html. 2013 (cf. p. 101, 104, 105).

[62] INDEGO. *Indego Bike Share Stations - OpenDataPhilly*. https://www.opendataphilly.org/dataset/bike-share-stations (cf. p. 19).

[63] *Institut géographique national*. http://www.ign.fr/. 2015 (cf. p. 85).

[64] Corina IOVAN, Ana-Maria OLTEANU-RAIMOND, Thomas COURONNÉ et Zbigniew SMOREDA. « Moving and Calling : Mobile Phone Data Quality Measurements and Spatiotemporal Uncertainty in Human Mobility Studies ». In : *Geographic Information Science at the Heart of Europe*. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, 2013, p. 247–265 (cf. p. 26).

[65] Corina IOVAN, Ana-Maria OLTEANU-RAIMOND, Thomas COURONNÉ et Zbigniew SMOREDA. « Moving and Calling : Mobile Phone Data Quality Measurements and Spatiotemporal Uncertainty in Human Mobility Studies ». In : *Geographic Information Science at the Heart of Europe*. Sous la dir. de Danny VANDENBROUCKE, Bénédicte BUCHER et Joep CROMPVOETS. Lecture Notes in Geoinformation and Cartography. Springer International Publishing, 2013, p. 247–265 (cf. p. 85).

[66] Sibren ISAACMAN, Richard BECKER, Ramón CÁCERES et al. « Human Mobility Modeling at Metropolitan Scales ». In : *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. MobiSys '12. New York, NY, USA : ACM, 2012, p. 239–252 (cf. p. 2).

[67] ITU-T. *Q.703 : Signalling link*. Rapp. tech. ITU-I, 1997 (cf. p. 5).

[68] ITU-T. *Q.713 : Signalling connection control part formats and codes*. Rapp. tech. ITU-I, 2001 (cf. p. 5).

[69] ITU-T. *Q.771 : Functional description of transaction capabilities*. Rapp. tech. ITU-I, 1997 (cf. p. 5).

[70] ITU-T. *Q.772 : Transaction capabilities information element definitions*. Rapp. tech. ITU-I, 1997 (cf. p. 5).

[71] ITU-T. *Q.773 : Transaction capabilities formats and encoding*. Rapp. tech. ITU-I, 1997 (cf. p. 5).

[72] ITU-T. *Q.774 : Transaction capabilities procedures*. Rapp. tech. ITU-I, 1997 (cf. p. 5).

[73] ITU-T. *Q.775 : Guidelines for using transaction capabilities*. Rapp. tech. ITU-I, 1997 (cf. p. 5).

[74] Yousun JEONG. *Big Telco Real-Time Network Analytics | Schedule | Spark Summit Europe 2015*. `https://spark-summit.org/eu-2015/events/big-telco-real-time-network-analytics/` (cf. p. 1).

[75] Shan JIANG, Joseph FERREIRA Jr. et Marta C GONZALEZ. « Discovering urban spatial-temporal structure from human activity patterns ». In : *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. ACM, 2012, p. 95–102 (cf. p. 2, 19).

[76] JSON. *JSON*. `http://www.json.org/` (cf. p. 25, 43).

[77] Chaogui KANG, Stanislav SOBOLEVSKY, Yu LIU et Carlo RATTI. « Exploring Human Movements in Singapore : A Comparative Analysis Based on Mobile Phone and Taxicab Usages ». In : *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*. UrbComp '13. 2013, 1 :1–1 :8 (cf. p. 77).

[78] Youngseok LEE, Wonchul KANG et Hyeongu SON. « An Internet traffic analysis method with MapReduce ». In : *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*. Avr. 2010, p. 357–361 (cf. p. 25).

[79] J. LINN. *The Kerberos Version 5 GSS-API Mechanism*. RFC1964. 1996 (cf. p. 50, 51).

[80] Lu LIU. « Data Model and Algorithms for Multimodal Route Planning with Transportation Networks ». Thèse de doct. Technical University of Munich (TUM), 2011 (cf. p. 29, 79).

[81] Xi LIU, Li GONG, Yongxi GONG et Yu LIU. « Revealing travel patterns and city structure with taxi trip data ». In : (2013). arXiv : 1310.6592 [physics.soc-ph] (cf. p. 19).

[82] Yu LIU, Zhengwei SUI, Chaogui KANG et Yong GAO. « Uncovering Patterns of Inter-Urban Trip and Spatial Interaction from Social Media Check-In Data ». In : *PLoS One* 9.1 (2014), e86026 (cf. p. 2, 19, 27).

