



La mobilité sociale : Modèles et traces

Hervé Costantini

► To cite this version:

Hervé Costantini. La mobilité sociale : Modèles et traces. Autre [cs.OH]. Conservatoire national des arts et métiers - CNAM, 2012. Français. NNT : 2012CNAM0827 . tel-00770978

HAL Id: tel-00770978

<https://theses.hal.science/tel-00770978>

Submitted on 7 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École Doctorale du Conservatoire National des Arts et Métiers, ED 130

Centre de Recherche en Informatique et Communication
Laboratoire CEDRIC, EA 4629

THÈSE DE DOCTORAT

présentée par : **Hervé COSTANTINI**

soutenue le : **jeudi 18 octobre 2012**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Discipline / Spécialité : **Informatique**

La mobilité sociale Modèles et Traces

THÈSE DIRIGÉE PAR

Dr BOUMERDASSI Selma

MCF/HDR

CNAM - CEDRIC

RAPPORTEURS

Pr BEYLOT André-Luc

Professeur

INPT/ENSEEIH

Pr KHEDDOUCI Hamamache

Professeur

LIRIS/UCBL

EXAMINATEURS

Pr BARKAOUI Kamel

Professeur

CNAM

Pr MONSUEZ Bruno

Professeur

ENSTA

Dr RENAULT Éric

MCF/HDR

Télécom SudParis

MEMBRES INVITÉS

Dr BOUZEFRANE Samia

MCF/HDR

CNAM - CEDRIC

À mon épouse, Chantal.

À nos enfants, Isaac, Esther, Abraham, Jacob.

« Rien ne peut se faire simplement chez les gens qui montent d'un étage social à l'autre. »,

Honoré de Balzac

Remerciements

Mon père disait dans sa grande simplicité : « Remercier, c'est difficile. ». Donc en toute simplicité, voici par ordre alphabétique, les nom, fonction et actions dont j'ai bénéficiées, des personnes envers lesquelles je me sens redevable :

- BARKAOUI, Kamel, professeur des universités, responsable de l'équipe de Recherche VESPA à laquelle je suis rattaché, mise à disposition de son vaste savoir en matière de systèmes sûrs et de méthodes formelles, délégation de charges de cours au sein du département informatique du CNAM ;
- BERTHELIN, Joël, ingénieur de recherche, responsable du maintien en service des installations informatiques, conseils et soutien ;
- BEYLOT, André-Luc, professeur des universités, rapporteur de ma thèse ;
- BOUMERDASSI, Selma, maître de conférence, habilitée à diriger des recherches, qualifiée aux fonctions de professeur des université, direction de ma thèse, apprentissage à la rédaction et à la soumission d'articles internationaux, délégation de charges de cours au sein du département réseau du CNAM ;
- BOUZEFRANE, Samia, maître de conférence, rédaction de rapport en vue de l'inscription en thèse ;
- GAL, Viviane, secrétaire administrative du CEDRIC, conseils, soutien, disponibilité ;
- GRESSIER-SOUDAN, Éric, professeur des universités, élaboration de mon sujet de thèse, mon inscription initiale en thèse ;
- KHEDDOUCI, Hamamache, professeur des universités, rapporteur de ma thèse ;
- MILOCCO, Ruben, professeur des universités, invitation à la rédaction commune d'articles internationaux ;
- PUGLIESE, Henri, ingénieur d'étude, maintien en service des installations informa-

tiques ;

- RENAULT, Éric, maître de conférence, habilité à diriger des recherches, qualifié aux fonctions de professeur des universités, invitation à la rédaction commune d'articles internationaux, délégation de charges de cours à Telecom SudParis ;

Résumé

L’avenir de la communication est perçu comme étant quasiment exclusivement constitué de *nœuds mobiles* évoluant dans un *réseau mobile*. Dans ce contexte, différentes approches contribuent continuellement à l’amélioration directe ou indirecte des délais d’acheminement des informations échangées entre les utilisateurs, dont :

- l’analyse des traces ;
- l’évaluation des performances ;
- les services de localisation ;
- le routage.

Nous décrivons chacun de ses thèmes et proposons des solutions faisant évoluer l’état de l’art. Celles-ci prennent appui sur des méthodes et outils tels que :

- les Réseaux de Petri, pour l’analyse des traces ;
- les modèles de mobilité, pour l’évaluation des performances ;
- l’introduction du social dans les services de localisation ;
- la mise en place d’une nouvelle métrique pour le routage.

Nous montrons comment ces solutions concourent de façon complémentaire les unes avec les autres, à améliorer l’expérience de l’utilisateur.

Mots clés : Réseaux sans fil, analyse de traces, modèles de mobilité, protocoles de routage, réseaux de Petri stochastiques, chaînes de Markov, aggrégation de données, métrique, traitement du signal.

Abstract

Future of communication is perceived as being almost exclusively composed of *mobile nodes* operating in a *mobile network*. In this context, different approaches contribute to continually improve directly or indirectly, delivery times of information exchanged between users, including :

- Trace analysis,
- Performance evaluation,
- Location services,
- Routing.

We describe each of these topics and propose solutions by changing the state of the art.

These are supported by tools and methods such as :

- Petri Nets for Trace analysis,
- Mobility Models for Performance evaluation,
- Social addings in Location services,
- The establishment of metrics for Routing.

We show how these solutions work together in a complementary manner with each other, to improve the user experience.

Keywords : Wireless networks, trace patterns, mobility models, performance evaluation, routing protocols, Petri nets, markov chains, metrics, signal processing.

Table des matières

Remerciements	i
Résumé	iii
Abstract	v
I Introduction	1
1 Introduction à la thèse	3
1.1 Introduction	3
1.2 Partie II : Mobilité	4
1.2.1 Profil de mobilité utilisateur	4
1.2.2 La mobilité sociale	5
1.3 Partie III : Applications	6
1.4 Partie IV : Mesures	6
II Mobilité	9
2 Introduction à la partie II	11
2.1 Introduction	11
2.2 Profil de mobilité utilisateur	11

2.3	Modèles de mobilité	12
3	Profil de mobilité utilisateur	15
3.1	Introduction	15
3.1.1	Définitions	15
3.1.2	Motivations	15
3.1.3	Présentation du chapitre	16
3.2	État de l’art	17
3.2.1	Prédictabilité	18
3.2.2	Prédiction	19
3.2.3	Maximisation de l’espérance	20
3.2.4	Filtrage	29
3.2.5	Chaînes de Markov à Temps Continu	30
3.3	Une nouvelle démarche	31
3.3.1	Introduction	31
3.3.2	Réseaux de Petri, Process mining, Algorithme- α	32
3.3.3	Lumpabilité	35
3.4	Implémentation	36
3.5	Conclusion	40
4	La mobilité sociale	41
4.1	Introduction	41
4.2	La notion du social	41
4.2.1	Introduction	41
4.2.2	Points d’Attraction (PA)	42
4.2.3	Modélisation	42
4.2.4	Application : Social Manhattan (SMN)	43

4.2.5	Conclusion	43
4.3	Les réseaux pondérés	44
4.3.1	Introduction	44
4.3.2	Points d'Attraction pondérés	45
4.3.3	Modélisation	45
4.3.4	Application : Weighted Manhattan (WMN)	46
4.3.5	Conclusion	46
4.4	Les systèmes de colonies de fourmis	47
4.4.1	Introduction	47
4.4.2	Modélisation	48
4.4.3	Application : Ant Manhattan (AMN)	48
4.4.4	Conclusion	49
4.5	Mixage de modèles	50
4.5.1	Introduction	50
4.5.2	Distribution des modèles	50
4.5.3	Configuration de test	50
4.5.4	Résultats	50
4.5.5	Interprétation	57
4.5.6	Conclusion	58
4.6	Conclusion	58
III	Applications	61
5	Introduction à la partie III	63
5.1	Introduction et motivation	63
5.2	Définition des services de localisation	64

TABLE DES MATIÈRES

5.3	Classification des services de localisation	66
5.4	DREAM	68
6	SFLS Social	71
6.1	Introduction	71
6.2	Semi-Flooding Location Service	72
6.3	S-SFLS : SFLS social	73
6.4	Analyse des performances	73
6.4.1	Distribution des nœuds	74
6.4.2	Évaluation d’SFLS sans communauté	75
6.4.3	Évaluation d’SFLS avec communauté	75
6.4.4	Impact des communautés sur SFLS	77
6.4.5	Évaluation de S-SFLS	77
6.4.6	Comparaison entre SFLS et S-SFLS	78
6.5	Simulation	78
6.5.1	Introduction	78
6.5.2	Évaluation d’SFLS sans communauté	79
6.5.3	Évaluation d’SFLS avec communauté	81
6.5.4	Évaluation de S-SFLS	82
6.6	Trajectoires modélisées	82
6.7	Conclusion	83
7	SLS Social	85
7.1	Introduction	85
7.2	SLS	86
7.3	Social	88
7.4	Evaluation d’S-SLS	90

7.4.1	Introduction	90
7.4.2	Simulation	91
7.4.3	Résultats	92
7.5	Conclusion	93
IV	Mesures	95
8	Introduction à la partie IV	97
8.1	Préambule	97
8.2	Introduction	97
8.3	Approche choisie	100
8.4	Aperçu de la partie	100
9	Mesures	103
9.1	Introduction	103
9.2	Modélisation	103
9.3	Fonction de coût	109
9.4	Conclusion	110
10	Optimisation	113
10.1	Introduction	113
10.2	Probabilité d'erreur	113
10.3	Probabilité de progression	114
10.4	Cas où $\mu/\sigma \leq 3$	116
10.5	Conclusion	116
11	Evaluation des mesures	119
11.1	Introduction	119

TABLE DES MATIÈRES

11.2 Configurations de test	120
11.3 Résultats	120
11.4 Conclusion	124
V Conclusions	127
12 Conclusions de la thèse	129
12.1 Partie II : Mobilité	129
12.2 Partie III : Applications	130
12.3 Partie IV : Mesures	132
12.4 Conclusions	133
12.5 Perspectives	134
Publications	137
13 Publications	137
Bibliographie	141
Annexes	153
A Implémentation de nos modèles de mobilité	153
A.1 SMN	153
A.1.1 Implémentation	153
A.1.2 Test	155
A.1.3 Résultats	159
A.2 Réseaux pondérés	163

A.2.1	Implémentation	163
A.2.2	Résultats	163
A.3	Les systèmes de colonies de fourmis	169
A.3.1	Résultats	169
B	Choix relatifs à la simulation et implémentation de nos modèles	177
B.1	Introduction	177
B.2	Présentation d' <i>Opnet Modeler</i>	178
B.3	Étude de l'existant	180
B.4	Nos premiers développements	195
B.5	Le problème des intersections manquées	195
B.6	Les 3 systèmes de coordonnées d' <i>Opnet Modeler</i>	196
B.6.1	Système de coordonnées global	196
B.6.2	Système de coordonnées relatif au sous-réseau	197
B.6.3	Système de coordonnées géocentrique	198
B.7	La cause du problème	199
B.8	La solution du problème	200
B.9	Création d'objet	200
B.9.1	Par duplication	201
B.9.2	Par dérivation	204
B.9.3	Création d'icône	207
B.9.4	Intégration d'icône et désignation standard	209
B.10	Attribut d'objet	211
B.10.1	Nom d'attribut	211
B.10.2	Modification	212
B.10.3	Création, suppression	218

B.11	Attribut étendu et XML ou limites de l'outil	220
C	Routage social	225
C.1	Introduction	225
C.2	Le protocole <i>OLSR</i>	227
C.3	<i>OLSR</i> et énergie : e-OLSR	229
C.4	<i>OLSR</i> social : OLSR-S	230
C.5	Application	231
C.6	Implémentation	232
C.7	Résultats	237
C.8	Conclusion	239
D	Implémentation d'OLSR-S	241
E	Outil de conversion de trajectoires Opnet vers NS2	253
E.1	Conversion des Positions de départ	253
E.2	Conversion des Trajectoires	255

Liste des tableaux

3.1	Fréquences connues $2 \leq i < 10$	21
3.2	Fréquences inconnues $0 \leq j < 2 - c$ =itération	22
3.3	Fréquences inconnues $0 \leq j < 2 - c$ =itération	24
3.4	Données de positionnement	29
3.5	Vitesse max des transports	30
3.6	Un journal de Workflow	36
4.1	Mixte Distribution	50
4.2	Simulation	51
4.3	Légende des graphiques perspective réseau	56
5.1	Avantages et limitations des services de localisation	68
6.1	Definition of parameters and variables.	75
6.2	Simulation	79
7.1	Simulation	91
11.1	Simulation	121
A.1	Simulation	159
A.2	Simulation	165
A.3	Simulation	171

Table des figures

3.1	Notre démarche par les Réseaux de Petri	17
3.2	Réseau de Petri Workflow-net (WF-net) avec 3 transitions parallèles	33
4.1	SLS versus Social SLS - Request/Reply transmitted - 50 nodes	44
4.2	Weighted Manhattan (WMN) : schéma de principe	47
4.3	Mouvement d'un nœud selon Ant Manhattan : Schéma fonctionnel	49
4.4	Load (Mbps)	52
4.5	Throughput (Mbps)	53
4.6	Delay (sec)	54
4.7	Data dropped (retry) (Mbps)	55
5.1	Location Service Big Picture	65
5.2	Classification des services de localisation	66
5.3	DREAM : Distance Effect and Forwarding Cone	70
6.1	Propagation of location packets.	72
6.2	SFLS algorithms.	73
6.3	S-SFLS algorithm.	74
6.4	Distribution of neighbours around a node.	76
6.5	Factor reduction vs. maximum path length for SFLS.	77
6.6	Factor reduction vs. maximum path length with communities.	78

TABLE DES FIGURES

6.7	SFLS - One Group - Pedestrian (0-2 mps) - 50/100 nodes	80
6.8	SFLS - One Group - Car (0-15 mps) - 50/100 nodes	81
6.9	SFLS - 4 Groups - Pedestrian (0-3 mps)/Car (0-15 mps) - 50 nodes	81
6.10	S-SFLS - 4 Groups - Pedestrian (0-3 mps)/Car (0-15 mps) - 50 nodes	82
7.1	SLS versus Social SLS - Request/Reply transmitted - 50 nodes	92
7.2	SLS versus Social SLS - Request/Reply received - 50 nodes	93
9.1	Jointure entre η, d et ensemble $\mathcal{D} = \{(\eta, d) z > \bar{z}\}$ qui définit la probabilité d'arrivée effective de l'information au prochain relais	108
10.1	Error probability versus progress for 5 relays. Dot-line for $\sigma_\eta = 3$, full-line for $\sigma_\eta = 1$. Squared for Deterministic decision and circle for stochastic one-hop decision. $\sigma = 210\text{ m} \Rightarrow \mu/\sigma < 3$	116
10.2	Error probability versus progress for 10 relays. Dot-line for $\sigma_\eta = 3$, full- line for $\sigma_\eta = 1$. Square for deterministic decision and circle for stochastic one-hop-decision.	117
11.1	DREAM : cône d'émission	119
11.2	Control transmitted - 50/100 nodes	121
11.3	Control received - 50/100 nodes	122
11.4	Data transmitted - 50/100 nodes	122
11.5	Data received - 50/100 nodes	123
11.6	Total transmitted - 50/100 nodes	123
11.7	Total received - 50/100 nodes	124
11.8	End-to-end transmitted - 50/100 nodes	124
11.9	End-to-end received - 50/100 nodes	125
11.10	Delivery ratio - 50/100 nodes	125
11.11	Number of hops - 50/100 nodes	126

11.12	End-to-end delay - 50/100 nodes	126
A.1	Attributs Mobility Config Ext	155
A.2	Nouveaux objets visuels <i>Opnet Modeler</i>	156
A.3	Attributs Sociaux	156
A.4	SMN - Variables statistiques réseau	160
A.5	WMN - Variables statistiques réseau	166
A.6	AMN - Variables statistiques réseau	172
B.1	OPNET Modeler	178
B.2	Notre version d'OPNET Modeler	179
B.3	Mentions légales OPNET Modeler à destination des Académies	179
B.4	L'Interface Homme-Machine (IHM) d' <i>OPNET Modeler</i>	180
B.5	Projet Exemple Random_mobility d'Opnet Modeler	182
B.6	Editeur d'attributs de l'objet Mobility Config	183
B.7	Éditeur d'attributs avancé de l'objet Mobility Config	187
B.8	Mise en garde concernant les modifications de modèle d'objet <i>Opnet</i>	188
B.9	Implementation de Manhattan dans Opnet Modeler	195
B.10	Le système de coordonnées global d'Opnet Modeler	197
B.11	Le système de coordonnées relatif au sous-réseau d'Opnet Modeler	198
B.12	Le système de coordonnées géocentrique d'Opnet Modeler	199
B.13	Erreur de conversion dans Opnet Modeler X 1000	200
B.14	Boîte de dialogue Icon View d'Opnet Modeler	202
B.15	Boîte de dialogue Configure Palette d'Opnet Modeler	202
B.16	Boîte de dialogue Select Included Entries d'Opnet Modeler	203
B.17	Configure Palette après ajout de l'objet mobility_config_ext	204
B.18	Configure Palette après ajout de l'objet mobility_config_ext	205

TABLE DES FIGURES

B.19 Boîte de dialogue Derive New Node Model d'Opnet Modeler	205
B.20 Boîte de dialogue Edit Derived Model d'Opnet Modeler	206
B.21 Boîte de dialogue Icon Palette d'Opnet Modeler	206
B.22 Boîte de dialogue Image Editor d'Opnet Modeler	208
B.23 Boîte de dialogue Color Palette Editor d'Opnet Modeler	208
B.24 Icône distinctive de notre nouvel objet <i>Mobility Config Ext</i>	209
B.25 Fenêtre de l'éditeur de modèle de nœud d'Opnet	213
B.26 Fenêtre de l'éditeur de processus d'Opnet	214
B.27 Boîte de dialogue Model Attributes Initiale de mobility_config_ext	214
B.28 Boîte de dialogue Model Attributes de Mobility Config Ext	215
B.29 Modification du code de l'état Init du processus Mobility Config Ext	215
B.30 Boîte de dialogue de compilation réussie	216
B.31 Boîte de dialogue Rename/Merge Attributes : Mobility Config Ext	217
B.32 Boîte de dialogue Random Mobility Definition Properties	218
B.33 Boîte de dialogue Random Mobility Definition Properties	219
C.1 Gain de retransmission 50 % : 12 au lieu de 24 : (a) sans MPR, (b) avec MPR	228
C.2 2 mécanismes d'optimisation énergétique utilisant OLSR	230

Première partie

Introduction

Chapitre 1

Introduction à la thèse

1.1 Introduction

Ce document rend compte d'un travail auquel nous avons consacré de nombreuses années de recherche. Une recherche intense, une recherche passionnée car faite en collaboration avec des personnes passionnées, j'ai cité le Pr Kamel Barkaoui, le Dr Selma Boumerdassi, le Pr Ruben Milocco pour ne citer que ceux qui nous ont formés et élevés au niveau auquel modestement nous sommes aujourd'hui. Ce, malgré toutes les difficultés que traverse la recherche actuellement en France, en Europe ou au niveau mondial. Mais, la passion l'emporte et l'emportera encore de nombreuses années. En effet, si la Recherche avec un grand 'R' devient la propriété de quelques uns ou quelques unes, la liberté de pensée qui est la condition *sine qua non* d'une recherche authentique et sérieuse, n'est plus satisfaite et tous les efforts d'innovation au sens le plus noble, au sens le plus humain, sont vains, car cantonnés, bornés, aux seuls desideratas de cette minorité qui payent. Au contrario, lorsqu'elle est payée par tous les contribuables français, européens ou mondiaux, au prorata de leur ressources, la Recherche est française, européenne, mondiale. Elle devient le rêve des français, des européens, de tous les hommes. « You may say I'm a dreamer, but I'm not the only one », chantait qui vous savez. Rêver est sans doute la plus grande richesse de l'Homme. Rêver l'Homme est le plus beau de tous les rêves, peut-être. Il est, sans aucun doute, notre définition de la Recherche.

Le domaine des réseaux sans fil est depuis de nombreuses années notre domaine de prédilection. Les aspects sociaux occupant depuis quelques années déjà et de plus en plus

le devant de la scène de la Recherche, il était naturel (c'est d'ailleurs un de nos objets de recherche : les points d'attraction) que nous nous y intéressions également. D'où la naissance de cet intéressement au social appliqué à notre domaine qui a donné naissance à des modèles sociaux traitant de la mobilité dans les réseaux sans fil. Les points forts de cette thèse sont les suivants :

- utilisation des *Réseaux de Petri* dans l'élaboration du *Profil de Mobilité d'un Utilisateur* ;
- *Modélisation Sociale de la Mobilité* pour l'évaluation des performances ;
- introduction du *Social* dans 2 *Services de Localisation* ;
- mise en place d'une *Metric* permettant, lors de l'acheminement d'une donnée, de choisir le *prochain nœud*.

En première partie, les mobilités individuelle et groupale sont définies, leurs états de l'art exposés et des propositions de modélisation implémentées tantôt sous la forme de pseudo-code, tantôt sous la forme de code proto-C.

En deuxième partie, les services de localisation sont choisis comme cadre applicatif aux notions du social. Leurs implémentations en C++ sont offertes à la communauté scientifique afin qu'elle puisse juger de notre approche et poursuivre dans ce sens, le cas échéant.

En troisième partie, une démarche en matière de mesure de la qualité du lien internodal dans un réseau mobile, nous est apparue d'une pertinence particulière et nous l'avons étudiée et améliorée avec son auteur. En outre, dans le cadre de travaux futurs, nous envisageons de poursuivre dans cette voie en transférant ce savoir à l'intégration dans les couches basses protocolaires des réseaux mobiles, d'aspects sociaux modélisés sous formes de courbes gaussiennes.

1.2 Partie II : Mobilité

1.2.1 Profil de mobilité utilisateur

La notion de *Profil de Mobilité Utilisateur* possède différentes définitions. D'aucuns la définissent comme « l'information qui transforme votre métier » (Vodafone [2012]) ! D'autres plus proches de nous tels que (Bayir et al. [2009]), la définissent ainsi : « A

mobility profile for a cellphone user includes personal mobility patterns with contextual time data and distribution of spatiotemporal locations for that user. ». Notre définition est la suivante et se veut plus conceptuelle : « La notion de profil de mobilité d'un utilisateur de téléphone cellulaire couvre la modélisation de ses schémas personnels de mobilité assortis d'éléments temporels contextualisés et de la distribution spatio-temporelle de ses lieux de fréquentation. ».

Être capable de caractériser le profil des utilisateurs de téléphone portable est un défi crucial à plus d'un titre.

En premier lieu, les *protocoles de routage* des réseaux mobiles sont les clients par essence de ce type d'information. En effet, tester un protocole à l'aide de modèle de déplacement aléatoire et les tester à l'aide de déplacement proche des déplacements humains a permis de mettre en évidence des écarts de plus de 12 % de moins sur les performances annoncées par les constructeurs (Costantini et al. [2011]).

Le suivi des épidémies (Colizza and Vespignani [2008]), celui des personnes vulnérables telles les adolescents (Wiehe et al. [2008]) sont d'autres champs applicatifs de ce type d'information.

Mais il y a aussi la *personnalisation des services* qui, outre les profils utilisateurs construits par l'enregistrement et la fouille des clics sur les pages web, peut profiter également des profils de déplacement de ces mêmes utilisateurs (Kleinhans [2010]).

Ce chapitre après son *état de l'art*, expose notre proposition en matière de détermination du *Profil de Mobilité Utilisateur* ainsi que notre implémentation.

1.2.2 La mobilité sociale

Un modèle de mobilité est destiné à décrire, en termes d'environnement, le mode de circulation des nœuds mobiles. Le défi est de trouver des modèles qui reproduisent le plus fidèlement les comportements réels des utilisateurs. Ce chapitre présente de nouveaux modèles de mobilité sociaux, et montre leurs bénéfices par rapport aux modèles existants. Le focus est mis sur le fait qu'en se rapprochant des comportements sociaux au travers de l'utilisation de la notion de Points d'Attraction, cela change considérablement les ré-

sultats des tests de performances des protocoles de routage des réseaux mobiles. Dans ce chapitre, nous traitons tout d'abord des limites des modèles de mobilité existants et les classifions. Ensuite, nous définissons nos propres modèles sociaux et les implémentons afin d'en mesurer l'impact sur les performances d'un réseau mobile simulé. Nous montrons l'influence de ces critères et l'importance d'un choix adéquat du modèle de mobilité utilisé dans l'environnement de test des réseaux de communication.

1.3 Partie III : Applications

La localisation géographique des nœuds (i.e. connaître les coordonnées d'un nœud) a été proposée à des fins d'adressage. Les protocoles de routage basés sur la localisation géographique, utilisent cette dernière dans leur prise de décision de routage. Ses performances sont donc fortement liées à celles de son service de localisation. Les services de localisation sont présentés en suivant le plan suivant :

- définitions
- propriétés essentielles
- classifications.

C'est l'actualité et la pertinence de ce domaine qui nous l'ont fait choisir pour démontrer l'apport de la prise en compte des aspects sociaux dans les réseaux mobiles.

1.4 Partie IV : Mesures

Cette partie est un peu atypique par rapport aux autres parties. En effet, les autres traitent des aspects humains individuels ou groupaux qui interviennent dans la prédiction des déplacements. La présente partie en revanche propose une approche novatrice de la mesure de la qualité du lien internodal dans les réseaux sans fil.

L'idée consiste à adapter des éléments de la mathématique du signal à la mesure de cette qualité. Ces éléments ont pour base commune, la nature gaussienne de l'incertitude des mesures élémentaires qui entrent dans le calcul de cette mesure de qualité. Toutefois, la démarche est suffisamment généraliste pour nous permettre d'envisager de l'utiliser à d'autres fins. Plus précisément, une adaptation semblable à celle décrite dans le présent

chapitre, nous apparaît applicable à l'intégration dans nos modèles de mobilité, de, par exemple, des phénomènes sociaux gaussiens tels que Isaksson [2009]. Ce dernier montre qu'il est possible de prévoir le candidat qui aura la faveur de l'électorat par une approche gaussienne. Rien ne s'oppose donc a priori à utiliser ses résultats à la prédiction des choix de déplacement de groupes, dans nos modèles de mobilité.

Deuxième partie

Mobilité

Chapitre 2

Introduction à la partie II

2.1 Introduction

Cette partie concerne la mobilité. Elle commence par la mobilité individuelle et finit par la mobilité de groupe. La mobilité individuelle est vue sous l'angle de l'analyse des traces issues des infrastructures de géolocalisation. La mobilité de groupe, a contrario, part de la théorie du social en passant par la modélisation pour aboutir aux traces. Ces démarches sont complémentaires l'une de l'autre mais sont rarement rapprochées de cette façon dans la littérature. En effet, en règle générale, les résultats sont issus soit de l'analyse de traces soit de la modélisation. Dans cette thèse, on trouvera des résultats de l'une, de l'autre voire des deux démarches conjointes.

2.2 Profil de mobilité utilisateur

Afin de définir la notion de *Profil de Mobilité Utilisateur*, nous partons tout d'abord de travaux récents (Bayir et al. [2009]) qui proposent ceci : « A mobility profile for a cellphone user includes personal mobility patterns with contextual time data and distribution of spatiotemporal locations for that user. ». Ensuite, nous traduisons librement en ajoutant notre touche personnelle comme suit : « La notion de profil de mobilité d'un utilisateur de téléphone cellulaire couvre la modélisation de ses schémas personnels de mobilité assortis d'éléments temporels contextualisés et de la distribution spatio-temporelle de ses lieux de fréquentation. ».

Cette définition étant posée, nous soulignons l'aspect crucial que revêt le profil des utilisateurs de téléphone portable.

Les clients par essence de ce type d'information, sont, au premier chef, les *protocoles de routage* des réseaux mobiles. Car, tester un protocole à l'aide de déplacement proche des déplacements humains, transforme les résultats avec des écarts de plus de 12 % de moins sur les performances annoncées par les constructeurs (Costantini et al. [2011]), par rapport au fait de le tester à l'aide de modèle de déplacement aléatoire.

Le suivi des catégories de personnes vulnérables telles les adolescents (Wiehe et al. [2008]) ou le suivi des épidémies (Colizza and Vespignani [2008]) sont autant d'autres applications possibles de cette d'information.

La *personnalisation des services*, outre les profils utilisateurs construits par l'enregistrement et la fouille des clics sur les pages web, peut bénéficier également des profils de déplacement de ces mêmes utilisateurs (Kleinhans [2010]).

Ce chapitre après son *état de l'art*, expose notre proposition en matière de détermination du *Profil de Mobilité Utilisateur* ainsi que notre implémentation.

2.3 Modèles de mobilité

À leurs débuts, les réseaux de télécommunications étaient pensés de manière statique. À l'aire du *Multimedia Mobile*, l'essor des réseaux sans fil comme réponse des opérateurs de télécommunications aux besoins de mobilité des utilisateurs est fulgurant. Un modèle de mobilité est destiné à décrire, en termes d'environnement, le mode de circulation des nœuds mobiles. Le défi est de trouver des modèles les plus fidèles possibles des comportements réels des utilisateurs. Les appareils mobiles étant portés par des utilisateurs qui ont un comportement social, des habitudes etc., cela nous a amené tout naturellement à l'idée d'inclure des éléments de cette composante sociale dans la modélisation de la mobilité. Ce chapitre présente de nouveaux modèles de mobilité sociaux, et montre leurs apports par rapport aux modèles existants. Le focus est mis sur le fait qu'en se rapprochant des comportements sociaux au travers de l'utilisation de la notion de Points d'Attraction, cela change considérablement les résultats des tests de performances des protocoles de routage

des réseaux mobiles. Dans ce chapitre, nous traitons tout d'abord des limites des modèles de mobilité existants et les classifions. Ensuite, nous définissons nos propres modèles sociaux et les implémentons afin d'en mesurer l'impact sur les performances d'un réseau mobile simulé. Nous montrons l'influence de ces critères et l'importance d'un choix adéquat du modèle de mobilité utilisé dans l'environnement de test des réseaux de communication.

Chapitre 3

Profil de mobilité utilisateur

3.1 Introduction

3.1.1 Définitions

Définie par certains comme « l'information qui transforme votre métier » (Vodafone [2012]) et par d'autres plus proches de nous (Bayir et al. [2009]) par : « A mobility profile for a cellphone user includes personal mobility patterns with contextual time data and distribution of spatiotemporal locations for that user. », nous définissons la notion de *Profil de Mobilité Utilisateur* de façon plus conceptuelle en ces termes : « La notion de profil de mobilité d'un utilisateur de téléphone cellulaire couvre la modélisation de ses schémas personnels de mobilité assortis d'éléments temporels contextualisés et de la distribution spatio-temporelle de ses lieux de fréquentation. ».

3.1.2 Motivations

Être capable de caractériser le profil des utilisateurs de téléphone portable est un défi crucial à plus d'un titre.

Tout d'abord, les *protocoles de routage* des réseaux mobiles sont les premiers concernés par ce type d'information. En effet, le test et la vérification d'un protocole à l'aide d'un modèle aléatoire n'a rien à voir avec le fait de le tester à l'aide de déplacements proches des déplacements humains. Cela a permis de relever des performances de l'ordre de 12 % inférieurs à celles annoncées par les constructeurs (Costantini et al. [2011]).

Suivre les épidémies (Colizza and Vespignani [2008]), les personnes vulnérables telles que les adolescents (Wiehe et al. [2008]) sont également des clients privilégiés de ce type d'information.

Mais, il y a aussi la *personnalisation des services* qui, outre les profils utilisateurs construits par l'enregistrement et la fouille des clics sur les pages web, peut profiter également des profils de déplacement de ces mêmes utilisateurs (Kleinhans [2010]).

3.1.3 Présentation du chapitre

La suite de ce chapitre suit le découpage suivant :

- un exposé de l'*état de l'art* dans la section 3.2 où nous voyons que les utilisateurs de téléphones portables ont des déplacements qui sont prévisibles dans 93 % des cas de figure. Ce résultat surprenant de Song et al. [2010], mérite qu'on s'y attarde. Les auteurs sont partis d'une base de données issue d'un opérateur de téléphonie mobile comprenant à l'origine de l'ordre de 50000 utilisateurs. Et, quoique 5000 utilisateurs n'utilisent pas suffisamment leurs téléphones portables pour être représentatifs et sont donc exclus d'office, c'est malgré tout, une base de 45000 utilisateurs « utiles », qui a servi à établir ce résultat et en ce sens est à même de l'asseoir. Une centaine d'utilisateurs utilisent leur téléphone au moins 1 fois par heure durant les 24 heures que comporte la journée. Ceux-ci ont été utilisés par les auteurs pour mettre au point une méthode d'extrapolation permettant de pallier à l'absence de communication de plusieurs heures d'affilées qui concerne la plupart des personnes de cette base de données ;
- notre proposition en la matière dans la section 3.3 qui, à l'approche décrite ci-dessus, apporte un autre regard consistant en une nouvelle démarche qui met en œuvre les *Réseaux de Petri*, *RdP* Petri [1966], cf. Fig. 3.1 :
 1. Nous partons des traces des déplacements de l'utilisateur (GPS, GLONASS, Galileo, WPS, 3G...) ;
 2. À partir de ces fichiers de journalisation *log file*, nous appliquons l'*algorithme- α* Van Der Aalst and Van Dongen [2002] pour découvrir les schémas récurrents *patterns* de mobilité de l'utilisateur pour aboutir ainsi à une représentation en

termes de Workflow Nets, une classe particulière de *Réseau de Petri*, *RdP*;

3. En attribuant une temporisation stochastique aux transitions (loi de distribution exponentielle, par exemple), nous obtenons un *Réseau de Petri Stochastique*, *RdPS*, Marsan [1995], dont le graphe des marquages accessibles est isomorphe à une *Chaîne de Markov à Temps Continue*, *CMTC*;
4. Nous pouvons alors effectuer tous les types d'évaluation permis par les chaînes de Markov.

- notre implémentation en section 3.4
- une conclusion en section 3.5

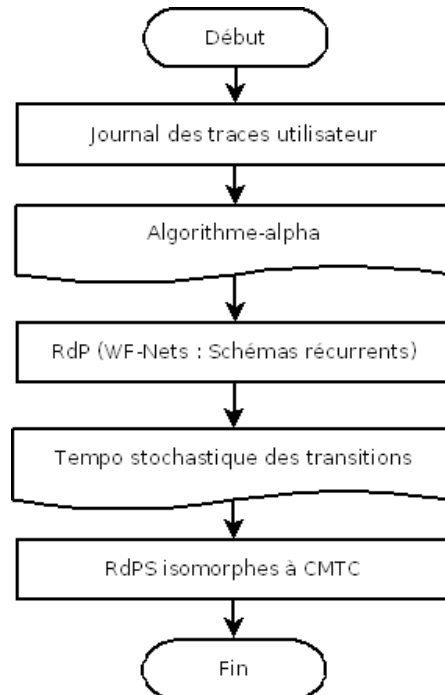


FIGURE 3.1 – Notre démarche par les Réseaux de Petri

3.2 État de l'art

Pour modéliser un *Profil de Mobilité*, la littérature mentionne différentes étapes que sont :

- la méthode dite de *Maximisation de l'espérance*;

- le *Filtrage* ;
- les Chaînes de Markov à Temps Continu.

C'est pourquoi nous traitons de ces différents sujets dans la présente section, dans une optique « état de l'art ».

En prélude, nous présentons une étude détaillée d'un travail concernant la *prédictabilité* dont les résultats sont fort surprenants et prometteurs, dans la section 3.2.1.

Ensuite, fort de ces résultats, nous traitons de la *prédiction* en elle-même, dans la section 3.2.2, en nous appuyant sur une autre étude récente (Kim and Song [2011]) qui nous permet de développer les sujets mentionnés ci-dessus :

- la *Maximisation de l'espérance* dans la section 3.2.3 ;
- le *Filtrage* dans la section 3.2.4 ainsi que
- les Chaînes de Markov à Temps Continu, dans la section 3.2.5.

3.2.1 Prédictabilité

Comme mentionné dans l'introduction du présent chapitre, les déplacements des utilisateurs de terminaux portables sont prévisibles dans 93 % des cas de figure (Song et al. [2010]). Précisons, car c'est primordial, que dire qu'un déplacement est prévisible ne signifie en rien qu'on est en mesure de le prévoir précisément dans les faits. La différence est de taille et témoigne de la distance qu'il y a entre *prédictabilité* et *prédiction*, dont nous parlons dans la section suivante.

Rappelons les faits. La base de données que les auteurs utilisent est issue d'un opérateur de téléphonie mobile et comprend à l'origine de l'ordre de 50000 utilisateurs. Quoique 5000 utilisateurs n'utilisent pas suffisamment leurs téléphones portables pour être représentatifs et sont donc exclus d'office, c'est malgré tout, une base de 45000 utilisateurs « utiles », qui a servi à établir ce résultat et en ce sens est à même de l'asseoir. Une centaine d'utilisateurs utilisent leur téléphone au moins 1 fois par heure durant les 24 heures que comporte la journée. Ceux-ci ont été utilisés par les auteurs pour mettre au point une méthode d'extrapolation permettant de pallier à l'absence de communication de plusieurs heures d'affilées qui concerne la plupart des personnes de cette base de données.

Cette base sans « données manquantes » donc, permet une analyse statistique des déplacements à l'aide d'une entropie S qui mesure non seulement la distribution des passages de l'utilisateur dans les cellules du réseau téléphonique mobile de l'opérateur considéré, prises séparément les unes des autres, mais aussi et surtout, l'ordre dans lequel les antennes ont enregistré le passage de l'utilisateur. Cet ordre nous renseigne sur les répétitions spatio-temporelles des enchaînements de déplacements. En parcourant la base à l'aide de cette entropie plutôt qu'une autre entropie indépendante de l'ordre (qu'elle soit atemporelle S^{unc} ou aléatoire i.e. uniforme S^{rand}), il a été constaté que l'on passe de 6 bits à 0,8 bit ! En d'autres termes, le degré d'incertitude sur le lieu passe de 64 à 1,74 lieux !

Fort de cette entropie spatio-temporelle S et du nombre N de lieux visités par l'utilisateur, l'on calcule $\Pi^{max}(S, N)$, Fano [1961], qui n'est autre que la probabilité de prédictibilité, c'est-à-dire la probabilité de trouver (et non pas le fait de trouver !) un algorithme qui parvienne à prévoir les déplacements de l'utilisateur considéré. Il va de soi que plus S est petit plus $\Pi^{max}(S, N)$ est grand. Or, compte tenu du fait que $S = 0,8$, les auteurs affichent $\Pi^{max}(S, N) = 0,93$ qui n'est autre que ces 93 % avec lesquels nous avons ouvert ce chapitre !

3.2.2 Prédiction

À nouveau, dire qu'un déplacement est prévisible est une chose, c'est l'objet de la section précédente, prévoir le déplacement en question en est une autre, c'est l'objet de la présente section et le passage d'un sujet à l'autre nous est facilité par Kim and Song [2011]. Contrairement à Song et al. [2010] qui utilise une base de données issue d'un opérateur de téléphonie mobile, le choix de Song and Kim [2012] a porté sur les données de géolocalisation mobile présentes dans les téléphones portables des auteurs. Celles-ci sont de 3 ordres : GPS, WPS (Wi-Fi) et 3G. La démonstration est faite que WPS et 3G ne sont pas suffisamment mûrs et le choix de GPS s'impose très vite aux auteurs. L'utilisation de l'algorithme de *Maximisation de l'espérance* ou *Expectation Maximisation* permet la mise en grappes (*clusterisation*) des données selon une stratégie s'inspirant notamment du *Levy Walk* (Rhee et al. [2008]). La clusterisation effectuée, les auteurs proposent récemment 2 extensions (Kim and Song [2012]) :

- la formalisation du modèle de déplacement d'un utilisateur sous forme de chaîne de markov à temps continu (CMTC) ;
- l'utilisation de la fonction de densité de probabilité des grappes, pour établir entre elles une relation d'inclusion (grappe/sous-grappe) menant à une notion nouvelle dénommée par les auteurs *micro-mobilité*.

Enfin, un aspect qui peut apparaître anodin mais qui a cependant une importance qui va, à notre sens, aller grandissante, consiste en le fait d'arriver *automatiquement* à rattacher les données géolocatives à des lieux physiques connus : le domicile, le lieu de travail, l'université, la laboratoire. . . Les auteurs effectuent la correspondance manuellement (utilisation de Google Map[®]) mais n'ont encore rien proposé du point de vue de l'automatisation.

3.2.3 Maximisation de l'espérance

Les traces, qu'il s'agisse de base de données d'opérateur de téléphonie mobile ou d'un système global de géolocalisation (GPS, GLONASS, Galileo, WPS, 3G. . .), comportent un élément qui les différencient particulièrement des données statistiques courantes : elles sont très souvent incomplètes pour des causes diverses (absence d'appel téléphonique durant des longues heures, erreur sur les données géolocatives. . .).

Dans ce contexte, la *Maximisation de l'espérance* ou *Expectation Maximization* est très présente.

Les pères de cette méthode, Hartley [1958] et Dempster et al. [1977] ont marqué leur génération mais les travaux de Gupta and Chen [2011] sur des cas plus récents nous permettent d'accéder à ce savoir.

Étude de cas : « Fougères »

Hartley [1958], après avoir pris soin de citer des démarches concurrentes du maximum de vraisemblance More [1952], introduit le sujet en citant l'exemple de Snedecor [1956] qui traite de la pollution de graines de fougères (*Phleum pratense*) par la présence de quelques graines nocives de mauvaises herbes.

La Table 3.1 fait état des fréquences connues de ces graines nocives dans 78 lots de graines de fougère. La colonne $x = i$ donne la fréquence ; par exemple la ligne 2 nous renseigne sur le nombre de lots où l'on a trouvé 2 graines nocives. Les colonnes n/a sont « non applicables », c'est-à-dire qu'elles ne sont pas utiles pour le moment, mais le deviendront dans les tables suivantes ; leur présence permet d'homogénéiser la présentation des tables et permettre un rapprochement plus aisé de celles-ci. La colonne n_i donne, pour chaque fréquence donnée, le nombre de lots concernés. Dans la suite, nous noterons : $n = \sum n_i$. La colonne in_i donne, pour chaque fréquence donnée, le nombre total de graines nocives.

TABLE 3.1 – Fréquences connues $2 \leq i < 10$

$x = i$	n/a	n/a	n_i	in_i
2			26	52
3			16	48
4			18	72
5			9	45
6			3	18
7			5	35
8			0	0
9			1	9
Total			78	279

Le but de la méthode est d'approximer de façon itérative l'ensemble des fréquences inconnues : cf. Tables 3.2, 3.3.

La Table 3.2 donne la toute première approximation de ces fréquences inconnues où $c = 0$ représente le numéro de l'itération 0.

Remarque importante : cette première estimation n'en est pas vraiment une puisqu'elle est fournie par le statisticien de manière grossière, comme point de départ, d'où la nécessité

3.2. ÉTAT DE L'ART

de choisir avec soin ces valeurs de départ.

La colonne $x = j$ donne la fréquence inconnue considérée ; par exemple la ligne 0 nous renseigne sur le nombre de lots où l'on n'a pas trouvé de graines nocives. Nous trouvons de nouveau des colonnes n/a qui sont toujours « non applicables », c'est-à-dire qu'elles ne sont pas utiles pour le moment, mais le deviendront dans les tables suivantes. La colonne ${}_cn_j$ donne pour l'itération c le nombre de lots contenant j graines nocives. La colonne $j{}_cn_j$ donne pour l'itération c et la fréquence j , le nombre de graines nocives.

TABLE 3.2 – Fréquences inconnues $0 \leq j < 2 - c$ = itération

$c = 0$				
$x = j$	n/a	n/a	${}_cn_j$	$j{}_cn_j$
0			4	0
1			14	14
Total			18	14

Dans la Table 3.3, la colonne $x = j$ donne la fréquence inconnue considérée ; par exemple la ligne 0 nous renseigne sur le nombre de lots où l'on n'a pas trouvé de graines nocives. Ce tableau donne la toute première approximation de ces fréquences inconnues où $c = 0$ représente le numéro de l'itération 0. *Cette première estimation n'en est pas vraiment une puisqu'elle est fournie par le statisticien de manière grossière, comme point de départ, d'où la nécessité de choisir avec soin ces valeurs de départ.* Nous trouvons de nouveau des colonnes n/a qui sont toujours « non applicables », c'est-à-dire qu'elles ne sont pas utiles pour le moment, mais le deviendront dans les tables suivantes. La colonne ${}_cn_j$ donne pour l'itération c le nombre de lots contenant j graines nocives. La colonne $j{}_cn_j$ donne pour l'itération c et la fréquence j , le total de graines nocives.

Ensuite, les approximations suivantes des fréquences inconnues se font en calculant à chaque itération c :

- la moyenne ${}_c\theta$ des fréquences connues, d'une part, et, d'autre part, des fréquences inconnues précédemment approximées ;
- la probabilité pour que x soit égale à j qui est notée $f(j, {}_c\theta)$ et qui suit une distribution de Poisson de paramètre égal à cette moyenne ;
- la nouvelle approximation des fréquences inconnues : ${}_cn_j$.

Les formules de calcul sont les suivantes :

$$\begin{aligned} {}_c n' &= \sum {}_c n_j & {}_c \theta &= \frac{\sum i n_i + \sum j {}_c n_j}{n + {}_c n'} \\ f(j, {}_c \theta) &= \frac{{}_c \theta^j}{j!} e^{-{}_c \theta} & f({}_c \theta) &= \sum f(j, {}_c \theta) \\ {}_c n_j &= n \frac{f(j, {}_c \theta)}{1 - f({}_c \theta)} \end{aligned}$$

Le calcul de ${}_c n_j$ inclut un facteur particulier qu'il convient d'expliquer, à savoir le facteur d'*allocation proportionnelle* : $\frac{n}{1 - f({}_c \theta)}$.

Le détail en est, à notre sens, le suivant :

$$100 \times \frac{n}{100} \times \frac{1}{1 - f({}_c \theta)}.$$

En effet, $f({}_c \theta)$ est la somme des probabilités de présence de graines nocives, donc il s'agit d'une probabilité.

$1 - f({}_c \theta)$ représente la probabilité des i vue des j , en quelque sorte.

Si bien qu'il est nécessaire d'obtenir le rapport :

$$\frac{\text{proba_reelle_des_}i}{\text{proba_estimee_des_}i_par_les_j}.$$

C'est pourquoi, l'on doit calculer $\frac{n}{100}$, la probabilité réelle des i .

Enfin, il est nécessaire de repasser en pourcentage car l'expérience concerne 100 lots.

En ce qui concerne la moyenne ${}_c \theta$, la seule valeur renseignée de cette colonne est la ligne « Total », bien qu'il ne s'agisse pas d'un total ; nous l'affichons là par simple commodité.

Pour finir, il est intéressant de souligner que Hartley [1958] ne décrit que les itérations 0 à 2 puis nous dit qu'il suffit de 2 autres itérations ou presque pour arriver à l'estimateur maximum avec 3 décimales à savoir ${}_c \theta = 3,026$. Effectivement, nos calculs montrent qu'il faut attendre en fait encore 2 autres itérations (la 6 donc) pour obtenir cette approximation de ${}_c \theta$ à 3 décimales, comme le montrent les itérations suivantes (7 et 8).

Étude de cas : « Animaux »

Dempster et al. [1977] commencent également par une illustration issue du fils spirituel du père de la méthode du maximum de vraisemblance Ronald Aylmer Fisher, Rao [1965],

TABLE 3.3 – Fréquences inconnues $0 \leq j < 2 - c$ =itération

$x = j$	$c\theta$	$f(j, c\theta)$	$c n_j$	$j c n_j$
$c = 1$				
0		0,0473	4,559	0,000
1		0,1442	13,916	13,916
Total	3,052	0,1915	18,475	13,916
$c = 2$				
0		0,0480	4,646	0,000
1		0,1458	14,106	14,106
Total	3,036	0,1938	18,751	14,106
$c = 3$				
0		0,0483	4,683	0,000
1		0,1464	14,186	14,186
Total	3,029	0,1948	18,869	14,186
$c = 4$				
0		0,0485	4,699	0,000
1		0,1467	14,221	14,221
Total	3,0266	0,1952	18,919	14,221
$c = 5$				
0		0,0485	4,705	0,000
1		0,1468	14,235	14,235
Totals	3,0254	0,1954	18,941	14,235
$c = 6$				
0		0,0486	4,708	0,000
1		0,1469	14,242	14,242
Totals	3,0249	0,1955	18,950	14,242
$c = 7$				
0		0,0486	4,709	0,000
1		0,1469	14,244	14,244
Totals	3,0247	0,1955	18,954	14,244
$c = 8$				
0		0,0486	4,710	0,000
1		0,1469	14,246	14,246
Totals	3,0246	0,1955	18,956	14,246

concernant 197 animaux distribués selon une loi multinomiale dans 4 catégories,

$$y = (125, 18, 20, 34) = (y_1, y_2, y_3, y_4)$$

Un modèle génétique spécifie leurs probabilités par catégorie, comme suit :

$$\left[\left(\frac{1}{2} + \frac{1}{4}\pi \right), \left(\frac{1}{4} - \frac{1}{4}\pi \right), \left(\frac{1}{4} - \frac{1}{4}\pi \right), \left(\frac{1}{4}\pi \right) \right]$$

où $0 \leq \pi \leq 1$. Il vient donc, bien entendu :

$$g(y|\pi) = \frac{(y_1 + y_2 + y_3 + y_4)!}{y_1!y_2!y_3!y_4!} \left(\frac{1}{2} + \frac{1}{4}\pi \right)^{y_1} \left(\frac{1}{4} - \frac{1}{4}\pi \right)^{y_2} \left(\frac{1}{4} - \frac{1}{4}\pi \right)^{y_3} \left(\frac{1}{4}\pi \right)^{y_4}$$

De là nous construisons y , un échantillonnage incomplet, tel que :

$$y1 = x1 + x2, \quad y2 = x3, \quad y3 = x4, \quad y4 = x5$$

où $x1$ et $x2$ sont inconnues et dont les probabilités sont :

$$\left[\left(\frac{1}{2} \right), \left(\frac{1}{4}\pi \right), \left(\frac{1}{4} - \frac{\pi}{4} \right), \left(\frac{1}{4} - \frac{\pi}{4} \right), \left(\frac{1}{4}\pi \right) \right]$$

D'où l'échantillonnage complet suivant :

$$f(x|\pi) = \frac{(x_1 + x_2 + x_3 + x_4 + x_5)!}{x_1!x_2!x_3!x_4!x_5!} \left(\frac{1}{2} \right)^{x_1} \left(\frac{1}{4}\pi \right)^{x_2} \left(\frac{1}{4} - \frac{1}{4}\pi \right)^{x_3} \left(\frac{1}{4} - \frac{1}{4}\pi \right)^{x_4} \left(\frac{1}{4}\pi \right)^{x_5}$$

Nous pouvons maintenant appliquer la formule suivante :

$$g(y|\pi) = \int f(x|\pi) dx$$

sachant que x_3, x_4 et x_5 sont simplement remplacés par leurs valeurs respectives 18, 20 et 34, tandis que x_1 et x_2 parcourent $(0,125), (1,124) \dots (125,0)$. Afin de définir l'algorithme EM, il suffit de montrer comment obtenir $\pi^{(p+1)}$ à partir de $\pi^{(p)}$ c'est-à-dire à partir de la valeur de π à l'itération p , pour $p \in \{1, 2, 3, \dots\}$. Deux étapes sont nécessaires : l'étape moyenne (*Expectation*) détermine les variables statistiques « suffisantes » sachant y . Dans notre cas, 10, 20 et 34 sont connus, si bien que les variables statistiques suffisantes sont x_1 et x_2 avec $x_1 + x_2 = 125$. Les moyennes $x_1^{(p)}$ et $x_2^{(p)}$ à partir de $\pi^{(p)}$ sont fournies de façon immédiate par la formule classique de moyenne de 2 variables aléatoires x_i et x_j de probabilité $p(x_i)$ et $p(x_j)$, $E(x_i) = n \cdot \frac{p(x_i)}{p(x_i) + p(x_j)}$, idem pour $E(x_j)$, soit :

$$x_1^{(p)} = 125 \frac{\frac{1}{2}}{\frac{1}{2} + \frac{1}{4}\pi^{(p)}}, \quad x_2^{(p)} = 125 \frac{\frac{1}{4}\pi^{(p)}}{\frac{1}{2} + \frac{1}{4}\pi^{(p)}}$$

Ces résultats sont injectés dans la forme logarithmique de $f(x|\pi)$:

$$\begin{aligned}\ln(f(x|\pi)) &= \ln\left(\frac{(x_1 + x_2 + x_3 + x_4 + x_5)!}{x_1!x_2!x_3!x_4!x_5!}\right) \\ &\quad + x_1 \ln\left(\frac{1}{2}\right) \\ &\quad + (x_2 + x_5) \ln\left(\frac{1}{4}\pi\right) \\ &\quad + (x_3 + x_4) \ln\left(\frac{1}{4} - \frac{1}{4}\pi\right)\end{aligned}$$

après avoir pris sa moyenne comme suit :

$$\begin{aligned}E(\ln(f(x|\pi))) &= E \ln\left(\frac{(x_1 + x_2 + x_3 + x_4 + x_5)!}{x_1!x_2!x_3!x_4!x_5!}\right) \\ &\quad + x_1^{(p)} \ln\left(\frac{1}{2}\right) \\ &\quad + (x_2^{(p)} + x_5) \ln\left(\frac{1}{4}\pi\right) \\ &\quad + (x_3 + x_4) \ln\left(\frac{1}{4} - \frac{1}{4}\pi\right)\end{aligned}$$

L'étape de Maximisation part des données complètes estimées $x_1^{(p)}$, $x_2^{(p)}$, 18, 20 et 34 et estime π par la méthode du maximum de vraisemblance comme si les données complètes

estimées avaient été réellement observées. Il vient donc :

$$\begin{aligned}
 \frac{\partial}{\partial \pi} E(\ln(f(x|\pi))) &= (x_2^{(p)} + x_5) \left(\frac{1}{\pi} \right) \\
 &\quad + (x_3 + x_4) \left(-\frac{1}{1-\pi} \right) \\
 &= 0 \\
 &\iff \\
 (x_2^{(p)} + x_5) \left(\frac{1}{\pi} \right) &= (x_3 + x_4) \left(\frac{1}{1-\pi} \right) \\
 &\iff \\
 (x_2^{(p)} + x_5) (1-\pi) &= (x_3 + x_4) (\pi) \\
 &\iff \\
 x_2^{(p)} + x_5 - x_2^{(p)}\pi - x_5\pi &= x_3\pi + x_4\pi \\
 &\iff \\
 x_2^{(p)} + x_5 &= x_2^{(p)}\pi + x_5\pi + x_3\pi + x_4\pi \\
 &\iff \\
 \pi &= \frac{x_2^{(p)} + x_5}{x_2^{(p)} + x_5 + x_3 + x_4} \\
 &\iff \\
 \pi^{(p+1)} &= \frac{x_2^{(p)} + 34}{x_2^{(p)} + 34 + 18 + 20}
 \end{aligned}$$

Étude de cas : « Ampoules électriques »

Flury and Zoppe [2000] suppose que la durée de vie d'une ampoule électrique suit une loi exponentielle de paramètre θ . Il nous propose de suivre la durée de vie d'ampoules électriques.