[83] Mario LÓPEZ-MARTÍNEZ, Juan J ALCARAZ, Javier VALES-ALONSO et Joan GARCIA-HARO. « Automated spectrum trading mechanisms : understanding the big picture ». en. In : *Wireless Netw* 21.2 (2014), p. 685–708 (cf. p. 101).

[84] Thomas LOUAIL, Maxime LENORMAND, Oliva G CANTU ROS et al. « From mobile phone data to the spatial structure of cities ». en. In : *Sci. Rep.* 4 (2014), p. 5276 (cf. p. 27).

[85] M-LAB. *M-Lab*. http://www.measurementlab.net/ (cf. p. 25).

[86] G Robert MALAN et Farnam JAHANIAN. « An Extensible Probe Architecture for Network Protocol Performance Measurement ». In : *Proceedings of the ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. SIGCOMM '98. New York, NY, USA : ACM, 1998, p. 215–227 (cf. p. 23, 33).

[87] Florian METZGER, Albert RAFETSEDER, Peter ROMIRER-MAIERHOFER et Kurt TUTSCHKU. « Exploratory Analysis of a GGSNs PDP Context Signaling Load ». In : *Journal of Computer Networks and Communications* 2014 (2014), p. 1–13 (cf. p. 21, 112).

[88] MOBIPERF. *Mobiperf/MobiPerf*. https://github.com/Mobiperf/MobiPerf (cf. p. 20).

[89] Yves-Alexandre de MONTJOYE, César A. HIDALGO, Michel VERLEYSEN et Vincent D. BLONDEL. « Unique in the Crowd : The privacy bounds of human mobility ». In : *Sci. Rep.* 3 (2013) (cf. p. 79).

[90] *Moves*. https://www.moves-app.com/. 2015 (cf. p. 85, 93).

[91] MOZILLA. *MLS - Overview*. https://location.services.mozilla.com/ (cf. p. 20).

[92] NAPATECH. *Napatech*. http://www.napatech.com/ (cf. p. 23, 37).

[93] Rob van NES. « Design of multimodal transport networks : a hierarchical approach ». Thèse de doct. Technical University of Delft (DUP), 2002 (cf. p. 29).

[94] Paul NEWSON et John KRUMM. « Hidden Markov Map Matching Through Noise and Sparseness ». In : *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '09. Seattle, Washington : ACM, 2009, p. 336–343 (cf. p. 28, 92).

[95] Anastasios NOULAS, Salvatore SCELLATO, Renaud LAMBIOTTE, Massimiliano PONTIL et Cecilia MASCOLO. « A tale of many cities : universal patterns in human urban mobility ». In : (2011). arXiv : 1108.5355 [physics.soc-ph] (cf. p. 2, 19, 27).

[96] NTOP. *PF_RING*. http://www.ntop.org/products/packet-capture/pf_ring/ (cf. p. 23, 36).

[97] OPENSTACK. *Welcome to Swifts documentation ! — swift 2.5.1.dev236 documentation*. http://docs.openstack.org/developer/swift/ (cf. p. 43).

[98] *OpenStreetMap project*. http://www.OpenStreetMap.org/. 2015 (cf. p. 85).

[99] ORANGE. *D4D Challenge Sénégal est une réussite !* http://www.d4d.orange.com/fr/presentation/dotation-et-jury/Folder/D4D-Challenge-Senegal-est-une-reussite (cf. p. 21).

[100] PACKETLOOP. *packetloop/packetpig*. https://github.com/packetloop/packetpig (cf. p. 25).

[101] C. PERKINS. *IP Encapsulation within IP*. RFC2003. 1996 (cf. p. 7).

[102] J. POSTEL. *User Datagram Protocol*. RFC0768. 1980 (cf. p. 4).

[103] Inc RED HAT. *Storage for your Cloud. — Gluster*. http://www.gluster.org/ (cf. p. 43).

[104] Sasank REDDY, Min MUN, Jeff BURKE et al. « Using Mobile Phones to Determine Transportation Modes ». In : *ACM Trans. Sen. Netw.* 6.2 (2010), 13 :1–13 :27 (cf. p. 77).

[105] RIPE. *RIPE Atlas - RIPE Network Coordination Centre*. https://atlas.ripe.net/ (cf. p. 25).

[106] RIPE-NCC. *RIPE-NCC/hadoop-pcap*. https://github.com/RIPE-NCC/hadoop-pcap (cf. p. 13, 25).

[107] D RIVERA, S BLASCO, J BUSTOS-JIMENEZ et J SIMMONDS. « Spin lock killed the performance star ». In : *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*. Nov. 2015, p. 1–6 (cf. p. 35).

[108] Luigi RIZZO. « Revisiting Network I/O APIs : The netmap Framework ». In : *Queueing Syst.* 10.1 (2012), p. 30 (cf. p. 23, 36).