Lors d'une première expérience :

$$(u_1, u_2, \dots, u_n)$$

les durées de vie de n ampoules sont toutes enregistrées.

En revanche, dans une seconde expérience : $(v_1, v_2, \dots, v_m)$ m ampoules sont testées

mais la seule information enregistrée est r qui représente le nombre d'ampoules grillées à t_0 . Cela revient à dire que nous disposons de :

$$u_1, u_2, \dots, u_n \sim i.i.d. e^\theta, \text{ observables}$$

$$v_1, v_2, \dots, v_m \sim i.i.d. e^\theta, \text{ nonobservables}$$

Avec pour fonction de densité :

$$\begin{aligned} f(u|\theta) &= \frac{1}{\theta} e^{-\frac{u}{\theta}} \\ &= \frac{1}{\theta^n} e^{-\frac{\sum_{i=1}^n u}{\theta}} \\ f(v|\theta) &= \frac{1}{\theta} e^{-\frac{v}{\theta}} \\ &= \frac{1}{\theta^m} e^{-\frac{\sum_{i=1}^m v}{\theta}} \end{aligned}$$

La première étape qui calcule la moyenne (*Expectation*) s'écrit ainsi :

$$\begin{aligned} f(u, v|\theta) &= \frac{1}{\theta^{m+n}} e^{-\frac{\sum_{i=1}^n u_i + \sum_{i=1}^m v_i}{\theta}} \\ &\iff \\ \ln f(u, v|\theta) &= -(m+n) \ln \theta - \frac{1}{\theta} \left(\sum_{i=1}^n u_i + \sum_{i=1}^m v_i \right) \\ &\iff \\ Q(\theta|\theta^{(p)}) &= E(\ln f(v|\theta)|u, \theta) \\ &= -(m+n) \ln \theta - \frac{E\left(\sum_{i=1}^n u_i\right) + E\left(\sum_{i=1}^m v_i\right)}{\theta} \end{aligned}$$

avec :

$$E\left(\sum_{i=1}^m v_i\right) = r \left(\theta^{(p)} - t_0 h \right) + (m-r) \left(t_0 + \theta^{(p)} \right)$$

il vient alors :

$$h = \frac{e^{-\frac{t_0}{\theta^{(p)}}}}{1 - e^{-\frac{t_0}{\theta^{(p)}}}}$$

Quant à l'étape de *Maximisation*, elle s'écrit alors ainsi :

$$\frac{\partial}{\partial \theta} Q(\theta | \theta^{(p)}) = -\frac{m+n}{\theta} + \frac{E\left(\sum_{i=1}^n u_i\right) + E\left(\sum_{i=1}^m v_i\right)}{\theta^2} = 0$$

En remplaçant θ et en considérant les valeurs connues de $\sum_{i=1}^n u_i$, on obtient :

$$\theta^{(p+1)} = \frac{E\left(\sum_{i=1}^n u_i\right) + E\left(\sum_{i=1}^m v_i\right)}{n+m}$$

3.2.4 Filtrage

Les données émanant des systèmes globaux de géolocalisation (GPS, GLONASS, Galileo, WPS, 3G...) comportent un grand nombre d'erreurs qu'il faut savoir filtrer. Song and Kim [2012] nous permettent d'accéder à cette problématique. Plus de 12 % des données de positionnement géographique issues des terminaux mobiles de poche (*smart phone*) ou des tablettes sont erronées. Parfois, certains terminaux utilisent jusqu'à 3 sources de données qui ne sont pas toutes d'accord entre elles¹. La Table 3.4 décrit les données de positionnement. P_t est le n-uplet suivant $(t, lat_t, lon_t, D_t, V_t, a_t)$. lat_t définit la latitude à

TABLE 3.4 – Données de positionnement

P_t	$(t, lat_t, lon_t, D_t, V_t, a_t)$
lat_t	latitude à t
lon_t	longitude à t
D_t	distance géodésique Vincenty [1975] à t à partir de (lat_t, lon_t) et $(lat_{t-\Delta_t}, lon_{t-\Delta_t})$
V_t	vitesse à $t = \frac{D_t}{\Delta_t}$
a_t	accélération à t
V_{max}	250 m/s soit 900 km/h
a_{max}	permet de fixer une limite supérieure à l'accélération, le cas échéant

l'instant t . lon_t est la longitude à l'instant t . D_t est la distance géodésique calculée selon Vincenty [1975] à l'instant t à partir de (lat_t, lon_t) et $(lat_{t-\Delta_t}, lon_{t-\Delta_t})$ où Δ_t est décrit plus bas. V_t est la vitesse à l'instant t son calcul est élémentaire : $\frac{D_t}{\Delta_t}$. a_t est l'accélération à l'instant t . La vitesse maximale admise est de $V_{max} = 250 \text{ m/s}$ soit 900 km/h , après quoi la trace est exclue. Les vitesses maximales ne sont que très rarement atteintes dans les

1. L'iPhone 3GS muni d'iOS5, par exemple, utilise aussi bien l'« approvisionnement par la foule » (*crowdsourcing*) Wi-Fi et les réseaux cellulaires que le GPS.

TABLE 3.5 – Vitesse max des transports

Moyen	Vitesse (m/s)
Ambulatoire	3
Vélo	33,33
Auto	92,78
Voiture de course	244,44
TGV	159,67
Avion	528

traces analysées à ce jour ; cf. la Table 3.5 qui fournit les vitesses maximales en fonction des moyens de transport utilisés. Une accélération trop brutale est suspecte et exige de s'être interrogé sur le moyen de transport utilisé avant de la valider. L'accélération maximale tolérée peut également faire l'objet d'un paramètre a_{max} . Pour le calcul de Δ_t , la notion de *vitesse moyenne de déplacement sur n données* est introduite V_μ^n . De même, est introduite une notion d'*écart type de déplacement sur n données*, D_σ^n . n est dit *taille de la fenêtre* ou *window size*. Ainsi, à l'arrivée d'un P_t , si sa V_t ne respecte pas une distribution des vitesses conformes à (μ, σ) , P_t est rejeté. La condition de rejet peut être formulée ainsi : $V_t > V_\mu^n + s \times D_\sigma^n$ où s serait un *niveau de sensibilité de filtrage* ajustable de façon empirique. Dans le cas où P_t est accepté, V_μ^n et D_σ^n sont recalculés. Compte tenu de la simplicité des formules, les calculs correspondants peuvent s'effectuer en temps réel.

3.2.5 Chaînes de Markov à Temps Continu

Dans cette section, nous rappelons brièvement ce qu'est une *chaîne de Markov*, sous la forme d'une « micro » fiche de lecture de Norris [1998]. Après quoi, nous donnons des exemples de son utilisation dans le domaine qui nous concerne.

Soit un processus $(X_t)_{t \geq 0}$ indexé par le temps continu $t \in \mathbb{R}$ à valeurs dans un espace fini ou dénombrable I . Une trajectoire du processus est une fonction $t \mapsto X_t(\omega)$ de $\mathbb{R}_+$ dans I qui dépend de l'aléas ω telle que $\forall \omega, \forall t \geq 0, \exists \epsilon > 0 | \forall s \in [t, t + \epsilon], X_s(\omega) = X_t(\omega)$. Les trajectoires sont a priori continues à droite. Les *temps de saut* sont : $J_0 = 0, J_{n+1} = \inf\{t \geq J_n; X_t \neq X_{J_n}\}$. Les *temps d'attente* sont : $S_n = J_n - J_{n-1}$ si $J_{n-1} < \infty$ et $S_n = \infty$, dans le cas contraire. Un *générateur* sur I est une matrice $Q = (q_{ij})_{i,j \in I}$ qui vérifie : i) $q_{ii} < 0$; ii) $q_{ij} \geq 0$; iii) $\sum_{j \in I} q_{ij} = 0$. Notons $q_i = q(i) = -q_{ii}$. Une *matrice de saut* $\Pi = (\pi_{ij})_{i,j \in I}$

associée à Q est définie par : $\pi_{ij} = q_{ij}/q_i$, si $q_i \neq 0$ et $\pi_{ij} = 0$, dans le cas contraire. $\pi_{ii} = 0$, si $q_i \neq 0$, $\pi_{ii} = 1$, dans le cas contraire. Une *chaîne de Markov à temps continu* $(X_t)_{t \geq 0}$ est déterminée par : i) une mesure λ sur I (identifiée à un vecteur ligne, loi initiale de X_0); ii) un générateur Q qui détermine la dynamique de la chaîne; iii) $(Y_n)_{n \geq 0}$ est une *chaîne de Markov à temps discret* de loi initiale λ et de matrice de transition Π ; iv) en posant $Y_0 = i_0, \dots, Y_{n-1} = i_{n-1}$, les temps d'attente $S_1, \dots, S_n$ sont des variables aléatoires exponentielles indépendantes de paramètres respectifs $q_{i_0}, \dots, q_{i_{n-1}}$.

Autrement dit, supposons que la chaîne X soit à l'état i . Elle y reste un temps exponentiel de paramètre q_i puis passe à un nouvel état, mettons l'état j , avec la probabilité π_{ij} . Elle oublie ensuite ce qui s'est passé, attend un nouveau temps exponentiel de paramètre q_j (indépendant du précédent) et le cycle se répète.

La modélisation sous la forme de *chaînes de Markov à temps continu* des déplacements d'un utilisateur a déjà fait l'objet d'une publication : Kim and Song [2012]. Toutefois, les auteurs ne détaillent pas le procédé d'obtention de leurs chaînes. Ils se contentent de mentionner qu'ils ont, en dernière analyse, souhaité présenter leurs modèles sous cette forme.

D'autres auteurs proposent l'utilisation de *chaînes de Markov à temps continu* pour les déplacements sociaux à l'intérieur d'une famille, individuels ou dépendant de la classe sociale, Thorpe [2007]. Ils aboutissent cependant à la conclusion que *chaînes de Markov à temps continu* ne peuvent à elles seules rendre compte de phénomènes sociaux.

3.3 Une nouvelle démarche

3.3.1 Introduction

À l'approche décrite dans la section précédente, nous souhaitons apporter un autre regard consistant en une démarche nouvelle qui met en œuvre les *Réseaux de Petri*, *RdP* Petri [1966], cf. Fig. 3.1. Celle-ci est déjà présente dans la présentation du chapitre (cf. Sec. 3.1.3). Nous la rappelons ci-dessous brièvement :

1. Nous partons des traces des déplacements de l'utilisateur ;
2. Nous appliquons l'*algorithme- α* pour découvrir les schémas récurrents *patterns* de

mobilité de l'utilisateur et aboutir à un *Réseau de Petri*, *RdP*;

3. En attribuant une temporisation stochastique aux transitions, nous obtenons un *Réseau de Petri Stochastique*, *RdPS*, de graphe des marquages isomorphe à une *Chaîne de Markov à Temps Continue*, *CMTC*;
4. Nous pouvons alors effectuer tous les types d'évaluation permis par les chaînes de Markov.

Malgré les nombreux travaux sur la résolution numérique des chaînes de Markov, ce problème reste toujours très difficile dans le cas d'un espace d'états relativement grand. L'idée est alors d'appliquer l'*algorithme de Lumpabilité* de Kemeny and Snell [1976] afin de construire, par agrégation d'états, une nouvelle chaîne ayant une structure plus légère à résoudre, ce, quelque soit le type d'évaluation considérée. La *Lumpabilité* est en fait une partition des états. On dit que la chaîne est *agrégable au sens fort*, si et seulement si le processus obtenu en agrégeant les états selon la partition (tous les états d'un même ensemble sont regroupés en un seul point) reste markovien.

3.3.2 Réseaux de Petri, Process mining, Algorithme- α

Le formalisme des *Réseaux de Petri* est un outil permettant l'étude de systèmes à événements discrets où la concurrence, la synchronisation, les conflits, la coopération sont prépondérants. Il s'agit d'une représentation mathématique permettant la modélisation d'un système. L'analyse d'un réseau de Petri peut révéler des caractéristiques importantes du système concernant sa structure et son comportement dynamique. Les résultats de cette analyse sont utilisés pour évaluer le système et en permettre la modification voire l'amélioration.

Plus spécifiquement, lorsque le système est l'ensemble des activités ou *workflows* d'une équipe, d'un service, d'un département, d'une division, d'une entité, d'un groupe, l'analyse des traces des *workflows* est dite *Process mining* (Van der Aalst [2011]).

Souvent, cette analyse débute par la mise en œuvre de l'*algorithme- α* Van Der Aalst and Van Dongen [2002]. Un exemple de *Réseau de Petri* issu de cet algorithme est donné Fig. 3.2. On y retrouve :

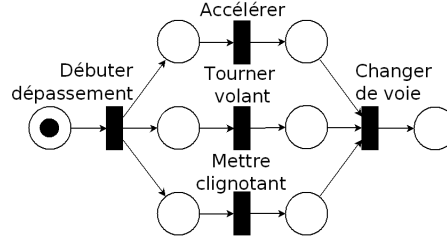


FIGURE 3.2 – Réseau de Petri Workflow-net (WF-net) avec 3 transitions parallèles

- des rectangles pour les *transitions* ou actions ou tâches possibles du *workflow* dont 3 sont parallèles ;
- des cercles pour les *places* ou états possibles du *workflow*.

Une *place d'entrée* i_W et une *em place de sortie* o_W dénotent le début et la fin du *workflow*. Un *marquage* des places par des *jetons* (noirs en l'occurrence, mais ils peuvent être colorés) caractérise l'état du réseau (ici le réseau est dans son état initial).

Une trace s est une séquence $t_1 t_2 \dots t_n$ d'événements produits par l'exécution des tâches du *workflow*. Chaque événement est caractérisé par un type t ou *tâche* et chaque type d'événement est associé à une *transition* du *WF-net* qui l'a produit. La trace d'un *WF-net* est donc la séquence de déclenchement des transitions du *WF-net*.

Le cœur de l'algorithme- α ci-dessous décrit, consiste, à l'étape 4, en la construction d' X_W qui s'appuie sur les quatre relations d'ordre suivantes entre types d'événements : $>_W$, $\rightarrow_W$, $\#_W$ et $\parallel_W$. En Notant a et b deux types d'événements présents dans l'ensemble W des traces du *WF-log*, il vient :

- $a >_W b \iff \exists \sigma = t_1 t_2 t_3 \dots t_{n-1} \in W \wedge i \in \{1, \dots, n-2\} | t_i = a \wedge t_{i+1} = b$;
signifie « a peut être suivi directement de b ». Autrement dit, il existe une trace de W où un événement de type a est directement suivi d'un événement de type b ;
- $a \rightarrow_W b \iff a >_W b \wedge b \not>_W a$;
signifie « a peut être directement suivi de b , mais b n'est jamais directement suivi d' a » ;
- $a \#_W b \iff a \not>_W b \wedge b \not>_W a$;
signifie « a n'est jamais directement suivi de b et b n'est jamais directement suivi d' a ».

- $a \parallel_W b \iff a >_W b \wedge b >_W a$;

signifie « a peut être suivi directement de b et b peut être suivi directement d' a ».

L'algorithme- α se déroule en 8 étapes construisant progressivement les ensembles nécessaires à la production du *WF-net* (dernière étape) :

- Étapes 1 à 3. Construction de T_W , T_I et T_O ;
respectivement, l'ensemble des transitions, l'ensemble des transitions initiales (qui suivent immédiatement i_w), et l'ensemble des transitions finales (qui suivent immédiatement o_w) du *WF-net*. Les transitions correspondent aux types des événements observés dans les traces du *WF-log* ;
- Étape 4. Construction de $X_W = \{(A, B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall a \in A, \forall b \in B, a \rightarrow_W b \wedge \forall (a_1, a_2) \in A^2, a_1 \#_W a_2 \wedge \forall (b_1, b_2) \in B^2, b_1 \#_W b_2\}$;
caractérise un ensemble de places candidates pour le *WF-net*. Le calcul de cet ensemble constitue le point dur de cet algorithme. Chaque place candidate est caractérisée par ses transitions d'entrées et ses transitions de sorties ;
- Étape 5. Construction de $Y_W = \{(A, B) \in X_W \mid \forall (A', B') \in X_W, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\}$;
sous-ensemble de X_W qui ne retient que les couples d'ensembles les plus grands (recherche d'extremum), (A, B) , et élimine, ceux qu'ils incluent, (A', B') ; Y_W caractérise l'ensemble des places candidates, épuré des places implicites, places qui n'ajoutent aucune information au *WF-net* ;
- Étapes 6. Construction de $P_W = \{p(A, B) \mid (A, B) \in Y_W\} \cup \{i_w, o_w\}$;
l'ensemble formé d'une part, des places p du *WF-net*, qui admettent A en entrée et B en sortie, et, d'autre part, les places initiale et finale ;
- Étapes 7. Construction de $F_W = \{(a, p(A, B)) \mid (A, B) \in Y_W \wedge a \in A\} \cup \{(p(A, B), b) \mid (A, B) \in Y_W \wedge b \in B\} \cup \{(i_w, t) \mid t \in T_I\} \cup \{(t, o_w) \mid t \in T_O\}$;
l'ensemble des arcs du *WF-net* déduits de Y_W , T_I et T_O ;
- Étape 8. Construction de $\alpha(W) = (P_W, T_W, F_W)$;
le *WF-net* reconstruit.

La notion de complétude d'un *WF-log* est intimement liée à la relation $>_W$. Un *WF-log* est considéré comme complet au regard d'un modèle, si, pour chaque couple de types

d'événements, le *WF-log* suffit à caractériser correctement cette relation $>_W$. Étant donné un *WF-log* complet d'un *WF-net*, l'algorithme- α est capable de reconstruire ce *WF-net*, au renommage des places près.

3.3.3 Lumpabilité

L'utilisation de l'*algorithme de Lumpabilité*, en tant que technique d'agrégation des données statistiques présentées sous la forme de *chaînes de Markov*, fait l'objet d'actives recherches.

Évoquant l'explosion combinatoire du coût de résolution d'un problème lorsqu'on se donne une inconnue supplémentaire (c'est-à-dire un degré de liberté supplémentaire, ou, d'un certain point de vue, une dimension d'espace supplémentaire), Richard Ernest Bellman a lancé l'expression « fléau de la dimension » (*Curse of dimensionality*). C'est par cette citation que Zakharin [1972] ouvre son article qui propose une parade à cette explosion combinatoire consistant en l'association au processus stochastique d'origine d'un processus groupant (*lumping process*) qui, selon certaines conditions, peut être de même classe que le processus aléatoire d'origine. L'article se limite aux classes de processus markoviens ou semi-markoviens.

Considérons le processus aléatoire $\{\Xi(t), t \in T\}$, évolution de (A, σ_A) , espace d'états où σ_A est une σ -algèbre ou tribu sur A , à savoir un sous-ensemble de $\text{Pr}(A)$ stable pour le complémentarité et l'union. Par exemple, A peut représenter l'ensemble des états possibles d'un système physique complexe et $\Xi(t) \in A$ son état à l'instant t . Soit, une relation d'équivalence quelconque sur A . Celle-ci définit, bien entendu, une partition sur A c'est-à-dire un ensemble de parties disjointes dont l'union couvre l'intégralité de A . Ces parties sont dites *classes d'équivalence* de la relation considérée. Notons a l'ensemble des classes et σ_a une σ -algèbre quelconque contenant l'ensemble des singletons de a .

Soit maintenant, une fonction φ de A vers a , (σ_A, σ_a) -mesurable, c'est-à-dire que $\forall b \in \sigma_a, \varphi^{-1}(b) = \bigcup_{\alpha \in b} \varphi^{-1}(\alpha) \in \sigma_A$. Nous nommons le processus aléatoire $\xi(t) = \varphi(\Xi(t))$ sur (a, σ_a) *processus φ -groupant* et disons que ξ est le résultat du groupement d'états du processus original Ξ par l'intermédiaire de la partition sur A définie par $\varphi^{-1}(\alpha), \alpha \in a$, ou, plus simplement, que ξ est le résultat du processus de groupement Ξ .

Derisavi et al. [2003] nous propose d'en étudier le coût qui est du type :

$$\mathcal{O}(m \lg n)$$

3.4 Implémentation

Notre but est d'implémenter notre modèle d'analyse de traces en lui fournissant en entrée, les traces issues de nos modèles de mobilité sociale, objet du chapitre suivant, aménagées de telle sorte qu'elles soient assimilables à des traces de *PGI/ERP*² ou de *GRC/CRM*³ desquels sont généralement issues les traces fournies en entrée de l'algorithme- α , à savoir une forme proche de la Table 3.6.

TABLE 3.6 – Un journal de Workflow

Cas	Tâche	Début	Fin
1	A	12/09/2012 10 h 01	12/09/2012 10 h 05
2	A	12/09/2012 10 h 05	12/09/2012 10 h 09
3	A	12/09/2012 10 h 16	12/09/2012 10 h 21
1	B	12/09/2012 10 h 22	12/09/2012 10 h 23
1	C	12/09/2012 10 h 26	12/09/2012 10 h 40
2	B	12/09/2012 10 h 42	12/09/2012 10 h 51
1	D	12/09/2012 10 h 51	12/09/2012 10 h 57
2	D	12/09/2012 11 h 01	12/09/2012 11 h 05
1	E	12/09/2012 11 h 05	12/09/2012 11 h 17
2	C	12/09/2012 11 h 17	12/09/2012 11 h 28
3	F	12/09/2012 11 h 29	12/09/2012 11 h 35
2	E	12/09/2012 11 h 35	12/09/2012 11 h 44
3	G	12/09/2012 11 h 46	12/09/2012 12 h 02

Van Dongen and Van der Aalst [2004] puis Ling et al. [2008] proposent un modèle d'implémentation de l'*algorithme- α* . Ce dernier article est difficile à lire car certains éléments restent indéterminés. Par exemple, on trouve *listOfFirst* et *listOfLast*, non expliqués et non utilisés... Par ailleurs, *leftSide* et *rightSide*, non expliqués et utilisés partout... Notre reformulation est une tentative de résolution de toutes ces ambiguïtés ou de ces non-dits que d'aucuns pourraient qualifier d'anodins et qui ont été, pour nous, sources

2. Progiciel de Gestion Intégré/Enterprise Resource Planning

3. Gestion de la Relation Client/Customer Relationship Management

de difficulté. Cette reformulation s'appuie sur Matheron et al. [2011], Van Der Aalst and Van Dongen [2002].

La structure de données comporte les éléments suivants :

- *File Class* : *Premier entré, premier Servi, PEPS ou First in, first out, FIFO*, elle admet, outre ses opérations de base, celles de recherche de données spécifiques et de comparaison de nœuds. Il n'est pas nécessaire d'en définir la taille car elle est extensible par nature. Elle constitue l'objet de base de l'algorithme ;
- *Transition Class* : une transition (=une tâche) du *WF-net* avec *id* et *nomUnique* ;
- *Place Class* : une place du *WF-net* avec *id* et *nomUnique* ;
- *RelationCausale Class* : une relation de *cause* c'est-à-dire une *Transition* c'est-à-dire une *Tâche*, à *effet* c'est-à-dire une autre *Transition* c'est-à-dire une autre *Tâche*. Des booléens *ÀSupprimer* et *Utilable* commandent la destruction des doublons et données obsolètes.

À l'aide de la structure de données ci-dessus et des relations d'ordre ci-dessous, l'algorithme- α s'implémente aisément (W =workflow ; s =trace ; a, b, t_i =tâche=transition) :

$$\begin{cases} a >_W b & \iff \exists s = \{t_1, \dots, t_n\} \in W | a = t_i \wedge b = t_{i+1} \\ a \rightarrow_W b & \iff a >_W b \wedge b \not>_W a \\ a \#_W b & \iff a \not>_W b \wedge b \not>_W a \\ a \parallel_W b & \iff a >_W b \wedge b >_W a \end{cases}$$

En entrée, *WF-log* avec un éventuel pré-traitement qui consiste en l'élagage de toutes les traces parasites correspondant à des doublons.

En sortie, une *file* incluant les places et transitions générées par la fouille (*mining*) du *WF-log*.

Soit les ensembles de *relationCausales* suivantes :

- T : l'ensemble de toutes les tâches t des traces de *WF-log*
- I : l'ensemble des tâches t Initiales des traces ;
- O : l'ensemble des tâches t (Out) Finales des traces.

3.4. IMPLÉMENTATION

L'algorithme- α s'écrit alors comme suit :

// Contruire G , le parcours de Gauche à droite des places c'est-à-dire des *relationCausales* du *WF-net* en allant des causes G_c vers les effets G_e :

POUR chaque $t_c \in T$ *FAIRE*

$trouvé = FAUX$

TANTQUE il reste $T_{\#} \subset Pr_{\#}(T)$ *ET* $trouvé = FAUX$

TANTQUE il reste $t_{\#} \in T_{\#}$ *ET* $trouvé = FAUX$

SI $t_c \rightarrow_W t_{\#}$ *ALORS*

$G_c \leftarrow t_c$

$G_e \leftarrow T_{\#}$

$trouvé = VRAI$

FINSI

FINTANTQUE

FINTANTQUE

FINPOUR

// élaguer G

POUR chaque $t_{c_1} \in G_c$ *FAIRE*

POUR chaque $t_{c_2} \in G_c$ *FAIRE*

SI $t_{c_1} \#_W t_{c_2}$ *ALORS*

$trouvé = FAUX$

TANTQUE il reste $T_e \subset G_e$ *AND* $trouvé = FAUX$ *FAIRE*

$trouvé \leftarrow t_{c_1} \in T_e \wedge t_{c_2} \in T_e$

FINTANTQUE

SI $trouvé$ *ALORS* Retirer t_{c_2} de G_c *FINSI*

FINSI

FINPOUR

FINPOUR

// Construire D , le parcours de Droite à gauche des places c'est-à-dire des *relationCausales* du *WF-net* en allant des effets D_e vers les causes D_c :

POUR chaque $t_e \in T$ *FAIRE*

$trouvé = FAUX$

TANTQUE il reste $T_{\#} \subset Pr_{\#}(T)$ *ET* $trouvé = FAUX$

TANTQUE il reste $t_{\#} \in T_{\#}$ *ET* $trouvé = FAUX$

SI $t_{\#} \rightarrow_W t_e$ *ALORS*

$D_c \leftarrow T_{\#}$

$D_e \leftarrow t_e$

$trouvé = VRAI$

FINSI

FINTANTQUE

FINTANTQUE

FINPOUR

// élaguer D

POUR chaque $t_{e_1} \in D_e$ *FAIRE*

POUR chaque $t_{e_2} \in D_e$ *FAIRE*

SI $t_{e_1} \#_W t_{e_2}$ *ALORS*

$trouvé = FAUX$

TANTQUE il reste $T_e \subset D_e$ *AND* $trouvé = FAUX$ *FAIRE*

$trouvé \leftarrow t_{e_1} \in T_e \wedge t_{e_2} \in T_e$

FINTANTQUE

SI $trouvé$ *ALORS* Retirer t_{e_2} de D_e *FINSI*

FINSI

FINPOUR

FINPOUR

3.5 Conclusion

Dans ce chapitre, nous avons introduit la notion de *Profil de Mobilité Utilisateur* par quelques définitions.

Nous nous sommes attachés à présenter un état de l'art qui s'appuie sur des publications des plus récentes dans ce domaine.

Celles-ci font appel à des méthodes statistiques avancées mettant en œuvre :

- diverses *entropies* ;
- la *Maximisation de l'espérance* ;
- le *Filtrage* des données ;
- les *Chaînes de Markov à Temps Continu*.

Pour chacun de ses thèmes, nous avons donné leurs bases formelles accompagnées d'exemples.

Nous avons proposé une démarche nouvelle qui met en œuvre :

- les *Réseaux de Petri* spécialisés que sont les *Workflow-Nets* ;
- lesquels sont issus de l'analyse des traces de l'utilisateur par l'*Algorithme- α* ;
- et sur lesquels nous appliquons l'*algorithme de Lumpabilité*.

Nous proposons enfin une implémentation de notre démarche sous la forme d'un pseudo-code. Celui-ci pourra faire l'objet d'une implémentation dans un langage de programmation. Des campagnes de tests pourront alors être lancées afin d'asseoir notre démarche. Les traces utilisateur utilisées pourront être de différents profils.

Nous aurons peut-être la chance d'utiliser des traces réelles. Toutefois, comme nous l'avons déjà publié dans un précédent travail (Costantini and Boumerdassi [2012]), les profils utilisateur issus de traces sont très limités dans l'espace et le temps. En revanche, définir un profil utilisateur en termes de mixage de nos modèles de mobilité (cf. Sect. 4.5), nous apparaît plus pérenne.

Chapitre 4

La mobilité sociale

4.1 Introduction

Les terminaux mobiles étant portés par des humains, leurs déplacements sont souvent répétitifs et peuvent être corrélés à certaines caractéristiques propres à l'humain et, d'une certaine manière, reproductibles en laboratoire. Or, cette reproductivité est capitale dans l'élaboration des environnements de test de nouveaux protocoles ou de nouvelles applications pour réseaux mobiles. Elle se décline formellement en *modèles de mobilité*. Nous en trouvons chez Mousavi et al. [2007] une définition concise et précise : « *Un modèle de mobilité est destiné à décrire le mode de circulation des utilisateurs mobiles et la façon dont leurs localisation, vitesse et accélération évoluent dans le temps* ». Le défi est de trouver des modèles les plus fidèles possible des comportements réels des utilisateurs. C'est ce défi que nous nous proposons de relever dans le présent chapitre.

4.2 La notion du social

4.2.1 Introduction

[Heck and Ghosh 2000] affirment qu'un réseau social permet de décrire un modèle de mobilité de groupe propre aux réseaux *ad-hoc* ([Musolesi and Mascolo 2009]). D'autres modèles de mobilité font se déplacer les nœuds vers des Points d'Attraction [Jardosh et al. 2005]. Un terminal mobile suit le comportement de son utilisateur, il est donc important de prendre en compte le comportement social des êtres humains dans la modélisation de la

mobilité de ces appareils. *La théorie du social* est de plus en plus largement utilisée dans le domaine des réseaux mobiles. Les fondements et les progrès réalisés dans l'analyse des réseaux sociaux complexes peuvent être trouvés dans Maxwell [1873] et Eason et al. [1955]. Ces modèles sont réputés permettre de décrire de nombreux types de relations, y compris les rapports sociaux réels [Jacobs and Bean 1963].

4.2.2 Points d'Attraction (PA)

Lors d'achats sur *Internet*, un produit classé comme le plus acheté nous attire. Au moment de choisir une destination de vacances, les lieux très fréquentés et réputés nous attirent. En période de soldes, plus un magasin est peuplé, plus il nous attire. Dans la littérature, les Points d'Attraction modélisent ces centres d'intérêt [Jardosh et al. 2005]. Nous utilisons cette notion de Points d'Attraction et y introduisons des comportements sociaux. Pour modéliser les comportements cités ci-dessus et l'appliquer au domaine de la modélisation de la mobilité, la première étape consiste à identifier les *Points d'Attraction* du réseau. Ensuite, est associée à un *Points d'Attraction* sa réputation sous la forme d'une valeur sociale dite *attractivité* qui représente sa popularité relativement aux autres points. Un nœud mobile visite un sous-ensemble de Points d'Attraction qui correspond à ses centres d'intérêt. Il visite toujours le Point d'Attraction le plus proche et/ou le plus intéressant c'est-à-dire qui a la plus grande valeur d'attractivité.

4.2.3 Modélisation

Soit N , l'ensemble des nœuds d'un réseau sans fil. Soit A , un ensemble de Points d'Attraction. Soit T , l'ensemble des points du temps, tels que $t \in T \iff t = b\Delta$ avec $b \in \mathbb{N}$, $\Delta \in \mathbb{R}$ et $t \leq D$, i.e. la durée de simulation. Soit $r : A \rightarrow [0; 1]$, la normalisation de la valeur d'attractivité $u(a)$ avec $a \in A$, telle que :

$$r(a) = \frac{u(a)}{\sum_{a' \in A} u(a')}$$

avec :

$$u(a) = \begin{cases} user_value & \text{si défini par l'utilisateur} \\ \frac{1}{|A|} & \text{sinon} \end{cases} \quad (4.1)$$

Soit $d : A \times N \times T \rightarrow \mathbb{R}$, la distance euclidienne du Point d'Attraction a au nœud n à l'instant t . Soit $v : A \times N \times T \rightarrow \{V, F\}$ où V et F signifient VRAI et FAUX, l'application qui définit l'ensemble des Points d'Attraction à visiter à l'instant t par le nœud n :

$$v(a, n, t) = \begin{cases} T & \text{si non défini par l'utilisateur} \\ F & \text{si visité} \end{cases}$$

Finalement : n attiré par $a \iff$

$$a = \arg \max_{a' | v(a', n, t) = V} \frac{r(a')}{d(a', n, t)} \quad (4.2)$$

4.2.4 Application : Social Manhattan (SMN)

Notre première classe de modèles de mobilité sociale, *Social Mobility Models* introduit nos Points d'Attraction. Comme exemple d'application, nous implémentons nos Points d'Attraction sociaux sur le modèle de mobilité *Manhattan Grid Mobility Model* de Jayakumar and Gopinath [2008]. Ce modèle décrit le mouvement des nœuds dans une ville à l'aide d'une grille représentant des routes à double sens qui se croisent à angle droit. À une intersection, un nœud mobile continue tout droit avec une probabilité 1/2, tourne à gauche ou à droite avec une probabilité 1/4 pour chaque direction.

La partie gauche de la Fig. 4.1 présente un schéma de principe de Manhattan.

Le schéma de principe de notre modèle *Social Manhattan Mobility Model (SMN)* [Boumerdassi et al. 2009] est donné dans la partie droite de la Fig. 4.1. Se référer au paragraphe 4.2.2, pour les motivations.

4.2.5 Conclusion

Nous élaborons dans cette section, un premier modèle de mobilité à Points d'Attraction à attractivité fixe.

Nous mettons en évidence (Annexe A) que le manque de prise en compte de l'aspect social résulte en une surestimation des performances. Notamment :

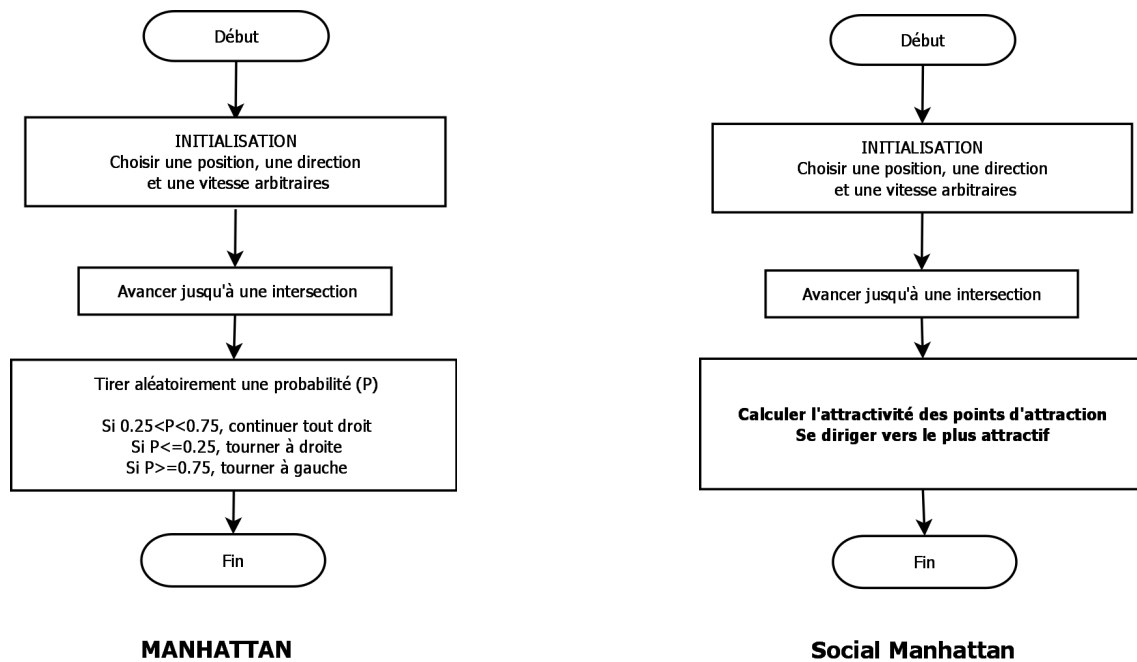


FIGURE 4.1 – SLS versus Social SLS - Request/Reply transmitted - 50 nodes

La moyenne globale de Load lors du passage de Manhattan (MAN) à Social Manhattan (SMN) est de : -0,13. Cela signifie que Manhattan est surévalué en l'absence de prise en compte de l'aspect social.

Ces résultats nous permettent de conclure que la prise en compte de l'aspect social des communications dans les réseaux sans fil devient une nécessité tant pour les protocoles réactifs que les protocoles proactifs, d'une part, et, d'autre part, tant du point de vue débit (cruciaux pour les applications multimédia en plein essor) que des point de vue vitesse de déplacement (la mobilité étant LE challenge d'aujourd'hui) et la densité laquelle fait l'objet de nombreuses études qui s'appuient souvent sur des traces réelles afin d'anticiper les goulots d'étranglement.

4.3 Les réseaux pondérés

4.3.1 Introduction

Dans cette section, nous proposons une nouvelle classe de modèles de mobilité sociale pondérée, *Weighted Social Mobility Models*, dans laquelle les valeurs sociales affectées aux Points d'Attraction changent au fil du temps. Elles sont étroitement liées à la fois au

nombre de visites et à la distance qui les séparent des nœuds mobiles de façon à modéliser un *effet boule de neige* ou *de bouche à oreille*. La notion de « poids social », *weight*, d'un nœud du réseau est définie et fait varier l'attractivité d'un Points d'Attraction (augmenter l'attractivité) en fonction du poids du nœud mobile à chaque fois qu'un nœud mobile visite un lieu.

4.3.2 Points d'Attraction pondérés

Nous introduisons les caractéristiques suivantes dans notre nouvelle classe de modèles de mobilité sociale pondérée *Weighted Social Mobility Models* :

- des *Points d'Attraction (PA)*, avec des attractivités croissantes $r(t)$;
- des *nœuds mobiles (NM)*, avec des poids w représentant la classe sociale de l'utilisateur.

Un nœud mobile se dirige vers le Point d'Attraction vers lequel il est le plus attiré ($\arg \max_i r_i(t)$). En y parvenant, il provoque l'ajout à la popularité de ce dernier, du poids social du visiteur : $r(t)+ = w$.

4.3.3 Modélisation

Nous reprenons notre modèle décrit en section 4.2.3, p. 42 et y introduisons les changements nécessaires pour l'adapter à nos nouveaux modèles de mobilité sociale pondérée.

Pour notre nouveau modèle, un poids aléatoire $w(n)$ est associé à chaque nœud mobile n . Chaque fois qu'un nœud arrive à un Point d'Attraction, le poids de ce point augmente. Les nœuds choisissent pour nouvelle destination, le Point d'Attraction d'attractivité la plus forte.

Soit l'application $w : N \rightarrow [0; 1]$ avec $\sum_{n \in N} w(n) = 1$, le poids social (i.e. la classe sociale) des nœuds. En introduisant w , l'attractivité de $a \in A$ à l'instant $t \in T$ devient l'application $r : A \times T \rightarrow [0; 1]$, telle que :

$$r(a, t) =$$

$$\begin{cases} \frac{u(a)}{\sum_{a' \in A} u(a')} & t = 0 \\ \frac{\sum_{n' \in N} w(n') \min(r_1(a, n', t - \Delta), r_{max}(t))}{\sum_{a' \in A} \sum_{n' \in N} w(n') \min(r_1(a', n', t - \Delta), r_{max}(t))} & t \neq 0 \end{cases}$$

avec $u(a)$ tel que défini à l'Eq. 4.1, p. 42 de la section 4.2 où

$$r_1(a, n, t) = \frac{\frac{r(a, t)}{d(a, n, t)}}{\sum_{a' \in A} \frac{r(a', t)}{d(a', n, t)}} \quad (4.3)$$

4.3.4 Application : Weighted Manhattan (WMN)

Comme pour notre première classe de modèles de mobilité sociale, nous choisissons le modèle *Manhattan* comme exemple d'application pour notre classe de modèles de mobilité sociale pondérée. Le modèle *Weighted Social Manhattan (WMN)* [Boumerdassi et al. 2010] introduit le modèle mathématique précédent dans *Manhattan*.

Le schéma de principe de *WMN* est donné en figure Fig. 4.2.

L'implémentation sous Opnet Modeler et les configurations de test sont décrites en Annexe A.2.1.

4.3.5 Conclusion

Nous élaborons dans ce chapitre, un deuxième modèle de mobilité dont l'attractivité des Points d'Attraction croît proportionnellement au nombre de visites.

Nous mettons en évidence (Annexe A) que le manque de prise en compte de l'aspect social résulte en une surestimation des performances. Notamment :

La moyenne globale de Load lors du passage de Manhattan (MAN) à Weighted Manhattan (WMN) est de : -0,10. Cela signifie que Manhattan est surévalué en l'absence de prise en compte de l'aspect social.

Ces résultats nous permettent de conclure que la prise en compte de l'aspect social évo-

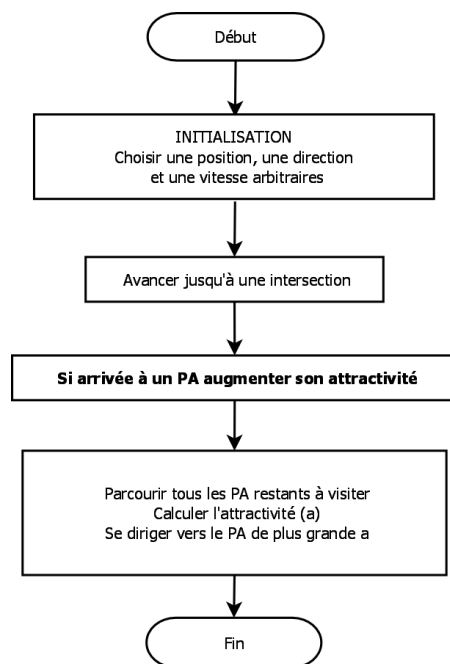
**Weighted Manhattan**

FIGURE 4.2 – Weighted Manhattan (WMN) : schéma de principe

lutf des Points d'Attraction, doit faire l'objet d'attention dans la conception des protocoles de communications dans les réseaux sans fil, au même titre que *SMN*.

4.4 Les systèmes de colonies de fourmis

4.4.1 Introduction

Dans cette section, nous intégrons dans nos modèles de mobilité sociale, l'*intelligence en essaim* qui désigne les systèmes où l'intelligence de l'ensemble est supérieure à la somme de ses parties. Dans celle-ci, les systèmes de colonies de fourmis, *Ant Colony System (ACS)* forment une des branches les plus importantes [Dorigo 1992], [Günes et al. 2002].

La suite de cette section est organisée comme suit, le modèle associé et un exemple d'application *Ant Manhattan Mobility Model (AMN)* sont respectivement présentés en sections 4.4.2 et 4.4.3. Enfin la section 4.4.4 conclut ce chapitre.

Les configurations de tests et les résultats associés sont présentés et discutés en An-

nexe A.3.1,

4.4.2 Modélisation

Nous introduisons le mécanisme de phéromone de la section précédente, dans notre modèle mathématique. Eq. 4.3, p. 46 devient :

$$r_1(a, n, t) = \frac{\frac{r(a, t)}{d(a, n, t)} + \tau(a, t)}{\sum_{a' \in A} \frac{r(a', t)}{d(a', n, t)} + \tau(a', t)}$$

où $\tau(a, t)$ est la quantité de phéromone au Point d'Attraction a à l'instant t tel que :

$$\tau(a, t) = \begin{cases} 0 & t = 0 \\ (1 - \alpha)\tau(a) & t \neq 0 \\ \tau(a, t) + w(n) & n \text{ arrive à } a \end{cases} \quad (4.4)$$

où $0 \leq \alpha \leq 1$ est le taux de décroissance i.e. d'évaporation de la phéromone.

Enfin, pour modéliser l'importance relative de la distance par rapport au taux d'évaporation de la phéromone, il vient : n attiré par $a \iff$

$$a = \arg \max_{a' | v(a', n, t) = T} \frac{r(a', t)}{d(a', n, t)^\beta} \quad (4.5)$$

où β permet de paramétrer l'importance relative de la distance par rapport au taux d'évaporation de la phéromone. ($\beta > 0$).

4.4.3 Application : Ant Manhattan (AMN)

Comme pour les modèles *Social Manhattan (SMN)* et *Weighted Manhattan (WMN)*, un exemple d'application est donné en utilisant comme base le modèle *Manhattan (MN/MAN)* auquel nous appliquons notre modèle mathématique.

Son schéma fonctionnel est donné en Fig. 4.3.

Le modèle *Ant Manhattan* rajoute *Pheromone Decay Factor* et *Pheromone Distance Exponent* qui implémente les paramètres α et β , cf. les équations Eq. 4.4 et Eq. 4.5, p. 48.

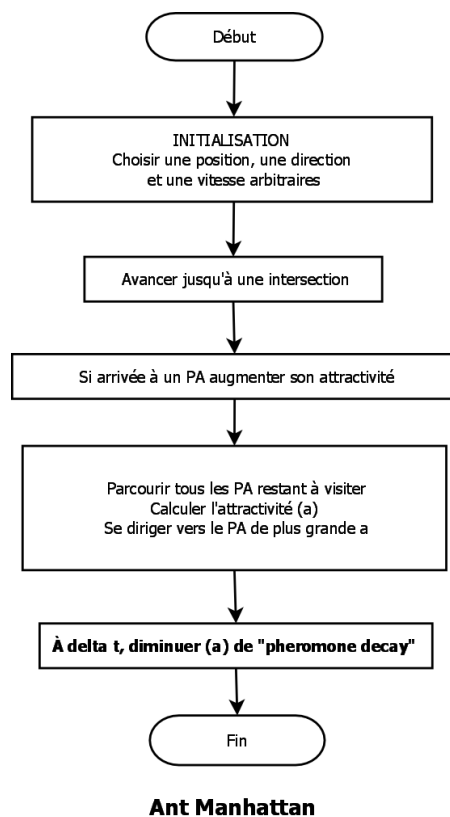


FIGURE 4.3 – Mouvement d'un nœud selon Ant Manhattan : Schéma fonctionnel

4.4.4 Conclusion

Nous élaborons dans cette section, un troisième modèle de mobilité à Points d'attraction à attractivité croissante et décroissante selon le nombre de visites.

Nous mettons en évidence (Annexe A) que le manque de prise en compte de l'aspect social résulte en une surestimation des performances. Notamment :

La moyenne globale de Load lors du passage de Manhattan (MAN) à Ant Manhattan (AMN) est de : -0,10. Cela signifie que Manhattan est surévalué en l'absence de prise en compte de l'aspect social.

Ces résultats nous permettent de conclure que la prise en compte de l'aspect social évolutif des Points d'Attraction, c'est-à-dire, le fait de faire croître leurs attractivités *mais également* le fait de la faire décroître en fonction des visites, doit faire l'objet d'attention dans la conception des protocoles de communications dans les réseaux sans fil, au même

TABLE 4.1 – Mixte Distribution

Model	%	Density 20	Density 40
SMN	50	10	20
WMN	20	4	8
AMN1	10	2	4
AMN2	10	2	4
AMN3	10	2	4

titre qu'*SMN* et *WMN*.

4.5 Mixage de modèles

4.5.1 Introduction

Dans cette section, nous traitons du fait que les gens ne se déplacent pas selon un unique modèle de mobilité mais selon de multiples modèles. C'est la raison pour laquelle nous reprenons l'ensemble des scénarios précédemment exposés en y adjoignant le mixte de l'ensemble des modèles dans chaque scénario. Les résultats sont donnés dans les Fig. 4.4 à 4.7.

4.5.2 Distribution des modèles

Le mixte suit la distribution donnée en Table 4.1.