[109] Sanae ROSEN, Haokun LUO, Qi Alfred CHEN et al. « Understanding RRC state dynamics through client measurements with mobilyzer ». In : *Proceedings of the 6th annual workshop on Wireless of the students, by the students, for the students*. ACM, 2014, p. 17–20 (cf. p. 20).

[110] M. ROSVALL, A. TRUSINA, P. MINNHAGEN et K. SNEPPEN. « Networks and Cities : An Information Perspective ». In : *Phys. Rev. Lett.* 94.2 (2005) (cf. p. 86).

[111] Camille ROTH, Soong Moon KANG, Michael BATTY et Marc BARTHÉLEMY. « Structure of Urban Movements : Polycentric Activity and Entangled Hierarchical Flows ». In : *PLoS One* 6.1 (2011), e15923 (cf. p. 19, 27).

[112] By Sandy RYZA. *Apache Spark Resource Management and YARN App Models - Cloudera Engineering Blog*. http://blog.cloudera.com/blog/2014/05/apache-spark-resource-management-and-yarn-app-models/. 2014 (cf. p. 49).

[113] SCYLLADB. *Scylla DB*. http://www.scylladb.com/ (cf. p. 36).

[114] SEAGATE TECHOLOGY. *Lustre*. http://lustre.org/ (cf. p. 43).

[115] SECDEV. *Secdev/scapy*. https://github.com/secdev/scapy (cf. p. 15).

[116] Julian SEWARD. *bzip2 : Home*. http://www.bzip.org/ (cf. p. 45).

[117] Filippo SIMINI, Marta C. GONZÁLEZ, Amos MARITAN et Albert-László BARABÁSI. « A universal model for mobility and migration patterns ». In : *Nature* 484.7392 (2012), p. 96–100 (cf. p. 27).

[118] Zbigniew SMOREDA, Ana-Maria OLTEANU-RAIMOND et Thomas COURONNÉ. « Spatio-temporal data from mobile phones for personal mobility assessment ». In : *Transport survey methods : best practice for decision making*. Sous la dir. de Johanna ZMUD, Martin LEE-GOSSELIN, Juan Antonio CARRASCO et Marcela A. MUNIZAGA. Emerald Group Publishing, 2013 (cf. p. 28, 77, 78).

[119] K SNEPPEN, A TRUSINA et M ROSVALL. « Hide-and-seek on complex networks ». In : *Europhysics Letters (EPL)* 69.5 (2005), p. 853–859 (cf. p. 86).

[120] Chaoming SONG, Zehui QU, Nicholas BLUMM et Albert-László BARABÁSI. « Limits of predictability in human mobility ». In : *Science* 327.5968 (2010), p. 1018–1021 (cf. p. 27).

[121] STIF. *Comment compter les voyageurs ? les methodes utilisées sur notre ligne*. https://malignep.transilien.com/2015/03/17/comment-compter-les-voyageurs-les-methodes-utilisees-sur-notre-ligne/. 2015 (cf. p. 18).

[122] A SULTAN, M ELKOUKI, H AFIFI, V GAUTHIER et M MAROT. « A dynamic femto cell architecture using TV Whitespace improving user experience of urban Crowds ». In : *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*. Août 2015, p. 886–891 (cf. p. 14).

[123] Alexis SULTAN, Farid BENBADIS, Vincent GAUTHIER et Hossam AFIFI. « Mobile Data Network Analysis Platform ». In : *Proceedings of the 6th International Workshop on Hot Topics in Planet-Scale Measurement*. ACM, 2015, p. 13–18 (cf. p. 14).

[124] Lijun SUN, Kay W AXHAUSEN, Der-Horng LEE et Xianfeng HUANG. « Understanding metropolitan patterns of daily encounters ». en. In : *Proc. Natl. Acad. Sci. U. S. A.* 110.34 (2013), p. 13774–13779 (cf. p. 19, 27).

[125] TELECOM ITALIA. *The Contest | BigData Challenge*. http://www.telecomitalia.com/tit/en/bigdatachallenge/contest.html (cf. p. 21).

[126] Arvind THIAGARAJAN, Lenin RAVINDRANATH, Hari BALAKRISHNAN, Samuel MADDEN et Lewis GIROD. « Accurate, Low-energy Trajectory Mapping for Mobile Devices ». In : *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*. NSDI'11. USENIX Association, 2011, p. 267–280 (cf. p. 28, 92).

[127] Arvind THIAGARAJAN, Lenin RAVINDRANATH, Katrina LACURTS et al. « VTrack : Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones ». In : *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. SenSys '09. Berkeley, California, 2009, p. 85–98 (cf. p. 28, 92).