4.5.3 Configuration de test

Les différentes valeurs de nos paramètres de simulation sont résumés en Table 4.2.

4.5.4 Résultats

Les quatre Fig. 4.4 à 4.7 présentent les quatre variables statistiques réseau suivantes : Load, Throughput (Thru), Delay (Dlay), Data dropped (Retry Threshold Exceeded, DrpR).

La Table 4.3 en donne la légende.

TABLE 4.2 – Simulation

Simulateur	OPNET Modeler 16.0
Couche MAC	IEEE 802.11
Taux de transmission	250 m
Type de trafic	UDP
Destination	aléatoire
Caractéristiques physiques	Extended Rate PHY (802.11g)
Protocoles de routage	OLSR, AODV
Débit (Mbit/s)	11, 24, 54
Vitesse des nœuds (m/s)	2 (7 km/h, piéton, <i>pedestrian</i> , <i>ped</i>) 3 (11 km/h, bicyclette, <i>bicycle</i> , <i>bic</i>) 15 (52 km/h, véhicule, <i>car</i>)
Nombre de nœuds	20, 40
Aire de simulation	1000 m × 1000 m
Block length	50 m
Safety distance	30 m
Move step	25 m
Nombre de Points d'Attraction	4, 8
Attractivity	5
Modèle de mobilité	Manhattan (MN) Social Manhattan (SMN) Weighted Social Manhattan (WMN) Ant Manhattan (AMN)
(α, β)	AMN1 : 0.005 : norm. decay, 1.5 : norm. dist. AMN2 : 0.002 : weak decay, 2 : heavy dist. AMN3 : 0.01 : heavy decay, 0.5 : weak dist.

Selon la nomenclature utilisée dans l'implémentation (Annexe A), il vient le paramétrage suivant :

- *Safety Distance* : distance de sécurité routière ; valeur : 30 m, valeur moyenne en zone urbaine ;
- *Move Step* : unité de mouvement d'un nœud mobile ; valeur : 25 m, pour ne pas alourdir les temps de simulation ; diviseur du *Block Length* afin que les nœuds « passent » par les intersections ;
- *Block Length* : distance entre 2 lignes ou 2 colonnes de la grille ; valeur : 50 m, largeur moyenne d'un pâté de maisons ;
- x_{min} , y_{min} , x_{max} , y_{max} : aire de simulation ; valeur : 1 km^2 , superficie moyenne d'un quartier.

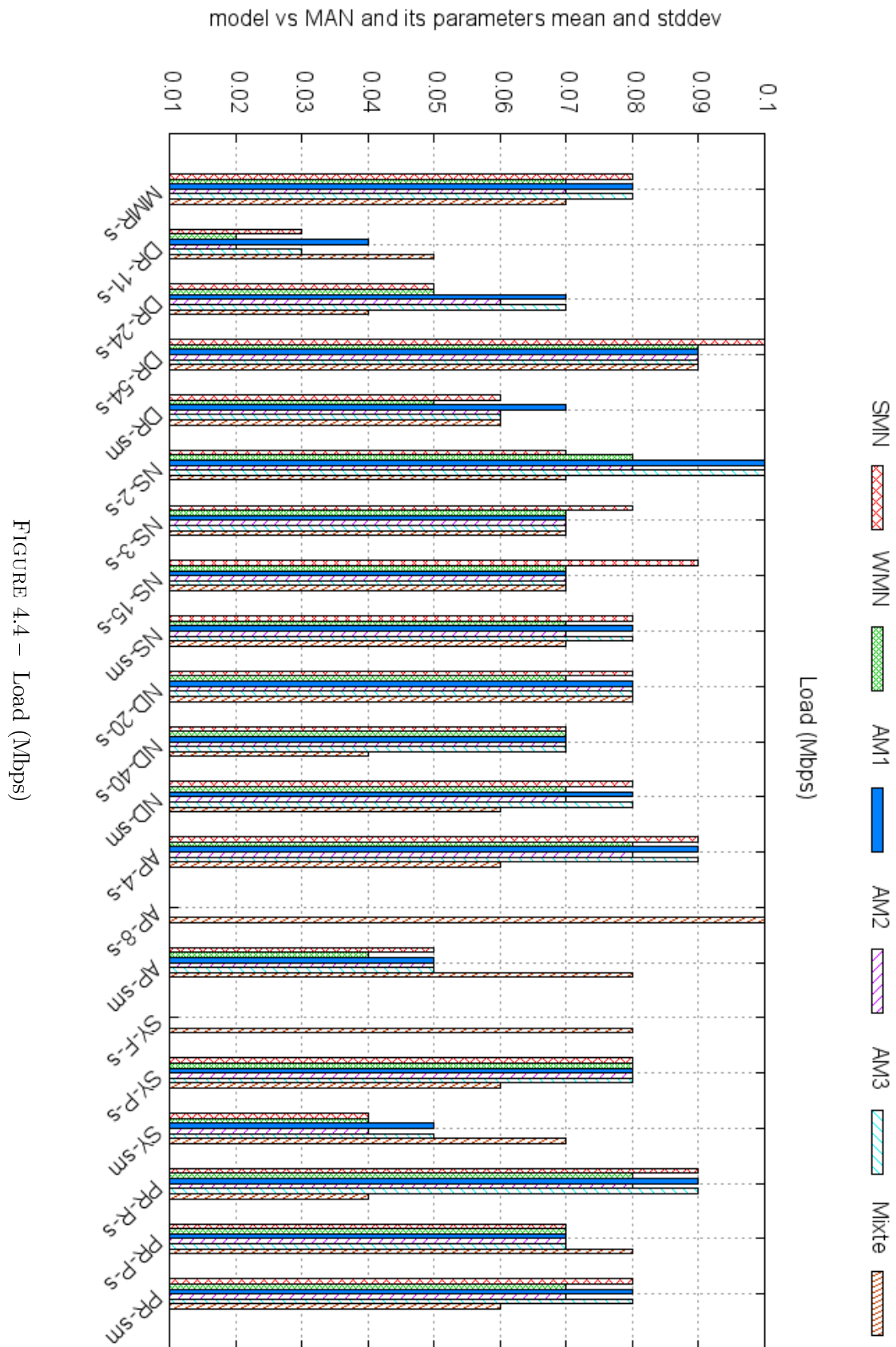


FIGURE 4.4 – Load (Mbps)

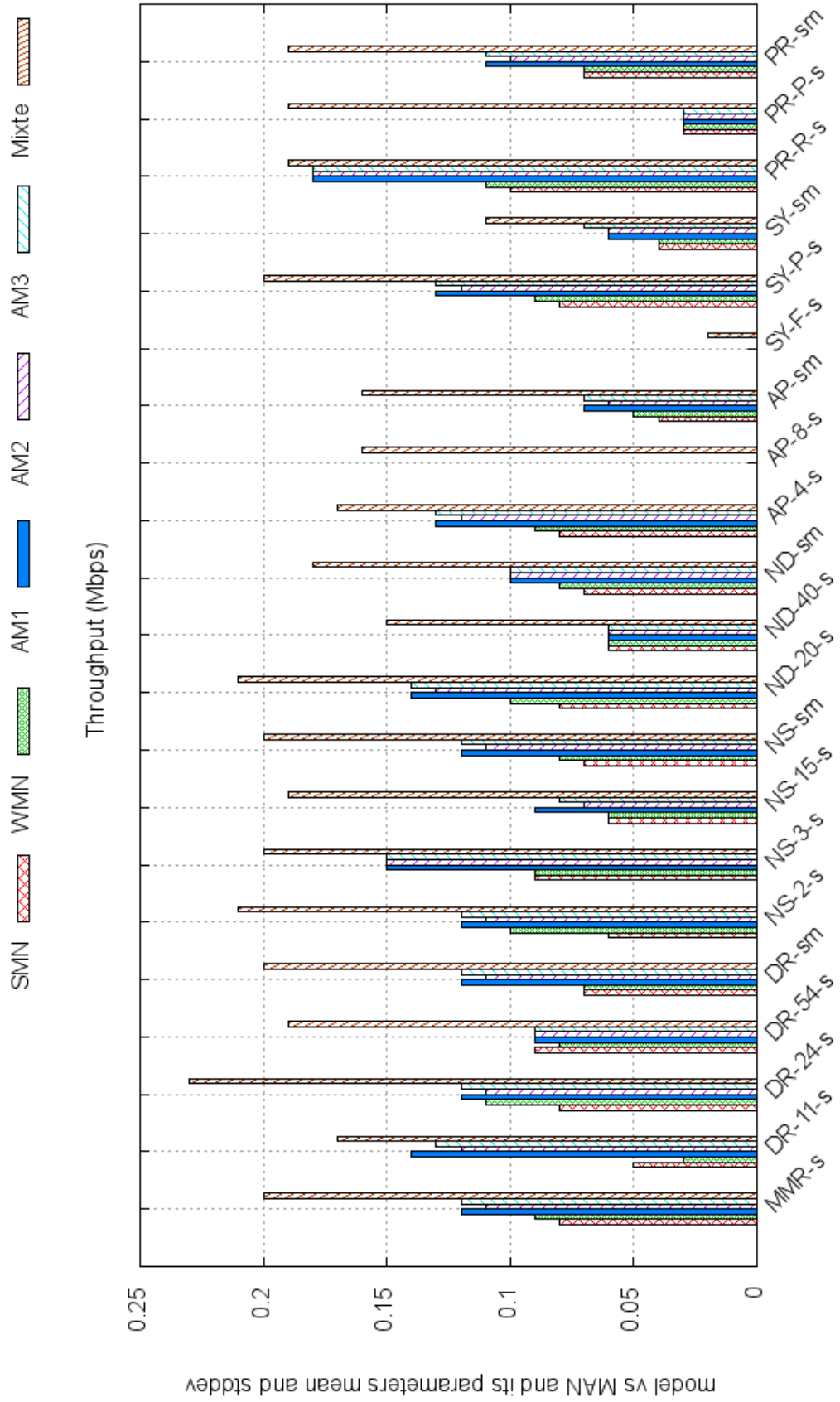


FIGURE 4.5 – Throughput (Mbps)

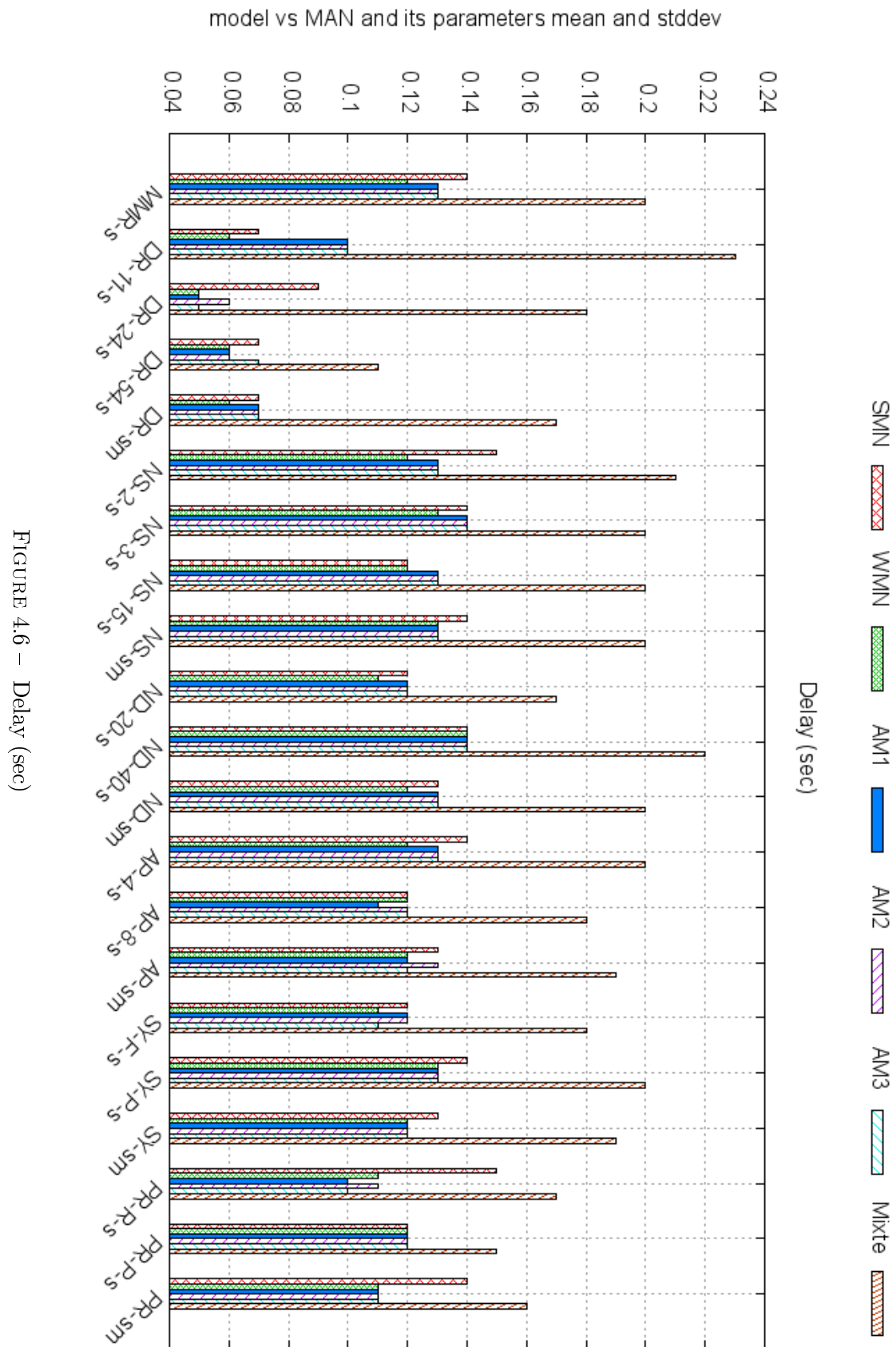


FIGURE 4.6 – Delay (sec)

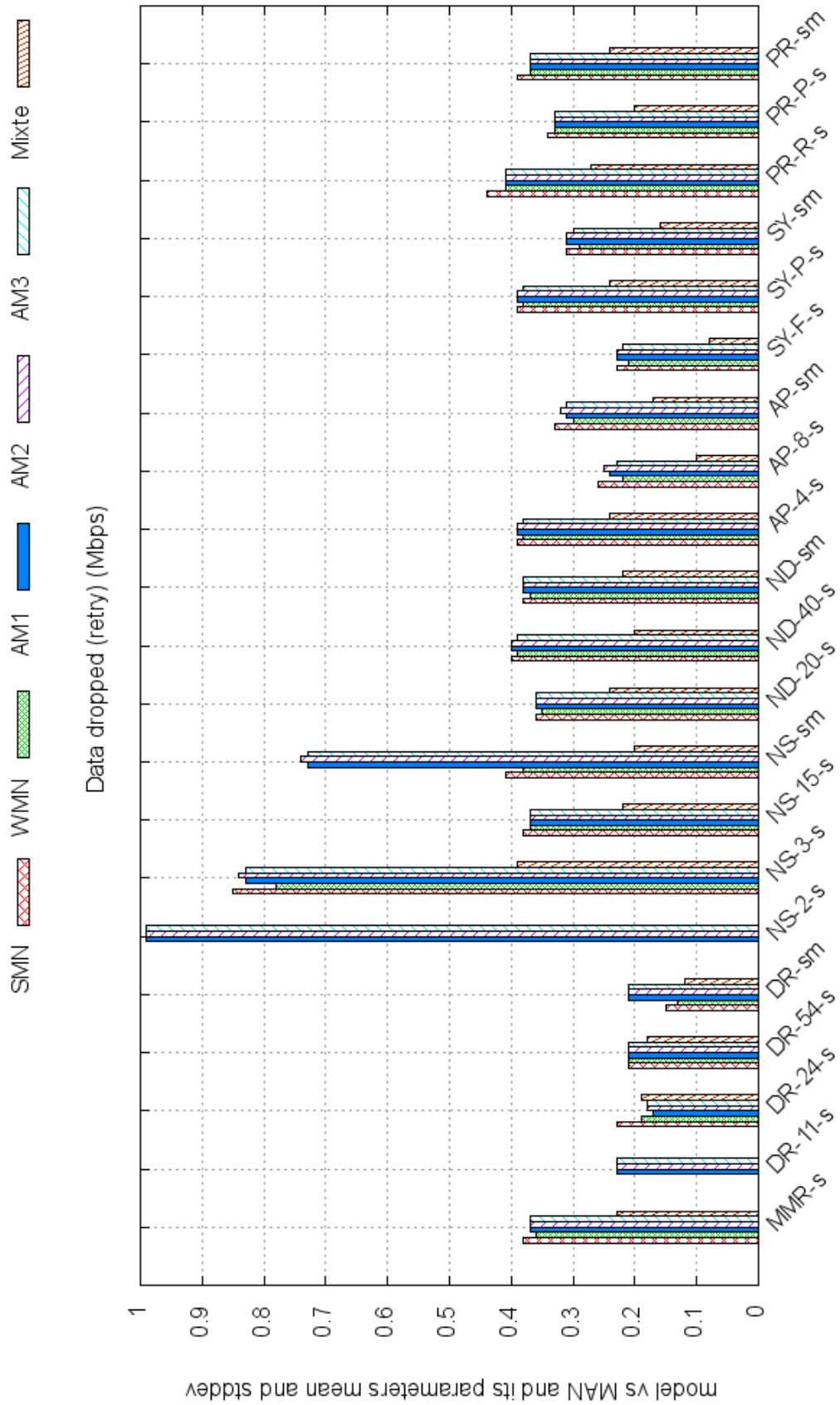


FIGURE 4.7 – Data dropped (retry) (Mbps)

MMR	"Model me mean" to "MAN me mean" Ratio with Model in {SMN,WMN,AM1,AM2,AM3} and me in {Load, Thru, Dlay, DrpR}
MMR-s	stddev of MMR of ALL scenarios
MMR-m	mean of MMR of ALL scenarios
DR-n	Data Rate of n Mbps with n in {11,24,54}
NS-n	Node Speed in range 0-n mps with n in {2,3,15}
ND-n	Network Density of n nodes with n in {20,40}
AP-n	Number of Attraction Points with n in {4,8}
SY-t	Sociability Type with t in {(F)ull,(P)artial}
PR-t	Protocol Type with t in {(R)eactive,(P)roactive}
X	MMR of All scenarios covered by X with X in {DR,NS,ND,AP,SY,PR}
X-n	with n in {relative parameter value, see above}
X-n-s	stddev of X-n
X-sm	mean of all X-n-s

TABLE 4.3 – Légende des graphiques perspective réseau

Pour une variable statistique, nous déroulons la procédure suivante pour une combinaison i des paramètres de la Table 4.3 : $(DR_i, NS_i, ND_i, AP_i, SY_i, PR_i)$:

- Soit m_i , la mesure de cette variable statistique pour le modèle Manhattan ;
- Soit c_i , la même mesure pour le modèle courant ;
- Soit $MMR = \frac{c_i}{c_i + m_i}$, la proportion de c_i sur la somme $c_i + m_i$

Manhattan ne nous sert que de point de référence. Tout autre modèle de la littérature aurait aussi bien convenu.

Le premier des 21 histogrammes de chaque figure donne l'écart type de MMR soit :

$$MMR-s = \sigma_{MMR}$$

Cela permet de mettre en évidence l'influence globale des paramètres sur le modèle considéré. Plus σ_{MMR} est grand, plus le modèle considéré a de l'intérêt, en ce sens où il offre de nombreuses possibilités de paramétrage et donc d'applications.

Les 4 histogrammes suivants parmi les 21 rendent compte de l'influence du paramètre *débit*, *data rate* pour les 3 valeurs qu'il peut prendre : 11, 24 et 54 *Mbits/sec*, puis toutes valeurs

confondues.

Suivent alors les autres paramètres selon le même principe : 1 histogramme par valeur possible puis 1 toutes valeurs confondues. On trouve par ordre d'apparition :

- 4 histogrammes pour *Node Speed*, *NS*, *vitesse des nœuds* ;
- 3 histogrammes pour *Network Density*, *ND*, *densité du réseau* ;
- 3 histogrammes pour *Attraction Point*, *AP*, *point d'attraction* ;
- 3 histogrammes pour *Sociability*, *SY*, *sociabilité* ;
- 3 histogrammes pour *Protocol*, *PR*, *Protocole*.

Une précision concernant la Sociabilité s'impose : le caractère entièrement social (*(F)ull*) signifie que tous les nœuds sont respectueux du modèle de mobilité en vigueur ; tandis que la présence d'éléments en « rébellion » ou asociaux (*(P)artial*) est modélisée par environ 15 % des nœuds qui manquent totalement de respect à cet ordre préétabli.

4.5.5 Interprétation

La Table suivante résume l'« influence » des paramètres selon les prérogatives globales suivantes concernant l'écart global d'un paramètre (dénové par X-s, conférer la Table 4.3 et l'axe des abscisses des graphiques) :

- inférieur à 0,05 : influence nulle ;
- entre 0,05 et 0,08 : influence faible ;
- entre 0,08 et 0,1 : influence moyenne ;
- entre 0,1 et 0,2 influence forte.
- supérieur à 0,2 influence très forte.

	Load	Throughput	Delay	Dropped
Global	faible	forte	forte	très forte
Debit	faible	forte	moyenne	forte
Vitesse	faible	forte	forte	très forte
Densité	faible	moyenne	forte	très forte
Nombre de Points d'Attraction	moyenne	moyenne	moyenne	forte
Sociabilité	faible	moyenne	moyenne	forte
Réactif / Proactif	faible	moyenne	moyenne	très forte

Quand le nombre de Points d'Attraction diminue, la concentration des nœuds augmente. Cela a pour effet de réduire le délai de transmission de bout en bout. En effet, il n'y a pratiquement plus besoin de nœuds intermédiaires. En revanche, nombre de données sont perdues pour cause de bourrage du médium air (Data dropped). L'explication est la suivante : la bande de fréquence étant fixe, l'augmentation de la densité géographique des nœuds résulte en une augmentation du nombre de collisions.

Nous constatons que, mis à part *Load*, la grande majorité des valeurs des paramètres ont une influence moyenne ou forte, voire très forte pour *Data dropped*. Cela démontre l'intérêt de la mixité des modèles de mobilité pour le test des protocoles de routage en ce sens où ceux-ci seront vérifiés pour un spectre d'applications large.

4.5.6 Conclusion

Dans cette section, nous nous proposons d'étudier l'influence des différents paramètres du réseau sur la mixité des modèles, laquelle pouvant être considérée comme particulièrement proche de la réalité vu sa diversité. Les tests de simulation nous donnent raison. Nous constatons en effet que, mis à part 1 mesure particulière, la grande majorité des valeurs des paramètres ont une influence moyenne ou forte, voire très forte pour une autre mesure. Cela démontre l'intérêt de la mixité des modèles de mobilité pour le test des protocoles de routage en ce sens où ceux-ci seront vérifiés sur un spectre large d'applications qui en conséquence, se rapprochent de la réalité.

4.6 Conclusion

Ce chapitre présente de nouveaux modèles de mobilité intégrant la notion de points d'attraction dont l'attractivité peut être :

- constante : *Social Manhattan* ;
- croissante : *Weighted Manhattan* ;
- variable : *Ant Manhattan*.

Les résultats obtenus montrent que la prise en compte des aspects sociaux influence considérablement les performances des protocoles de routage simulés. Les utilisateurs de termi-

naux portables se déplaçant suivant des modalités différentes, nous proposons un modèle de mobilité qui est le mixte des modèles précédents. Ce mixte montre qu'il est très sensible aux différents paramètres utilisés et donc représentatif d'un spectre d'applications large. L'implémentation des modèles de mobilité a donné lieu à une charge de développement logiciel en Opnet Proto-C très importante. Une partie de ce travail est portée en annexe. Du point de vue de la recherche, ce travail a ouvert des voies qui sont au nombre des indices tangibles que nos tests ont mis en exergue. Nous avons, certes, axé nos travaux sur les points d'attraction. Cependant, d'autres démarches sont en cours : les traces bien sûr dont nous traitons en Partie I de certains aspects, et qui sont très prometteuses pour réduire les goulots d'étranglement lors des meetings ou mises à jour logiciel ; les réseaux communautaires et tout ce qu'ils renferment d'éléments qui émeuvent chaque jour le néophyte comme le scientifique des sciences humaines et sociales, pour ne citer que 2 exemples.

4.6. CONCLUSION

Troisième partie

Applications

Chapitre 5

Introduction à la partie III

5.1 Introduction et motivation

Les caractéristiques des réseaux ad hoc mobiles (*Mobile Ad-hoc NETWORKS*, MANET) imposent de nouvelles exigences aux protocoles de routage. Une des caractéristiques les plus importantes est la *dynamique de la topologie* laquelle est une conséquence de la mobilité des nœuds : les nœuds du réseau changent de position fréquemment. Les protocoles de routage doivent donc pouvoir s'adapter rapidement aux changements de topologie. Les nœuds d'un réseau ad hoc sont souvent très limités en termes de ressources. Les protocoles de routage doivent donc également tenter de réduire le trafic de contrôle voire la durée d'allocation des ressources du réseau. La localisation géographique des nœuds (i.e. connaître les coordonnées d'un nœud) a été proposée à des fins d'adressage. Les protocoles de routage basés sur la localisation géographique utilisent cette dernière dans leur prise de décision de routage. En effet, le routage géographique consiste à mettre en œuvre un mécanisme de découverte de l'emplacement d'un nœud particulier : la destination ultime des paquets d'information à transmettre ou un nœud intermédiaire de relaying de ces paquets. Cette détermination de l'emplacement de la destination est la toute première étape de la communication vers cette destination. Pour ce faire, un nœud émetteur peut utiliser un service de localisation car celui-ci permet à l'expéditeur d'un paquet de déterminer l'emplacement de sa destination. Dans ce cas, bien évidemment, les performances du protocole de routage géographique seront fortement liées à celles de son service de localisation.

C'est l'actualité et la pertinence de ce domaine qui nous l'ont fait choisir pour démontrer

l'apport de la prise en compte des aspects sociaux dans les réseaux mobiles.

Nous aurions pu choisir d'autres domaines, moins proches de nous cependant, tels que le routage, l'autoconfiguration, le contrôle de réseaux etc..

5.2 Définition des services de localisation

L'apparition dans les applications de géolocalisation pour les MANET, de l'information de localisation, d'une part, et, d'autre part, le besoin croissant de protocoles de routage adaptés à ces applications, ont mené tout naturellement à l'utilisation de l'information de localisation, désormais disponible, dans les protocoles de routage qui sont alors dits *basés sur la localisation* ou encore *protocoles de routage géographiques*. Les performances et la robustesse à la montée en charge (*scalability*) Mauve et al. [2001] de ces protocoles s'en sont trouvées améliorées. Un tel protocole achemine un paquet en se basant donc sur la localisation géographique du nœud destinataire du paquet. Cette information de location peut lui être délivrée par un *service de localisation*. Ce type de service met en œuvre un mécanisme de suivi des nœuds du réseau, apte à découvrir l'emplacement d'un nœud quelconque à la demande.

La Fig. 5.1 montre la structure générale d'un service de localisation. Celui-ci se compose principalement de trois entités : serveur de localisation, nœud source et nœud destination. Les opérations sont notamment : *demande d'emplacement* et *mise à jour d'emplacement*. Chaque nœud (destination) génère un paquet de *mise à jour d'emplacement* aux serveurs de localisation responsables de gérer ses informations de localisation. Avant d'envoyer un paquet, un nœud (source) envoie une *demande d'emplacement* au serveur de localisation ad hoc et récupère en retour l'emplacement demandé (typiquement celui d'un nœud destination, mais pas obligatoirement). Un nœud donné peut être son propre serveur de localisation. Dans lequel cas, toute *demande d'emplacement* le concernant sera résolue localement.

Un protocole qui met en œuvre un service de localisation efficace, doit présenter les propriétés essentielles suivantes :

- Efficacité : La surcharge induite par les paquets de contrôle (*demande d'emplacement* et *mise à jour d'emplacement*, notamment) doit être la plus réduite possible.

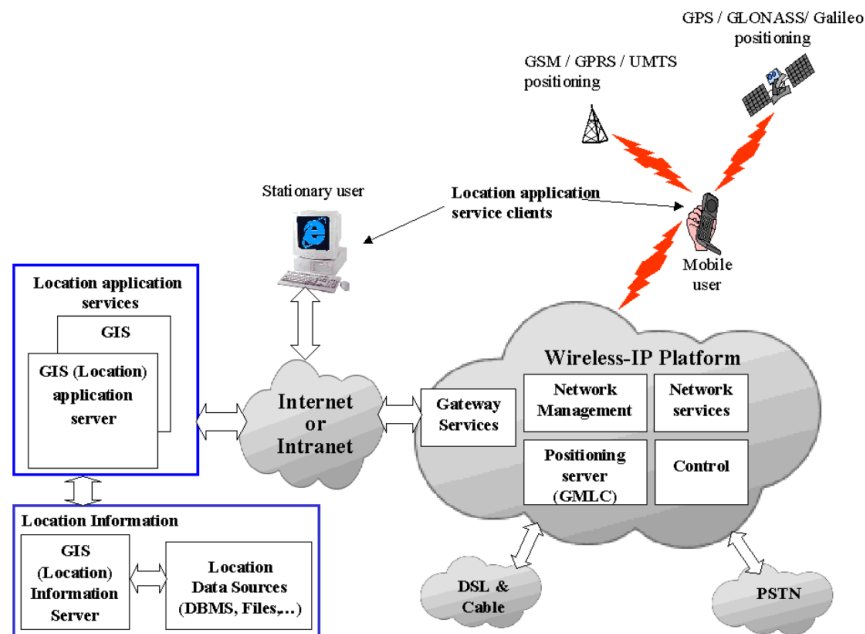


FIGURE 5.1 – Location Service Big Picture

- Robustesse à la panne : Le service de localisation doit prévenir toute panne susceptible de nuire à son bon fonctionnement : grande mobilité des nœuds, nœuds (notamment les serveurs de localisation) en panne ou déconnectés du réseau, par exemple.
- équilibrage des charges : Le protocole doit mettre en œuvre l'équilibrage du trafic afin d'éviter les problèmes de goulot d'étranglement.
- Robustesse à la montée en charge (*Scalability*) : Le protocole doit conserver toutes les propriétés ci-dessus lors de la montée en charge et ce y compris pour un grand nombre de nœuds.
- Proximité : La distance entre un serveur de localisation et un nœud source qui le sollicite ne doit pas excéder celle séparant la source du nœud destination. En d'autres termes, une *demande d'emplacement* émise par un nœud proche de sa destination ne doit pas errer au loin dans le réseau. Toutefois, cette fonctionnalité, pour être efficace, doit prendre appui sur une classification rigoureuse du trafic de données locales.

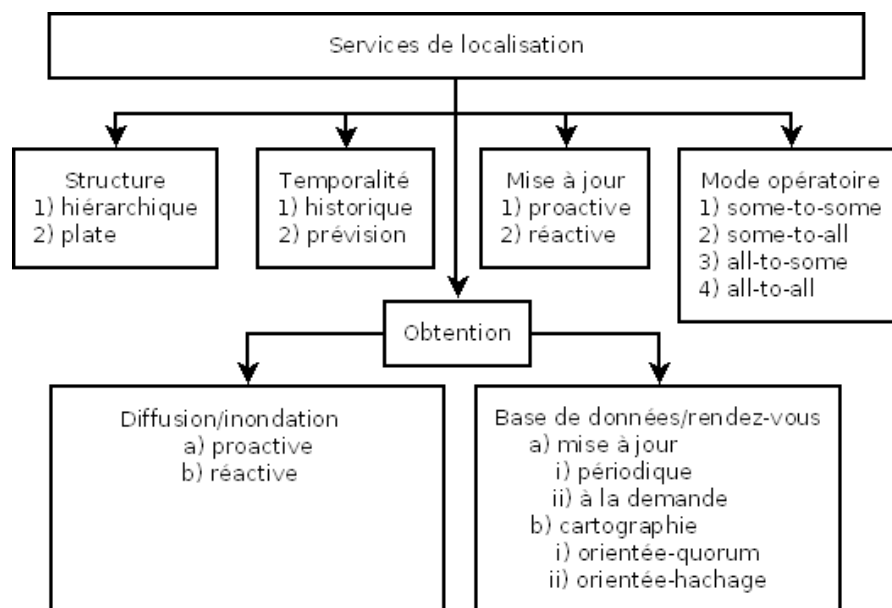


FIGURE 5.2 – Classification des services de localisation

5.3 Classification des services de localisation

Les services de localisation pour les réseaux ad hoc peuvent être classés de différentes manières en fonction des critères suivants :

- *Structure* : selon la structure du réseau, le service de localisation peut être géré de façon hiérarchique ou « à plat ».
- *Temporalité de l'information* : l'emplacement actuel d'un nœud est présent dans l'historique ou il résulte d'un calcul prévisionnel.
- *Mode opératoire* : en fonction de : i) le nombre de nœuds qui hébergent le service i.e. certains ou tous les nœuds sont serveurs de localisation et ii) la quantité d'information que renferme un serveur de localisation : informations de certains ou de tous les nœuds du réseau, le *mode opératoire* est dit : *certaines-pour-certaines*, *certaines-pour-tous*, *tous-pour-certaines* et *tous-pour-tous*.
- *Mise à jour de l'emplacement* : la manière de mettre à jour les informations de localisation peut se faire de façon proactive ou réactive.
- *Obtention de l'emplacement* : l'emplacement actuel d'un nœud est obtenu par *diffusion* à tous les nœuds ou bien à certains nœuds élus pour supporter le système de

base de données distribuée des emplacements.

Ce dernier critère définit donc deux méthodes de mise à disposition des informations de localisation : *diffusion* et *base de données*. Dans la *diffusion*, tout nœud maintient sa propre table d'emplacements et collecte les informations de localisation des autres nœuds en émettant des demandes par *inondation* périodique ou au besoin. Dans la *base de données*, des nœuds sont élus points de rendez-vous et servent de bases de données d'emplacements. Ils mettent à jour leurs données a priori de manière proactive et a posteriori en lançant une recherche pour une localisation urgente.

Un protocole *orienté-inondation* peut à son tour être :

- *proactive* : les nœuds se communiquent les emplacements périodiquement comme dans DREAM Basagni et al. [1998], SLS Camp et al. [2002] ;
- *réactive* : c'est-à-dire à la demande comme dans RLS Camp et al. [2002].

Dans un protocole *orienté-rendez-vous*, en revanche, tous les nœuds suivent une cartographie qui associe l'identifiant unique d'un nœud à un ou plusieurs autres nœuds qui sont en fait ses serveurs de localisation c'est-à-dire les lieux de prédilection : i) de ses mise à jour périodique d'emplacement et ii) des demandes d'emplacement le concernant. Il existe en outre deux façons différentes de réaliser cette cartographie : *orientée-quorum* et *orientée-hachage*.

Dans l'approche fondée sur le *quorum* (Stojmenovic et al. [2008], Liu et al. [2007]), une mise à jour d'emplacement de nœud est envoyée à un sous-ensemble de nœuds disponibles, dit *quorum de mise à jour*. Une demande d'emplacement de nœud est envoyée à un sous-ensemble potentiellement différents dit *quorum de demande*. Les deux sous-ensembles sont conçus de telle sorte que leur intersection est non vide. Une requête sera donc satisfaite par un nœud du *quorum de mise à jour*. Les protocoles *orientés-quorum* peuvent être qualifiés de *non-déterministes* dans la mesure où la sélection des nœuds constitutifs de la base de données (quorum) est non-déterministe.

Dans la cartographie *orientée-hachage* (Kiess et al. [2004]), les serveurs de localisation sont désignés par une fonction de hachage. Celle-ci hache soit l'identifiant unique soit la position d'un nœud et selon le résultat obtenu, décide s'il est serveur de localisation ou pas. Les protocoles correspondants, dits par extension *orientés-hachage*, utilisent donc

5.4. DREAM

une technique déterministe pour sélectionner les serveurs de localisation et peuvent en conséquence être qualifiés de *déterministes*.

La table 5.1 montre quelques-uns des avantages et limitations des services de localisation pour quelques critères décrits précédemment.

TABLE 5.1 – Avantages et limitations des services de localisation

Categorie	Avantages	Limitations
Proactif	Robuste au partitionnement réseau Sans demande (proactif) Conception aisée Gère les distributions non-uniformes	Non évolutif Consommateur de mises à jour
Réactif	Robuste au partitionnement réseau Sans mise à jour Conception aisée Gère les distributions non-uniformes	Non évolutif Consommateur de demande
à quorum	Conception aisée évolutif	Sensible aux erreurs de mise à jour Sensible à la mobilité Effet distance Charge de stockage non-uniforme
Hierarchique	évolutif Effet distance réduit Charge de stockage uniforme équilibrage des charges des demandes/mises à jour	Vulnérable aux erreurs de serveur de localisation Conception complexe Non-robuste au partitionnement réseau

5.4 DREAM

Nous présentons dans cette section le service de localisation *Distance Routing Effect Algorithm for Mobility*, *DREAM* car c'est celui que nous avons choisi pour les tests de simulation de cette partie et la suivante. *DREAM* a été introduit par Basagni et al. [1998] en tant que service de localisation inclus dans un protocole de routage géographique. Par

la suite, Camp et al. [2002] le rebaptise *DREAM Location Service*, *DLS*. *DLS* partage périodiquement l'information de localisation de ses nœuds mobiles par l'envoi de messages de mise à jour qui sont dits *Location Update Message*, *LUP*, lequel contient les coordonnées du nœud source, sa vitesse et l'heure à laquelle l'*LUP* a été transmis. Tout nœud transmet un *LUP* à ces voisins proches à une fréquence élevée. Tandis qu'il envoie à une fréquence moindre aux nœuds distants. En effet, plus 2 nœuds s'éloignent l'un de l'autre, plus l'amplitude perceptible des déplacements de l'un vu par l'autre diminue. Cela est dénommé *effet de distance*, *distance effect*. La partie gauche de la Fig. 5.3 illustre cet effet : du point de vue du nœud *A*, le déplacement du nœud *B* est perçu comme plus important que celui du nœud *C*, alors que dans les 2 cas, la distance parcourue est la même. Cette distinction établie entre les nœuds proches et distants, permet de réduire le trafic de contrôle du réseau. Lorsqu'un nœud émetteur a besoin de connaître la localisation d'un nœud destination, il interroge en premier lieu sa table de localisation. Il y a émission d'un paquet de demande (*request packet*) uniquement dans le cas où l'information n'est pas dans cette table ou a expiré. La demande est diffusée à l'ensemble du réseau (*inondation*, *flooding*). Lorsque le nœud destination reçoit la demande, il y répond en envoyant un paquet de réponse (*reply packet*) contenant sa position. Dès lors qu'une information de localisation est obsolète, elle est supprimée de la table de localisation. Ce protocole renferme un mécanisme de choix du relais suivant (*next hop*) qui utilise un cône d'émission $[-\alpha, +\alpha]$ autour de l'axe émetteur-destination θ . α est calculé en fonction de la vitesse de la destination, cf. la partie droite Fig. 5.3.

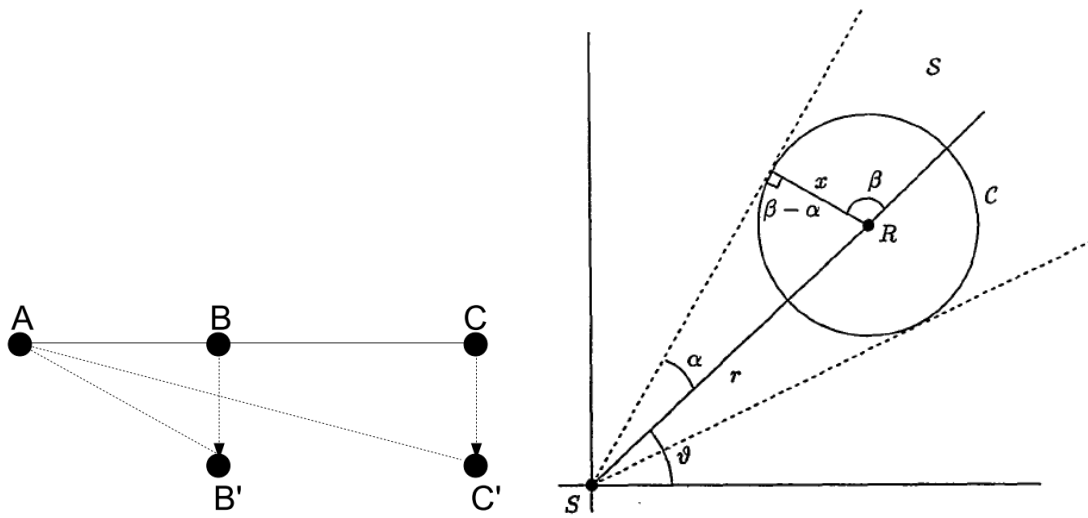


FIGURE 5.3 – DREAM : Distance Effect and Forwarding Cone

Chapitre 6

SFLS Social

6.1 Introduction

Renault et al. [2010a] et Renault et al. [2010b] proposent leur *Service de localisation à semi-inondation* ou *Semi-Flooding Location Service (SFLS)*. *SFLS* vise à réduire le trafic de contrôle¹ en minimisant le nombre de messages de mise à jour traversant le réseau, par diminution de la fréquence de relayage au niveau des nœuds intermédiaires au fur et à mesure que ceux-ci s'éloignent de la source. *SFLS* fournit donc une information de localisation qui est :

- exacte pour les nœuds voisins du nœud qui émet sa position ;
- de moins en moins exacte au fur et à mesure que l'on s'éloigne du nœud émetteur.

Tout cela s'avère lorsque les nœuds sont uniformément répartis sur le réseau. En effet, dans ce cas, l'éloignement est pour ainsi dire « existant par construction ». En revanche, il est clair que l'absence d'éloignement, par exemple dans le cas où les nœuds du réseau sont groupés, a pour conséquence immédiate une chute des performances d'*SFLS*. Cela est le cas en présence d'une communauté, au sens proximité géographique.

Ce chapitre propose de remédier à cette chute de performance. Il ajoute la prise de conscience communautaire, à *SFLS* qui devient alors *S-SFLS* pour *Social SFLS*.

La suite de ce chapitre se compose d'une Sec. 6.2 qui décrit *Semi-Flooding Location Service*, de façon succincte. La Sec. 6.3 présente *S-SFLS* et la manière d'introduire le comportement social en question. Le chapitre outre sa conclusion, analyse les performances

1. message d'administration de l'infrastructure réseau, par opposition au paquet de données

en montrant : i) l'impact communautaire ; ii) la possibilité de prendre en compte cet aspect dans *SFLS*.

Notons que *SFLS* n'a fait l'objet d'aucune implémentation de simulation avant celle proposée ci-après.

6.2 Semi-Flooding Location Service

Le protocole *Semi-Flooding Location Service (SFLS)* permet à tout nœud du réseau de disposer d'une information de localisation « suffisamment » correcte de tous les autres nœuds du réseau. Cela permet d'éviter la surcharge outre mesure du réseau. Cependant, *SFLS* est robuste lors des disparitions de nœud ou l'apparition de problèmes réseau.

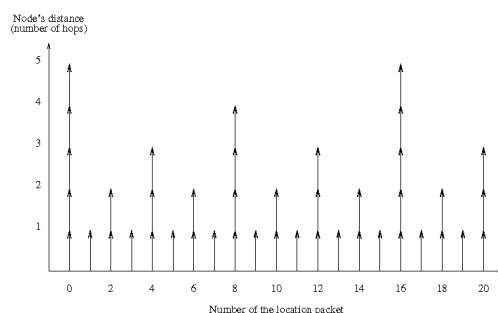


FIGURE 6.1 – Propagation of location packets.

L'idée majeure qui préside au fonctionnement d'*SFLS* est le fait de ne relayer un paquet qu'une fois sur 2 (Fig. 6.1). Un avantage certain de cette méthode consiste en le fait que les paquets de localisation n'inondent plus le réseau de façon aveugle et encombrante. En effet, une quantité moindre atteint réellement l'ensemble du réseau. La plupart des paquets ne font que mettre à jour les voisins à quelques nœuds du nœud émetteur.

La Figure 6.2 (a) présente l'algorithme de diffusion périodique de l'information de localisation. Tandis que la Fig. 6.2 (b) montre comment l'information de localisation des autres nœuds est mise à jour à réception d'un paquet de mise à jour ainsi que le mécanisme de division du relayage.

```

while ( 1 )
    broadcast ( MyID, CurrentPosition, Time )
    sleep ( tau )

```

(a) Periodical broadcast thread.

```

while ( 1 )
    receive ( & id, & pos, & time )
    if node [ id ] is not defined
        node [ id ] . position = pos
        node [ id ] . timestamps = time
        node [ id ] . forwarded = FALSE
    else
        if node [ id ] . timestamps < time
            node [ id ] . position = pos
            node [ id ] . timestamps = time
        if not node [ id ] . forwarded
            broadcast ( id, pos, time )
        node [ id ] . forwarded = not node [ id ] . forwarded

```

(b) Reception thread.

FIGURE 6.2 – SFLS algorithms.

6.3 S-SFLS : SFLS social

S-SFLS, *SFLS social* vise à inclure un certain comportement social dans SFLS afin que les nœuds à l'intérieur d'une communauté puissent être plus conscients de l'emplacement des partenaires de leur communauté que de celui des nœuds en dehors de leur communauté. Deux cas sont donc traités : d'une part, celui des nœuds à l'intérieur d'une même communauté qui échangent « plus » d'informations de localisation, et, d'autre part, celui des nœuds en dehors de leur communauté qui échangent « moins » d'information de localisation. Dans le cadre de cette étude, deux fréquences de mise à jour de localisation sont utilisées : une à l'intérieur d'une communauté, l'autre à l'extérieur de la communauté.

Par exemple, la Fig. 6.3 présente l'algorithme d'*S-SFLS* pour une fréquence $\frac{1}{2}$ à l'intérieur de la communauté, et $\frac{1}{4}$ à l'extérieur de celle-ci. La diffusion périodique quant à elle demeure la même (Fig. 6.2).

6.4 Analyse des performances

Afin d'illustrer l'impact du comportement social des nœuds dans *S-SFLS* par rapport au comportement non social des nœuds de *SFLS*, nous décidons de compter le nombre de messages échangés entre les nœuds à chaque émission de paquet de mise à jour.

```

while ( 1 ) {
    receive ( id, group, position, timestamp )
    if not node [ id ] defined {
        node [ id ] . group = group
        node [ id ] . position = position
        node [ id ] . timestamp = timestamp
        node [ id ] . forwarded = false
        node [ id ] . counter = 0
    }
    else {
        if node [ id ] . timestamp < timestamp {
            node [ id ] . position = position
            node [ id ] . timestamp = timestamp
        }
    }

    if MyGroup = node [ id ] . groupe {
        if not node [ id ] . forwarded {
            broadcast ( id, position, timestamp )
        }
        node [ id ] . forwarded = not node [ id ] . forwarded
    }
    else {
        if node [ id ] . counter mod 4 = 0 {
            broadcast ( id, position, timestamp )
        }
        node [ id ] . counter = node [ id ] . counter + 1
    }
}

```

FIGURE 6.3 – S-SFLS algorithm.

La Table 6.1 présente les paramètres et variables utilisés dans la suite de cette section.

6.4.1 Distribution des nœuds

La répartition des nœuds dans le réseau a un impact important sur les performances. Si presque tous les noeuds sont situés dans une zone réduite du réseau, les nœuds en dehors de celle-ci seront probablement déconnectés du reste du réseau conduisant à beaucoup de perte de paquets. Dans ce qui suit, on suppose que les noeuds sont répartis uniformément sur le réseau. La Fig 6.4 montre comment le réseau a été divisé autour de tout nœud pour déterminer la portion des nœuds à un certain « nombre de sauts » de n'importe quel nœud émetteur.

En supposant les noeuds répartis uniformément sur le réseau, la portion des nœuds à un certain « nombre de sauts » de n'importe quel nœud émetteur est donnée par :

$$p_i^{(j)} = \begin{cases} \frac{1}{N} & j = 0 \\ \frac{(N-1)(2j-1)}{Nl_i^2} & 1 < j < l_i \end{cases} \quad (6.1)$$

Parameters	
N	the number of nodes in the network
r	the transmission range of the nodes in meters
l_i	the length of the longest path from node i to any other nodes in hops
$a(l_i)$	l_i outside the community
$b(l_i)$	l_i inside the community

Nodes' distribution	
$p_i^{(j)}$	the ratio of the number of nodes located at j hops away from i

Variables	
m_i	the mean number of messages sent for each location update emitted by i

TABLE 6.1 – Definition of parameters and variables.

6.4.2 Évaluation d'SFLS sans communauté

En utilisant une distribution uniforme, le nombre total moyen de messages émis pour la mise à jour d'une information de localisation est donné par :

$$m_i = \frac{1}{l_i} \left[\left(2 + \frac{3}{l_i} \right) \left(1 - \frac{1}{2^{l_i}} \right) - 2 \right] (N - 1) + 1 \quad (6.2)$$

Soit $a(l_i)$, le facteur de réduction du relayage, c'est-à-dire, de réduction du nombre moyen de messages émis après la mise à jour du nœud i . $a(l_i)$ s'écrit comme suit :

$$a(l_i) = \frac{1}{l_i} \left[\left(2 + \frac{3}{l_i} \right) \left(1 - \frac{1}{2^{l_i}} \right) - 2 \right] \quad (6.3)$$

6.4.3 Évaluation d'SFLS avec communauté

En introduisant la notion de communauté dans *SFLS*, nous maintenons la même densité de nœuds par unité d'aire à l'extérieur de la communauté que celle que nous avons sans communauté. En conséquence, l'Eq. (1) doit être amendée afin de relater ce fait. Soit G_i , le nombre de nœuds de la communauté i . Le fait que seuls les nœuds en dehors de

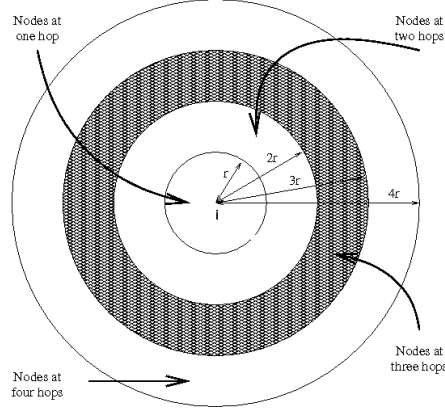


FIGURE 6.4 – Distribution of neighbours around a node.

la communauté soient uniformément distribués, exige de remplacer le facteur $(N - 1)$ de l'Eq. (1) par $(N - G_i)$ et une expression supplémentaire doit être ajoutée, afin de compléter la distribution et tenir compte des nœuds de la communauté. Par conséquent, la distribution des nœuds qui rend compte de la présence d'une communauté est :

$$p_i^{(j)} = \begin{cases} \frac{1}{N} & j = 0 \\ \frac{(N - G_i)(2j - 1)}{Nl_i^2} + 2\frac{(G_i - 1)(l_i - j + 1)}{N(l_i + 1)l_i} & 1 < j < l_i \end{cases} \quad (6.4)$$

Aussi, le nombre moyen de messages devient :

$$\begin{aligned} m_i^{(c)} &= \frac{1}{l_i} \left[\left(2 + \frac{3}{l_i} \right) \left(1 - \frac{1}{2^{l_i}} \right) - 2 \right] (N - G_i) \\ &\quad + \frac{2}{(l_i + 1)l_i} \left(l_i - 1 + \frac{1}{2^{l_i}} \right) (G_i - 1) + 1 \end{aligned} \quad (6.5)$$

En utilisant $a(l_i)$ comme défini plus haut et en introduisant $b(l_i)$ comme facteur de réduction du relayage (ou du nombre moyen de messages émis) dans la communauté, m_i s'écrit maintenant comme suit :

$$m_i^{(c)} = a(l_i) \times (N - G_i) + b(l_i) \times (G_i - 1) + 1 \quad (6.6)$$

où $b(l_i)$ s'écrit ainsi :

$$b(l_i) = \frac{2}{(l_i + 1)l_i} \left(l_i - 1 + \frac{1}{2^{l_i}} \right) \quad (6.7)$$

6.4.4 Impact des communautés sur SFLS

La Figure 6.5 montre la position relative de $a(l_i)$ et $b(l_i)$ lorsque le nombre maximal de sauts entre le nœud émetteur d'un paquet de mise à jour et les autres nœuds, augmente. Cela souligne clairement le fait que les communautés ont un impact important sur les performances : à l'intérieur de la communauté, les nœuds sont plus conscients de l'emplacement des autres nœuds de la communauté de par la diminution plus lente du facteur de réduction que pour les nœuds en dehors de la communauté. Pour des longueurs maximales de chemin plus importantes, le rapport entre les nœuds à l'intérieur et à l'extérieur atteint une valeur de 6.