[128] W3C. *Extensible Markup Language (XML)*. https://www.w3.org/XML/ (cf. p. 43).

[129] Fergal WALSH et Alexei POZDNOUKHOV. « Spatial structure and dynamics of urban communities ». In : (2011), p. 1–8 (cf. p. 21).

[130] Pu WANG, Timothy HUNTER, Alexandre M BAYEN, Katja SCHECHTNER et Marta C GONZÁLEZ. « Understanding Road Usage Patterns in Urban Areas ». In : *Sci. Rep.* 2 (2012) (cf. p. 27).

[131] Kevin WEIL. *kevinweil/elephant-bird*. https://github.com/kevinweil/elephant-bird (cf. p. 45).

[132] Sage A WEIL, Scott A BRANDT, Ethan L MILLER, Darrell D E LONG et Carlos MALTZAHN. « Ceph : a scalable, high-performance distributed file system ». In : *Proceedings of the 7th symposium on Operating systems design and implementation*. USENIX Association, 2006, p. 307–320 (cf. p. 43).

[133] WISH7CODE. *Openbmap*. https://radiocells.org/ (cf. p. 20).

[134] T WOLF, R RAMASWAMY et S BUNGA. « An Architecture for Distributed Real-Time Passive Network Measurement ». In : *14th IEEE International Symposium on Modeling, Analysis, and Simulation*. IEEE, p. 335–344 (cf. p. 23).

[135] Wei WU, Yue WANG, Joao Bartolo GOMES et al. « Oscillation Resolution for Mobile Phone Cellular Tower Data to Enable Mobility Modelling ». In : *Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management - Volume 01*. IEEE Computer Society, 2014, p. 321–328 (cf. p. 26, 69).

[136] Zhang XIAOLI, Lai YUE et Xie SHENGLI. « A GTP stateful inspection method based on network processor ». In : *2008 11th IEEE Singapore International Conference on Communication Systems*. IEEE, nov. 2008, p. 994–998 (cf. p. 23, 35).

[137] Hao XU, Hongchao LIU, Chin-Woo TAN et Yuanlu BAO. « Development and Application of a Kalman Filter and GPS Error Correction Approach for Improved Map matching ». In : *Journal of Intelligent Transportation Systems* 14.1 (2010), p. 27–36 (cf. p. 28).

[138] Ming XU, Jianping WU, Yiman DU et al. « Discovery of Important Crossroads in Road Network using Massive Taxi Trajectories ». In : (2014). arXiv : 1407.2506 [cs.AI] (cf. p. 2, 27).

[139] Qiang XU, Alexandre GERBER, Zhuoqing Morley MAO et Jeffrey PANG. « AccuLoc : practical localization of performance measurements in 3G networks ». In : *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, p. 183–196 (cf. p. 21, 22, 54).

[140] YAML. *The Official YAML Web Site*. http://yaml.org/ (cf. p. 43).

[141] Dingyu YANG, Dongxiang ZHANG, Kian-Lee TAN, Jian CAO et Frédéric LE MOUËL. « CANDS : continuous optimal navigation via distributed stream processing ». In : *Proceedings VLDB Endowment* 8.2 (2014), p. 137–148 (cf. p. 24).

[142] Jing YUAN, Yu ZHENG et Xing XIE. « Discovering Regions of Different Functions in a City Using Human Mobility and POIs ». In : *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. 2012, p. 186–194 (cf. p. 27).

[143] Matei ZAHARIA, Mosharaf CHOWDHURY, Tathagata DAS et al. « Resilient distributed datasets : a fault-tolerant abstraction for in-memory cluster computing ». In : *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2012, p. 2–2 (cf. p. 24).

[144] Martin ZALTZ AUSTWICK, Oliver O'BRIEN, Emanuele STRANO et Matheus VIANA. « The Structure of Spatial Networks and Communities in Bicycle Sharing Systems ». In : *PLoS One* 8 (sept. 2013), p. 74685 (cf. p. 2, 19).

[145] Desheng ZHANG, Jun HUANG, Ye LI et al. « Exploring Human Mobility with Multi-source Data at Extremely Large Metropolitan Scales ». In : *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. MobiCom '14. New York, NY, USA : ACM, 2014, p. 201–212 (cf. p. 21).

[146] X ZHOU, M PETROVIC, T ESKRIDGE, M CARVALHO et X TAO. « Exploring Netflow Data using Hadoop ». In : () (cf. p. 25).

[147] Lei ZHU, Vincent CHEN, John MALYAR, Subir DAS et Pete MCCANN. « Protocol to Access White-Space (PAWS) Databases ». In : (mai 2015) (cf. p. 101, 103, 104).

Table des figures

Liste des tableaux