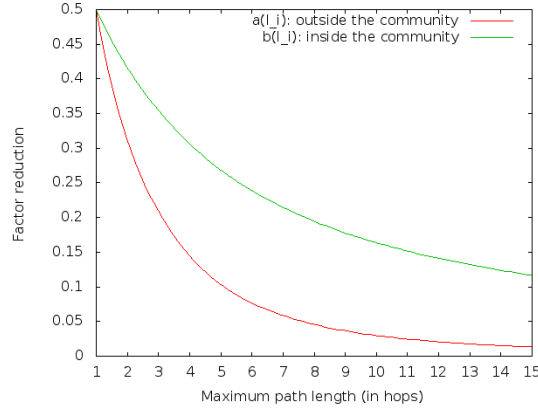


FIGURE 6.5 – Factor reduction vs. maximum path length for SFLS.

6.4.5 Évaluation de S-SFLS

L'évaluation du nombre moyen de messages envoyés lors de l'utilisation de S-SFLS est nécessairement relative à la distribution des nœuds (cf. Eq. 6.4). L'expression du nombre moyen de messages est similaire à celle présentée dans l'Eq. 6.6. $B(l_i)$ reste identique. En revanche, le facteur de réduction du nombre moyen de messages envoyés à chaque mise à jour, change. Soit $m_i^{(s)}$, le nombre moyen de messages envoyés via *S-SFLS* chaque fois qu'un paquet de mise à jour est émis par un nœud. $m_i^{(s)}$ peut s'écrire ainsi :

$$m_i^{(s)} = a^{(s)}(l_i) \times (N - G_i) + b(l_i) \times (G_i - 1) + 1 \quad (6.8)$$

En introduisant de nouveau le ratio entre la fréquence des mises à jour à l'extérieur de la communauté et celle à l'intérieur de cette dernière comme présenté par la Fig. 6.3, en extrayant $a^{(s)}(l_i)$ de $m_i^{(s)}$, il vient :

$$a^{(s)}(l_i) = \frac{1}{l_i^2 2^{2l_i}} (2^{2l_i} - 2l_i - 1) \quad (6.9)$$

6.4.6 Comparaison entre SFLS et S-SFSL

En traçant $a(l_i)$ et $a^{(s)}(l_i)$ sur le même graphique, comme le montre la Fig. 6.6, il apparaît clairement que le facteur de réduction d'*S-SFSL* convient davantage aux communautés que celui du *SFSL* original. Le ratio entre ces 2 derniers est de l'ordre de 1 sur 2.

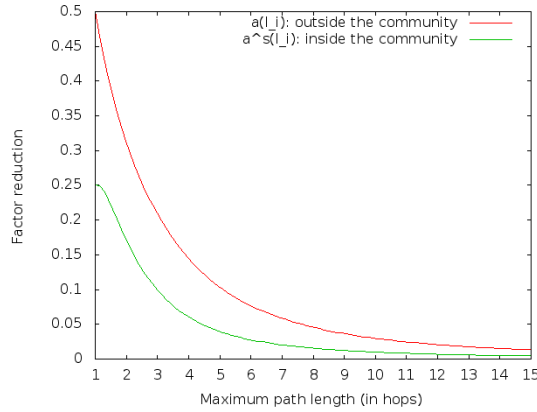


FIGURE 6.6 – Factor reduction vs. maximum path length with communities.

6.5 Simulation

6.5.1 Introduction

Notre objectif dans cette section est de réaliser à nouveau les évaluations précédentes, à savoir :

- Évaluation d'SFSL sans communauté ;
- Évaluation d'SFSL avec communauté ;
- Impact des communautés sur SFSL ;
- Évaluation de S-SFSL.

TABLE 6.2 – Simulation

Simulateur	Network Simulator 2 (NS2)
Couche MAC	IEEE 802.11
Taux de transmission	100 m
Type de trafic	peer-to-peer 64 bytes packets
Destination	aléatoire
Caractéristiques physiques	914MHz Lucent WaveLAN DSSS
Ad hoc protocole	AODV
Vitesse des nœuds (m/s)	2 (piéton), 3 (bicyclette), 15 (véhicule)
Nombre de nœuds	50, 100
Aire de simulation	600 m $\times$ 300 m
Protocole	Dream, SFLS-Half, SFLS-Third, SFLS-Quater
Nombre de groupes	1 ou 4

Nous rapprochons les modèles *SFLS* et *S-SFLS* du modèle de serveur de localisation à inondation le plus courant dans la communauté scientifique, à savoir *Dream* pour mettre en perspective les apports de ces modèles.

En outre, *SFLS* et *S-SFLS* sont déclinés en fonction du diviseur d'émission : 1 fois sur 2 (half), 1 fois sur 3 (third) et 1 fois sur 4 (quater). Les diviseurs supérieurs à 4 n'ont pas montré d'intérêt particulier car le gain en messages de mise à jour est dans ce cas perdu en couple de messages (*request,reply*).

Les différentes valeurs de nos paramètres de simulation sont résumées en Table 6.2.

6.5.2 Évaluation d'SFLS sans communauté

En premier lieu, sont présentés les résultats concernant un seul groupe c'est à dire une seule communauté, en ce sens où cela revient à ne pas tenir compte de l'aspect social : Fig. 6.7 et 6.8.

La Fig. 6.7 :

- concerne le mode de transport ou vitesse de nœud *pédestre* pour une densité de réseau de 50 et 100 nœuds ;
- donne en ordonnée le pourcentage de paquets émis par rapport aux paquets reçus.

En d'autres termes, il s'agit du nombre de paquets émis pour cent paquets reçus en

moyenne sur l'ensemble des nœuds ;

- présente 2 mesures et leur cumul :
 - Ctrl : paquets de contrôle ;
 - Data : paquets de données ;
 - All : cumul paquets de controle + paquets de données.

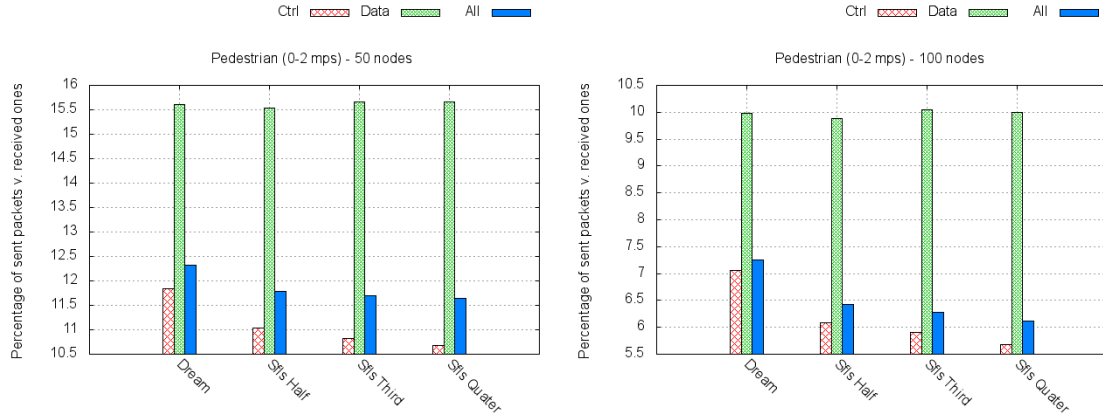


FIGURE 6.7 – SFLS - One Group - Pedestrian (0-2 mps) - 50/100 nodes

Nous constatons que le trafic de contrôle :

- est moindre en passant de Dream à SFLS ;
- est proportionnel au diviseur de relayage ;
- passe de 10,5-15,5 % à 5-10 %, en passant de 50 à 100 nœuds.

Un gain évident de ressources est donc constaté avec un protocole qui est en définitive bien plus simple à mettre en œuvre que Dream. En outre, nous constatons que *Dream* émet un trafic de contrôle totalement superflus de l'ordre de 1 à 1,5 % par rapport à *SFLS*. Ce même gain est également observable quoique moindre entre *SFLS-Half* et *SFLS-Third* puis *SFLS-Quarter*.

Rappelons que le trafic de contrôle concerne autant les messages de mise à jour (*Update packets*) que les échanges de demande/réponse (*Request packets and Reply packets*).

Les tendances sont pratiquement équivalentes pour le mode de transport ou vitesse de nœud *bicyclette* pour une densité de réseau de 50 et 100 nœuds : nous avons choisi de ne pas présenter ces graphiques afin de ne pas encombrer inutilement, cette section. En revanche, nous présentons Fig. 6.8, le mode de transport ou vitesse de nœud *car* pour une

densité de réseau de 50 et 100 nœuds, afin de montrer que les ratio sont plus importants consécutivement à l'accroissement de vitesse.

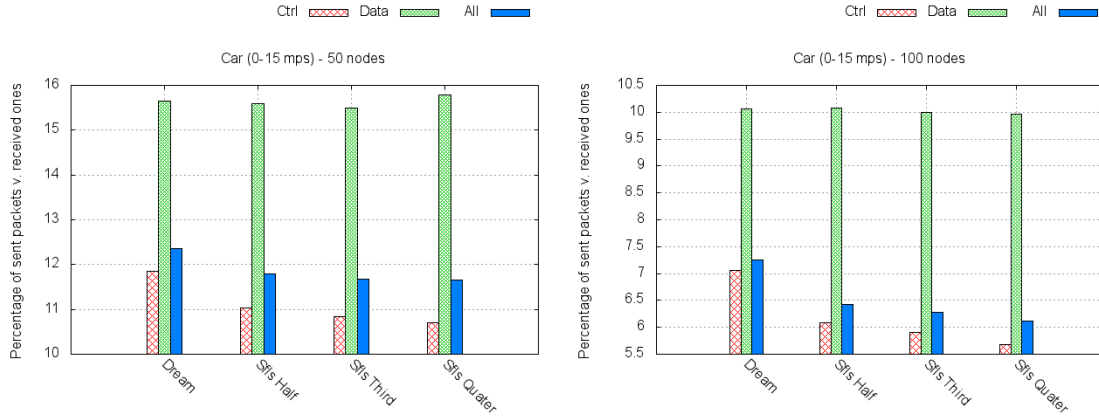


FIGURE 6.8 – SFLS - One Group - Car (0-15 mps) - 50/100 nodes

6.5.3 Évaluation d'SFLS avec communauté

Afin de connaître l'impact de la présence de communautés, les résultats du comportement du protocole *SFLS* pour 4 groupes i.e. 4 communautés, sont exposés Fig 6.9.

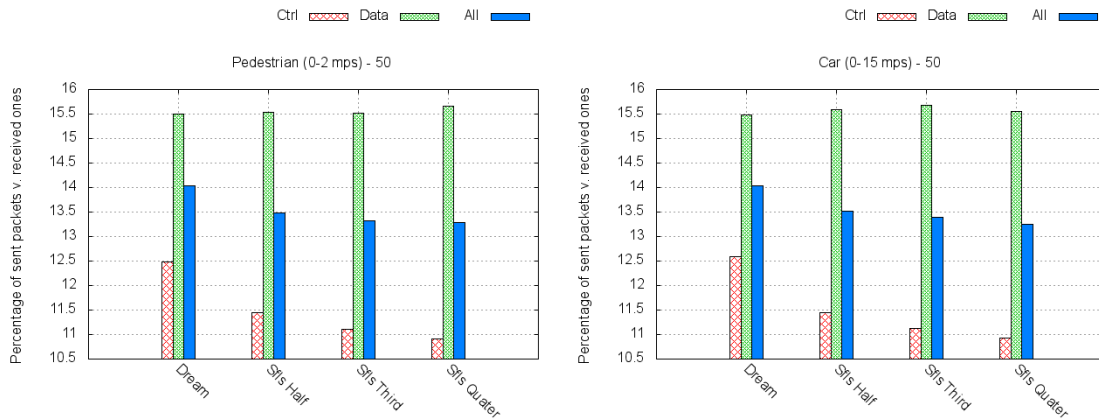


FIGURE 6.9 – SFLS - 4 Groups - Pedestrian (0-3 mps)/Car (0-15 mps) - 50 nodes

Nous constatons que le trafic de contrôle est plus important d'environ 0,5 %.

6.5.4 Évaluation de S-SFLS

Enfin, but ultime, nous montrons comment *S-SFLS* pallie à la chute de performance d'*SFLS* en présence de communautés. Dans cette optique, les résultats du comportement du protocole *S-SFLS* pour 4 groupes i.e. 4 communautés, sont exposés : Fig. 6.10.

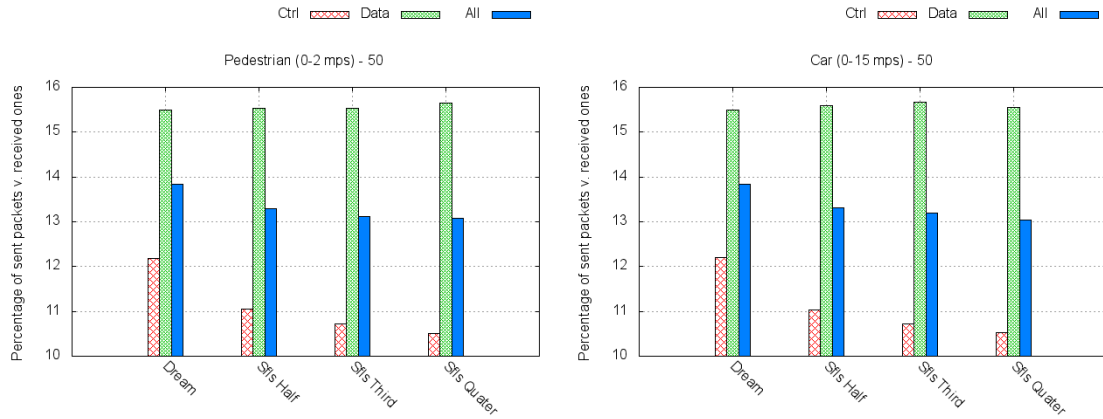


FIGURE 6.10 – S-SFLS - 4 Groups - Pedestrian (0-3 mps)/Car (0-15 mps) - 50 nodes

Nous retrouvons un trafic de contrôle conforme à celui que nous avons sans communauté en section 6.5.2.

6.6 Trajectoires modélisées

Nous utilisons dans cette section les trajectoires de 50 nœuds générées par Opnet Modeler en utilisant notre Modèle de Mobilité *Social Manhattan*. L'outil de conversion d'Opnet vers Network Simulator 2 est écrit en *Microsoft PowerShell*[©], par nos soins et est fourni en annexe (E).

La topologie suit celle native utilisée dans les tests de Dream sous *NS2* à savoir 300 m x 600 m. Les 50 nœuds sont décomposés en 4 groupes de 12 nœuds. 2 nœuds sont libres c'est-à-dire non soumis au modèle de mobilité en vigueur au centre du réseau. Les 4 groupes sont placés au 4 coins du réseau et doivent rejoindre chacun leur point d'attraction qui est situé à l'angle opposé en suivant la diagonale. Ainsi, les regroupements sont effectués indirectement de par le fait que les membres d'un groupe poursuivent le même objectif.

6.7 Conclusion

Ce chapitre relate la prise en compte d'un élément du social que représente l'appartenance à un groupe communautaire, dans le sens du regroupement géographique de ses membres, dans *Semi-Flooding Location Service*, *SFLS*. Le comportement de base d'*SFLS* est maintenu dans *Social SFLS*, *S-SFLS* à l'intérieur d'un groupe communautaire. À l'extérieur de celui-ci, le nombre de relayage de l'information de localisation est divisé par 4. Ce nouvel aspect est intégré au modèle. Il est ensuite implémenté en *C++*, sous *Network Simulator 2*, *NS2* en se basant sur le *service de localisation DREAM* qui fournit un cadre de travail simple et efficace pour le test des algorithmes d'inondation, *flooding*.

Du point vue des résultats, du côté du modèle, nous constatons que les pertes de gain en termes de trafic de contrôle, occasionnées par la présence de groupes communautaires, sont largement compensées par la réduction du nombre de relayage de l'information de localisation à l'extérieur des groupes communautaires. Du côté simulation, nous constatons qu'à nombre de paquets de données équivalents, nous avons un nombre de paquets de contrôle :

- moindre en passant de Dream à SFLS ;
- qui est proportionnel au diviseur de relayage.

Les résultats de l'évaluation du modèle, d'une part, et, d'autre part, ceux issus de la simulation vont donc dans le même sens :

Un gain évident de ressources est constaté avec un protocole qui est en définitive bien plus simple à mettre en œuvre que Dream.

Un évolution possible dans l'évaluation côté modèle de *Social SFLS* consiste en le fait d'utiliser une courbe non linéaire pour décrire l'amoindrissement de la densité du réseau en fonction de l'éloignement du nœud i (le nœud qui transmet sa position) défini en nombre de sauts *hops*. Les pertes de gain d'*SFLS* sont en effet moindres dans ce cas.

Nous souhaitons rappeler que les attraits de *SFLS* sont sa simplicité de faisabilité et mise en œuvre. Cela ne doit pas pour autant faire oublier que la raréfaction de l'actualisation des positions avec l'éloignement mutuel des nœuds peut être incompatible avec les contraintes de Qualité de Service, imposées par le cahier des charges du réseau sans fil. Néanmoins, il

6.7. CONCLUSION

est clair que le partitionnement du réseau en communauté ne peut qu'être profitable à la Qualité de Service.

Chapitre 7

SLS Social

7.1 Introduction

Dans ce chapitre, nous faisons état d'un travail de recherche supplémentaire qui met en évidence combien la prise en compte de l'aspect social des relations entre utilisateurs peut intervenir dans les retranchements les plus cachés des protocoles de routage ou plus exactement dans ce que nous avons décrit comme étant l'enjeu de demain pour les protocoles de routage, à savoir, les services de localisation. En effet, il n'est pas inutile de rappeler que les protocoles de routage sont responsables de l'acheminement le plus rapide possible entre un émetteur et son destinataire. Dans les réseaux sans fil, cela est un véritable défi de par les mouvements incessants des nœuds, notamment des terminaux portables des utilisateurs. Les services de localisation permettent aux protocoles de routage de déterminer au niveau du nœud émetteur *où* se trouve son destinataire et donc de ne diffuser l'information que dans cette direction. Cette diffusion peut se faire directement à son destinataire s'il est à sa portée. Si ça n'est pas le cas, la diffusion concernera le nœud voisin le plus susceptible de rapprocher l'information de son destinataire final. Dans cette optique, le fait que les nœuds soient dans les délais les plus brefs possibles au courant de la position de leur destinataire représente l'enjeu le plus important. C'est la raison pour laquelle nous tenons pour vrai et démontrons par simulation, l'assertion selon laquelle l'adjonction du social dans les services de localisation a un impact certain sur ces derniers et par voie de conséquence sur les protocoles de routage sous-jacents. L'impact sur les services de localisation s'exprime en termes de réduction des demandes de mise à jour des positions des nœuds.

La notion du social que nous mettons en œuvre est celle de groupe d'appartenance. Dans ce chapitre, nous démontrons par simulation, qu'un nœud appartenant à un groupe social est plus rapidement à jour du point de vue de la localisation des nœuds du réseau qu'un nœud n'appartenant pas à un groupe social. Nous utilisons pour l'illustrer le *service de localisation Simple Location Service, SLS* (Basagni et al. [1998]).

La suite de ce chapitre se compose de la description d'*SLS* (Sec. 7.3) puis de son pendant social (Sec. ??) avant de l'évaluer (Sec. 7.4) et de conclure.

7.2 SLS

Le protocole *Simple Location Service, SLS*, du point de vue classification (Sec. 5.3) est du type :

- Structure plate ;
- Temporalité par historique ;
- Mode opératoire all-to-all ;
- Obtention de l'information par inondation ;
- Mise à jour périodique.

Dans *SLS*, un nœud dispose d'une *table de localisation, LOCTABLE* qui contient l'ensemble des positions des nœuds du réseau. Il transmet un *paquet de localisation ou location packet, LP* à ses voisins à un intervalle de temps qui est une fonction de la vitesse de ses changements de position géographique. Cette période s'exprime par :

$$P = \frac{T_{range}}{\alpha} \times \frac{1}{\nu} = \frac{T_{range}}{\alpha \nu}$$

au moins toutes les Z secondes avec :

- T_{range} : Transmission range : portée radio ;
- $\frac{1}{\alpha}$: coefficient de pondération de T_{range} ;
- ν : vitesse de déplacement du nœud.

En outre, un *LP* contient jusqu'à E entrées issues de la *table de localisation* du nœud qui l'émet ; ces E entrées sont choisies selon un schéma *round-robin* : dès lors que le dernier a été transmis, on retransmet le premier. En d'autres termes, un nœud « profite » de

l'opportunité de transmission offerte par un *LP* pour partager avec ses homologues, sa « vision » du réseau en termes de localisation de ses nœuds. Dès lors que suffisamment de *LP* ont été transmis, toute l'information de localisation dont un nœud dispose est transmise à l'ensemble de ses voisins dans le réseau. Nous présentons ci-après sous forme de pseudo-code, l'algorithme correspondant.

```

////////////////////////////////////
// Transmission d'un LP (LocationPacket)
////////////////////////////////////
L = 0 //Initialiser l'indice dans LOCTABLE de la prochaine position à envoyer
TANTQUE VRAI FAIRE
    TANTQUE min(P,Z) secondes non écoulées FAIRE
        Dormir 1 seconde
    FINTANTQUE
    Initialiser le LP avec ma position
    POUR i ALLANT DE 1 À E FAIRE
        L = (L + i) modulo LOCTABLESIZE // modulo implémente Round-Robin
        Mettre dans LP la location LOCTABLE[L]
    FINPOUR
FINTANTQUE

```

Un nœud utilisant *SLS* reçoit périodiquement un *LP* d'un de ses voisins. Le nœud met alors à jour sa table de localisation en se basant sur les entrées de la table reçue, de sorte que chaque information de localisation soit la plus récente possible (par comparaison entre la table locale et la table reçue). Nous présentons ci-après sous forme de pseudo-code, l'algorithme correspondant.

```

////////////////////////////////////
// Réception d'un LP (LocationPacket)
////////////////////////////////////
TANTQUE VRAI FAIRE
    TANTQUE LP non reçu FAIRE

```

Dormir 0,1 seconde

FINTANTQUE

Lire dans *LP* la position *P* du nœud tiers *id*

SI P.heure > LOCTABLE[P.id].heure ALORS

LOCTABLE[P.id] = P

FINSI

TANTQUE il reste *P* dans *LP* non encore traité *FAIRE*

SI P.heure > LOCTABLE[P.id].heure ALORS

LOCTABLE[P.id] = P

FINSI

FINTANTQUE

FINTANTQUE

SLS dispose de quelques similarités avec *RIP*, le *Routing Information Protocol*. En particulier, tous deux transmettent des tables périodiquement à leurs voisins. Les différences entre ces 2 protocoles sont les suivantes : *SLS* envoie partiellement sa table de localisation tandis que *RIP* l'envoie systématiquement entièrement. Par ailleurs, à la différence de *RIP*, un nœud utilisant *SLS* utilise sa table de localisation locale en vue du calcul de sa nouvelle table de localisation. Enfin, *RIP* n'a pas été développé pour les réseaux ad hoc. C'est pourquoi, *SLS* partage sa table avec different voisins au fur et à mesure de l'évolution topologique du réseau ad hoc. *RIP* ne réalise pas ce type d'adaptation. En effet, *RIP* tout comme *OSPF*, dispose d'une vitesse de mise à jour qui n'est basée que sur le temps et non la distance parcourue.

7.3 Social

Comme nous l'avons introduit, l'aspect social que représente la notion de groupe d'appartenance, peut à nouveau trouver un champ applicatif intéressant dans *SLS*.

L'idée consiste, pour un nœud donné, à transmettre aux nœuds de son groupe d'appartenance l'ensemble de sa table de localisation plutôt qu'un fragment de celle-ci. Pour

les autres nœuds, le traitement est identique à SLS classique, seul un fragment de la table de localisation est transmis. Ainsi, les membres d'un même groupe d'appartenance sont mieux informés de leurs positions respectives que les nœuds étrangers à leur groupe. Cela correspond à une économie importante en termes de paquets de contrôle émis et donc en énergie. Ci-après, nous présentons sous forme de pseudo-code, l'algorithme de transmission correspondant.

```

////////////////////////////////////
// Transmission d'un LP Social
////////////////////////////////////

```

TANTQUE VRAI FAIRE

TANTQUE $\min(\frac{T_{range}}{\alpha v}, Z)$ secondes non écoulées *FAIRE*

Dormir 1 seconde

FINTANTQUE

Initialiser le *LP* avec ma position

SI *Dest.groupe* == *monGroupe* *ALORS*

END = LOCTABLESIZE

ELSE

END = MAXLOCINLP

FINSI

POUR i ALLANT DE 1 À END FAIRE

$L = (L + i) \text{ modulo } LOCTABLESIZE$ // Le modulo implémente l'algorithme Round-Robin

Mettre dans *LP* la location *LOCTABLE*[*L*]

FINPOUR

FINTANTQUE

7.4 Evaluation d'S-SLS

7.4.1 Introduction

Nous nous attachons dans ce qui suit à démontrer l'apport de l'adjonction du social au niveau du *serveur de localisation* et non au niveau du *protocole de routage*. En d'autres termes, nous signifions bien clairement qu'il ne s'agit pas ici de démontrer un gain en termes de *délai d'acheminement des paquets* ou de ratio $\frac{received_packets}{sent_packets}$.

Il n'est question ici que de 3 types de paquets qui sont :

- *update* : un paquet de mise à jour : il est transmis à l'initiative d'un nœud qui, suite à un déplacement, veut informer les autres nœuds de son réseau sans fil, qu'il a changé de position ;
- *request* : un paquet de demande : il est transmis par un nœud qui ne dispose pas dans sa table de localisation de la position du nœud auquel il souhaite envoyer une information ;
- *reply* : un paquet de réponse : il est transmis par un nœud en réponse d'un paquet de demande lorsque ce nœud dispose de l'information de localisation du nœud destination recherché par le demandeur.

Nous démontrons que la prise en compte de l'appartenance à un groupe social, permet de réduire le nombre de couples (*request*, *reply*) à l'intérieur d'un groupe de par le fait que ses membres se transmettent entre eux l'intégralité de leur table de localisation à chaque fois qu'ils se transmettent leurs propres positions.

Notre objectif dans cette section est donc d'évaluer l'influence de notre modèle *S-SLS* sur ses performances en passant d'un groupe d'appartenance à 2 groupes d'appartenance.

- SLS de base : une seule communauté ou pas de communauté du tout, ce qui revient au même : la notion de groupe d'appartenance n'a pas de sens de ce cas ;
- SLS Social : plusieurs communautés : la notion de groupe d'appartenance prend tout son sens et devient un élément déterminant dans la diffusion de l'information de localisation.

TABLE 7.1 – Simulation

Simulateur	Network Simulator 2 (NS2)
Couche MAC	IEEE 802.11
Taux de transmission	100 m
Type de trafic	peer-to-peer 64 bytes packets
Destination	aléatoire
Caractéristiques physiques	914MHz Lucent WaveLAN DSSS
Ad hoc protocole	AODV
Densité	50 ou 100 nœuds
Aire de simulation	600 m $\times$ 300 m
Taille Max de la table des positions	100 positions
Portion de la table des positions envoyée	de 10 à 100 positions
Nombre de groupes	1 ou 2

7.4.2 Simulation

Nous avons souhaité visualiser l'avantage de la prise en compte de cette notion de groupe d'appartenance au travers de l'implémentation de notre SLS Social sous *Network Simulator 2*, *NS2* qui dispose de la version de base d'SLS.

Nota bene : le protocole de routage utilisé dans ce chapitre est *SLC* (Basagni et al. [1998]). C'est ce dernier qui se sert de *SLS* comme de service de localisation.

Nous déclinons nos tests selon les critères suivants :

- densité du réseau mobile ;
- taille de la table des positions ;
- nombre de groupes.

Les paramètres de simulation sont ceux qui nous permettent la mise en perspective de notre modèle. Les valeurs choisies nous apparaissent suffisantes dans un premier temps pour connaître l'influence des paramètres correspondants sur le modèle étudié. Il s'agit de :

- taille de la table des positions : de 10 à 100 positions par pas de 10 ;
- nombre de nœuds : 50 nœuds, un groupe moyen et 100 nœuds, un groupe plus important ;
- nombre de groupes : 1 ou 2.

Les différentes valeurs de nos paramètres de simulation sont résumées en Table 7.1.

7.4.3 Résultats

Nous effectuons les simulations qui suivent avec 50 nœuds.

Les Fig. 7.1 et 7.2 donnent les résultats pour :

- SLS de base : une seule communauté ou pas de communauté du tout, ce qui revient au même : la notion de groupe d'appartenance n'a pas de sens de ce cas ;
- SLS Social : plusieurs communautés : la notion de groupe d'appartenance prend tout son sens et devient un élément déterminant dans la diffusion de l'information de localisation.

Nous remarquons qu'en proportion exprimée en pourcentage, les *requests* et les *replies* décroissent en passant de *SLS* à *SLS Social* avec le nombre d'entrées dans *LP* et qu'ils s'équilibrent dès lors que le nombre d'entrées dans *LP* augmentent. Cela est conforme à notre intuition initiale. En d'autres termes, pour des petits nombres d'entrées dans *LP*, *SLS Social* offre de meilleures performances et pour des nombres d'entrées importants dans *LP*, *SLS Social* et *SLS* se valent. En conséquence, il est plus avantageux d'utiliser *SLS Social*.

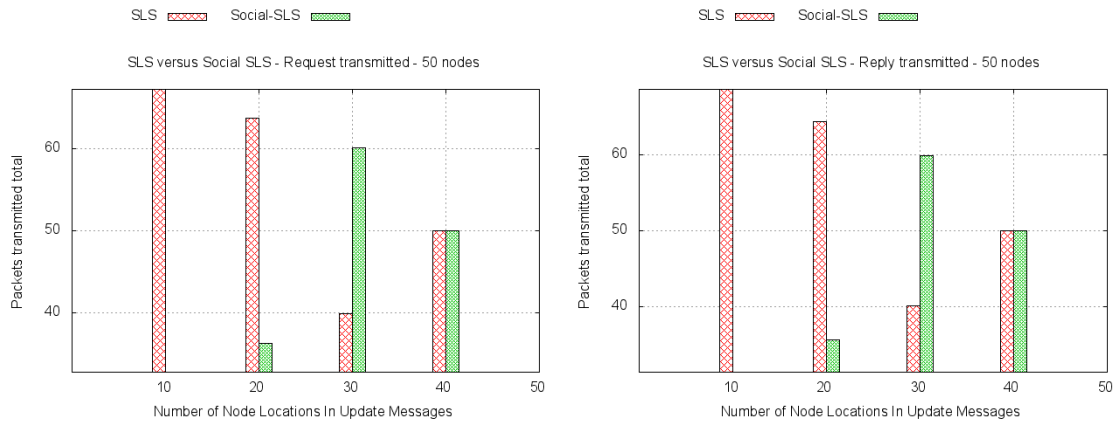


FIGURE 7.1 – SLS versus Social SLS - Request/Reply transmitted - 50 nodes

Nous répétons ces simulations avec 100 nœuds et trouvons des résultats similaires. Par conséquent, nous pouvons dire que les performances ne dépendent pas de la densité.

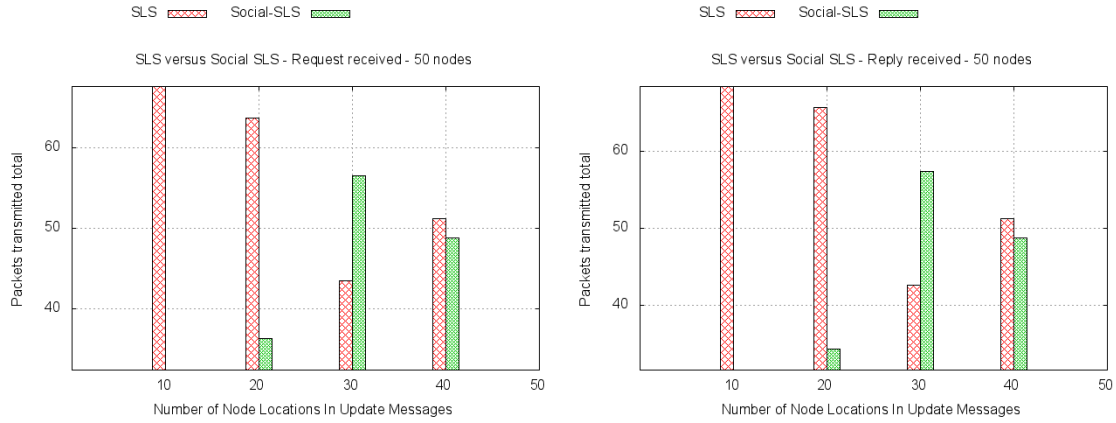


FIGURE 7.2 – SLS versus Social SLS - Request/Reply received - 50 nodes

7.5 Conclusion

Dans ce chapitre, nous présentons la notion de *groupe d'appartenance* ajoutée au *Simple Location Service*, *SLS* qui, comme son nom l'indique, est un *service de localisation* élémentaire et simple à amender.

Nous montrons que pour des petits nombres d'entrées dans *LP*, *SLS Social* offre de meilleures performances et pour des nombres d'entrées importants dans *LP*, *SLS Social* et *SLS* se valent. En conséquence, il est plus avantageux d'utiliser *SLS Social*.

Une évolution pourrait consister à, dans une optique probabiliste, prendre en compte le fait qu'il est clair que les nœuds d'un même groupe d'appartenance échangent davantage entre eux qu'avec l'extérieur de leur groupe. En conséquence, il serait envisageable de favoriser ces échanges en transmettant dans le groupe, les entrées du groupe présentent dans la table de localisation en priorité. Ainsi, nous gagnerions du point de vue de l'énergie pour peu de, voire aucune perte du point de vue de la communication.

7.5. CONCLUSION

Quatrième partie

Mesures

Chapitre 8

Introduction à la partie IV

8.1 Préambule

Cette partie est un peu atypique par rapport aux autres. En effet, les premières traitent des aspects humains individuels ou groupaux qui interviennent dans la prédiction des déplacements. La présente partie en revanche propose une approche novatrice de la mesure de la qualité du lien internodal dans les réseaux sans fil.

8.2 Introduction

Les protocoles de routage géographiques considèrent, dans la grande majorité des cas, l'information de positionnement « suffisamment » exacte et s'obstinent à ne pas prendre en compte l'incertitude consécutive aux :

- systèmes de positionnement géographique globaux comportant une tolérance pouvant s'étendre de quelques mètres à plusieurs centaines de mètres ;
- bruits électro-magnétiques environnementaux : *fading noise* dû à l'infrastructure (rebonds, diffraction), *thermal noise* dû déplacement des électrons.

Toutefois, Seada et al. [2004] tire la sonnette d'alarme et dénonce cet état de fait en mettant en évidence de larges déviations entre la vue idéalisée des modèles utilisés dans les outils de simulation communs, et la réalité du terrain des réseaux de capteurs. L'analyse et l'identification des causes mènent les auteurs à des propositions qui améliorent les performances globales et plus spécifiquement la consommation d'énergie, maillon faible des

réseaux de capteurs.

Citons également Song et al. [2004] qui fait le même constat et qui propose des améliorations par les modèles de prédiction de la mobilité. Après avoir démontré la supériorité des modèles markoviens d'ordre 2 muni d'une solution de repli de leur cru (*fallback*), les auteurs affichent des prédictions dont l'exactitude avoisine les 72 % sur un échantillon de traces d'environ 6000 utilisateurs Wi-Fi du campus de Dartmouth.

Enfin, Chen et al. [2010] dénonce la fâcheuse habitude de se borner à analyser les données statistiques des liens du réseau sans fil, globalement, sans se donner la peine d'approfondir les niveaux PHY et MAC, où les liens apparaissent dans toutes leurs faiblesses en termes d'incertitude. Après un constat chiffré, les auteurs nous livrent une approche dite « plan de reprise stochastique » (*stochastic recovery SR*) du problème du voyageur canadien (*Canadian Traveller Problem CTP*)¹. En outre, l'utilisation de l'*Optimal Stopping Theory* permet la prise en compte des liens à vitesse variable et réduit, dans ce cas précis, le *SRCTP* à un modèle de liens à 2 états. Cette prise en charge de l'incertitude aux niveaux PHY et MAC, a permis aux auteurs de publier des résultats comparés qui montrent des gains sur le délai de bout en bout de l'ordre de 51,15 à 73,02 % et de l'ordre de 94.44 à 99,76 % en ce qui concerne le rapport paquets envoyés sur paquets reçus.

Tous ces travaux convergent vers une vérité que tout le monde connaît mais a du mal à admettre : la prise en compte de l'incertitude dans le routage géographique est primordiale.

Notre idée est de proposer une métrique qui garantisse les termes d'un contrat de Qualité de Service dont les termes figent les contraintes suivantes :

- l'incertitude sur la distance entre 2 nœuds, caractérisée par son écart-type σ ;
- l'incertitude sur l'atténuation σ_η ;
- le bruit thermique N .

Dans ces conditions, il est clair que plus un nœud récepteur (dit par la suite relais) est proche de l'émetteur (P_r), moins il y aura de possibilités d'erreur (P_e). Réciproquement, plus la probabilité d'erreur est petite, plus le relais est proche de l'émetteur. En d'autres termes, la probabilité d'erreur et la distance émetteur-récepteur évoluent à l'inverse l'une de

1. version généralisée du problème du plus court chemin, à savoir que le graphe se construit en temps réel pendant son exploration en vue de son solutionnement

l'autre. Nous choisissons comme *critère de conception* une manière de pondérer le rapport entre ces 2 mesures. Puisqu'elles sont toutes deux définies positivement, une méthode qui réalise la décision optimale du choix du relais, consiste en la *fonction de coût* suivante :

$$J = P_e + \frac{\rho}{P_r} \quad (8.1)$$

où ρ est un scalaire positif.

L'idée consiste donc à construire cette fonction de coût à l'aide des éléments mathématiques de traitement du signal qui s'y rattache. Ces éléments ont pour base commune, la nature gaussienne de l'incertitude des mesures élémentaires qui entrent dans le calcul de cette fonction de coût. Toutefois, la démarche nous est apparue suffisamment généraliste pour nous permettre d'envisager de l'utiliser pour le calcul de cette nouvelle métrique de qualité de lien internodal dans les réseaux sans fil.

Toutefois, rien ne nous empêche de projeter dans l'avenir, de l'utiliser à d'autres fins. Plus précisément, une adaptation alternative à celle décrite dans le présent chapitre, nous apparaît fort possible. Celle-ci, si nous avons pu la réaliser, aurait permis de recentrer cette partie dans la droite ligne des 2 premières. Elle aurait consisté en l'intégration dans nos modèles de mobilité, de, par exemple, phénomènes sociaux gaussiens tels que Isaksson [2009]. Ce dernier montre qu'il est possible de prévoir le candidat qui aura la faveur de l'électorat par une approche gaussienne. Rien ne s'oppose a priori à utiliser ses résultats dans l'élaboration des choix de déplacement de groupes, dans nos modèles de mobilité.

Les protocoles de routage géographiques affichent d'ores et déjà des performances accrues et sont considérés comme les candidats les mieux placés pour les réseaux sans fil à grande échelle. Ces protocoles ont une charge de contrôle peu importante car allégée des mécanismes de gestion de routes. L'information de positionnement permet à tout nœud d'informer ses pairs de sa situation. L'instance décisionnelle d'acheminement est locale à chaque nœud en ce sens où ce dernier agit en qualité de routeur ou *relais* et choisit donc pour prochain saut (*next hop*) le voisin le plus prometteur en termes de performances d'acheminement du paquet d'information courant vers sa destination.

8.3 Approche choisie

Du point de vue purement systémique, l'acheminement d'une information est optimal dès lors que celle-ci est délivrée à bon port dans les délais les plus courts. Ce dernier critère peut être assimilé au chemin le plus court ou encore à la progression en *bits d'information par mètre* la plus grande vers la destination si l'on considère une évaluation continue durant l'acheminement. C'est ce dernier point de vue que nous adoptons.

Soit une source émettant r messages d'entropie² H , par seconde. La vitesse de transmission R en *bits/sec* est donnée par la formule évidente suivante : $R = rH$. Le *théorème de Shannon-Hartley* stipule que la seule et unique condition pour que les messages arrivent à bon port³ est $R < C$ où C est la capacité du réseau avec $C = B \log_2(1 + S/N)$, calculée à partir de son débit B en *bits/sec*, son signal reçu S en *Watt* et son bruit gaussien N en *Watt*. Le signal reçu S est calculé a priori par le nœud émetteur. S est une fonction non triviale de la distance d entre l'émetteur et le récepteur laquelle comprend une incertitude gaussienne. N comporte également une incertitude gaussienne. Si bien que la capacité C dépendant des variables aléatoires d et N , elle est variable aléatoire. Par conséquent, il est possible, pour un relais émetteur, d'exprimer la probabilité de livraison (ou resp. d'erreur de transmission) du message vers le prochain relais, en fonction de $R < C$ (ou resp. $R \geq C$). Il en va de même de la probabilité de progression, c'est-à-dire de d , la distance entre relais émetteur et récepteur, exprimable en termes de $\arg \max_d R < C$. Cela étant calculé pour chaque nœud, l'émetteur n'a plus qu'à choisir le prochain nœud i.e. le prochain saut (*next hop*), à l'aide de ces 2 probabilités. La qualité du lien entre nœuds émetteur et récepteur est ainsi mesurable à l'aune de ces 2 probabilités, ou en d'autres termes, de C .

8.4 Aperçu de la partie

- Le chapitre Mesures modélise la métrique ;
- Le chapitre Optimisation propose des simplifications qui permettent la réduction du temps de calcul de la métrique ;
- Le chapitre Évaluation des mesures fait état de nos campagnes de tests, des résultats

2. comprendre : codés sur H bits

3. moyennant détection et correction d'erreur éventuelles

obtenus et des interprétations proposées.

Chapitre 9

Mesures

9.1 Introduction

Ce chapitre propose une approche novatrice de la mesure de la qualité du lien internodal dans les réseaux sans fil. Nous modélisons cette mesure en Sec. 9.2, puis extrayons le critère de notre fonction de coût en Sec. 9.3.

9.2 Modélisation

Considérons un réseau sans fil muni d'un protocole de routage géographique. En supposant la position d'un nœud est connue précisément, le protocole peut sans trop de difficulté déterminer le prochain saut à effectuer pour acheminer un paquet d'information. En revanche, la donnée de positionnement prise en compte avec son incertitude, que nous supposons connue, exige du protocole de fournir un sous-ensemble de nœuds éligibles pour le prochain saut. Des éléments de la théorie du signal peuvent nous aider dans la définition de cette incertitude, d'une part, et, d'autre part, dans la détermination du sous-ensemble de nœuds éligibles. Prenons tout d'abord, l'équation suivante connue sous le nom de *théorème de Shannon-Hartley* :

$$C = B \log_2(1 + SNR) \tag{9.1}$$

Celle-ci nous permet de connaître la capacité C d'un médium air en *bits par seconde*, à partir de ses :

- débit (*Bandwidth*), en *Hertz*, B , et

- rapport signal sur bruit (*Signal on Noise Ratio*), sans dimension *Watt/Watt*, *SNR*, lorsque ce dernier comporte une incertitude du point de vue du bruit.

Afin de quantifier la dégradation du medium air en termes d'incertitudes, nous analysons en premier lieu, la capacité C en fonction de la distance entre relais émetteur et récepteur d . Tout d'abord, Su [2007] nous fournit la perte de puissance du signal en fonction de d :

$$Pl(d) = Pl(d_0) + 10\gamma \log_{10}(d/d_0) + \eta$$

où :

- $Pl(d)$ est la perte de puissance en dB après avoir parcouru d mètres,
- d_0 est la distance de référence,
- γ est l'exposant de pondération de la distance (entre 2 et 4),
- η est l'atténuation i.e. *slow fading noise (shadow noise)*.

η est initialement *log-normal*. Suite à sa conversion en dB par application de la fonction $\log$, cette variable devient normale :

$$\eta \sim \mathcal{N}(0, \sigma_\eta^2)$$

avec σ_η^2 entre 3 et 8 dB .

La puissance du signal reçu en dB est :

$$Pr(d) = Pt - Pl(d)$$

où Pt est la puissance du signal émis.

En prenant 1 m pour distance de référence, on obtient :

$$Pr(d) = \kappa - 10\gamma \log_{10}(d) - \eta$$

où :

$$\kappa = Pt - Pl(d_0)$$

est une constante.

À l'aide de ces définitions, le rapport signal sur bruit, en dB , s'écrit :

$$\begin{aligned} SNR^{dB} &= Pr(d) - N \\ &= \kappa - 10\gamma \log_{10}(d) - \eta - N \end{aligned}$$

où N est la puissance du bruit thermique en dB .

Le rapport signal sur bruit de l'Eq. 9.1 devient :

$$\begin{aligned} SNR &= 10^{SNR^{dB}/10} \\ &= 10^{(\kappa-N)/10} d^{-\gamma} 10^{(-\eta)/10} \\ &= \alpha d^{-\gamma} e^{-a\eta} \end{aligned} \tag{9.2}$$

où :

- $\alpha = 10^{(\kappa-N)/10}$, est une constante,
- $a = \ln(10)/10$, est une constante (à noter que $e^{ab} = (e^a)^b$),
- $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ où $2 \leq \sigma_\eta^2 \leq 4$.

En supposant que les coordonnées (x, y) ont une incertitude normale du type *independent and identical distribution iid* :

- $x_t \sim \mathcal{N}(\mu_{xt}, \sigma_t^2)$,
- $y_t \sim \mathcal{N}(\mu_{yt}, \sigma_t^2)$ pour l'émetteur et
- $x_r \sim \mathcal{N}(\mu_{xr}, \sigma_r^2)$,
- $y_r \sim \mathcal{N}(\mu_{yr}, \sigma_r^2)$ pour le récepteur,

la distance d est *Rice distributed* au sens de Papoulis and Pillai [2002] et Gudbjartsson and Patz [1995] :

$$d \sim \mathcal{R}(\mu, \sigma^2)$$

avec Peng et al. [2011] :

- $\mu = \sqrt{(\mu_{xt} - \mu_{xr})^2 + (\mu_{yt} - \mu_{yr})^2}$ et
- variance $\sigma^2 = \sigma_r^2 + \sigma_t^2$.

Une propriété importante de Gudbjartsson and Patz [1995] :

$$\begin{cases} d \sim \mathcal{R}(\mu, \sigma^2) \\ \mu \geq 3 \times \sigma \end{cases} \Rightarrow d \sim \mathcal{N}(\sqrt{\mu^2 + \sigma^2}, \sigma^2) \tag{9.3}$$

En posant $B = \ln(2)$, l'Eq. 9.1 devient :

$$\begin{aligned} C &= \ln(2) \times \frac{\ln(1 + SNR)}{\ln(2)} \\ &= \ln(1 + SNR) \end{aligned} \tag{9.4}$$

Dans la progression d'un message vers sa destination, à savoir :

$$(\mu_{xt} - \mu_{xr}) \gg (\mu_{yt} - \mu_{yr})$$

nous définissons une distance positive, dans le cas où il y a réelle progression vers la destination, et une distance négative dans le cas contraire.

En utilisant l'Eq. 9.2 dans l'Eq. 9.4, la capacité C peut s'exprimer ainsi :

$$C = \ln(1 + \alpha|d|^{-\gamma}e^{-a\eta}) \quad (9.5)$$

C représente la limite, en termes de *bits d'information par seconde*, qui ne peut être dépassée sans perte d'information. De plus, cela nous donne une indication de la qualité de la communication en fonction de :

- σ_η : l'incertitude sur l'atténuation incluse dans η ;
- σ : l'incertitude sur la distance incluse dans d et
- N : le bruit thermique constant inclus dans α .

Comme mentionné en section ??, nous cherchons à évaluer cette limite en tant que mesure représentative de la qualité du lien émetteur-récepteur. Ce, afin d'améliorer le choix du relais, dans l'optique d'une réduction des délais de transmission.

À cette fin, nous définissons les 2 mesures suivantes :

1. *error probability* : la probabilité d'excéder cette limite et de perdre de l'information ;
2. *progress information* : la probabilité sur la distance parcourue à chaque saut.

D'abord, notons que :

$$\begin{aligned} e^C &= 1 + \alpha|d|^{-\gamma}e^{-a\eta} \\ e^C - 1 &= \alpha|d|^{-\gamma}e^{-a\eta} \\ \frac{e^C - 1}{\alpha} &= |d|^{-\gamma}e^{-a\eta} \end{aligned}$$

Ensuite, définissons :

$$\begin{aligned} z &= \ln(|d|^{-\gamma}e^{-a\eta}) \\ z &= \ln(e^{-a\eta}|d|^{-\gamma}) \\ z &= -a\eta - \gamma \ln(|d|) \end{aligned}$$

Il est clair que toute condition sur C peut s'écrire en termes de z .

Soit $\underline{C}$ la capacité minimum fixée à la conception du réseau¹ sous laquelle toute communication est impossible. Ensuite, C est une capacité de lien admissible si et seulement si :

$$\begin{aligned} \underline{C} &< C \\ \iff \\ \bar{z} &< z \\ \bar{z} &< -a\eta - \gamma \ln(|d|) \end{aligned} \tag{9.6}$$

$$\begin{aligned} \eta &< -\frac{\gamma \ln(|d|) + \bar{z}}{a} \\ |d| &> e^{-\frac{a\eta + \bar{z}}{\gamma}} \end{aligned} \tag{9.7}$$

Soit $\mathcal{D}$ l'ensemble des couples (η, d) , solutions de l'Eq. 9.6. $\mathcal{D}$ est représentée par la zone blanche de la Figure 1.

Quant à la zone grise de la Figure 1, elle représente les conditions sur η et d sous lesquelles l'information n'atteindra pas le prochain relais.

Comme mentionné plus haut, l'incertitude sur d est normale. Elle ressemble donc à une cloche représentée par les ellipses concentriques et divisée par la courbe d'équation générale $y = -\ln(|x|)$, Figure 1, en 2 surfaces qui représentent exactement nos 2 mesures P_e and P_r avec bien sûr :

$$P_e + P_r = 1$$

Notre première mesure : *error probability* P_e peut être réécrite sous cette forme :

$$P_e = \mathcal{P}\{z \leq \bar{z}\} = 1 - \int \int_{\mathcal{D}} f_{\eta d}(\eta, d) d\eta dd$$

1. qui dépend de nombreux paramètres de conception tels que : la méthode d'encodage, la puissance d'émission, le bruit moyen attendu, etc.

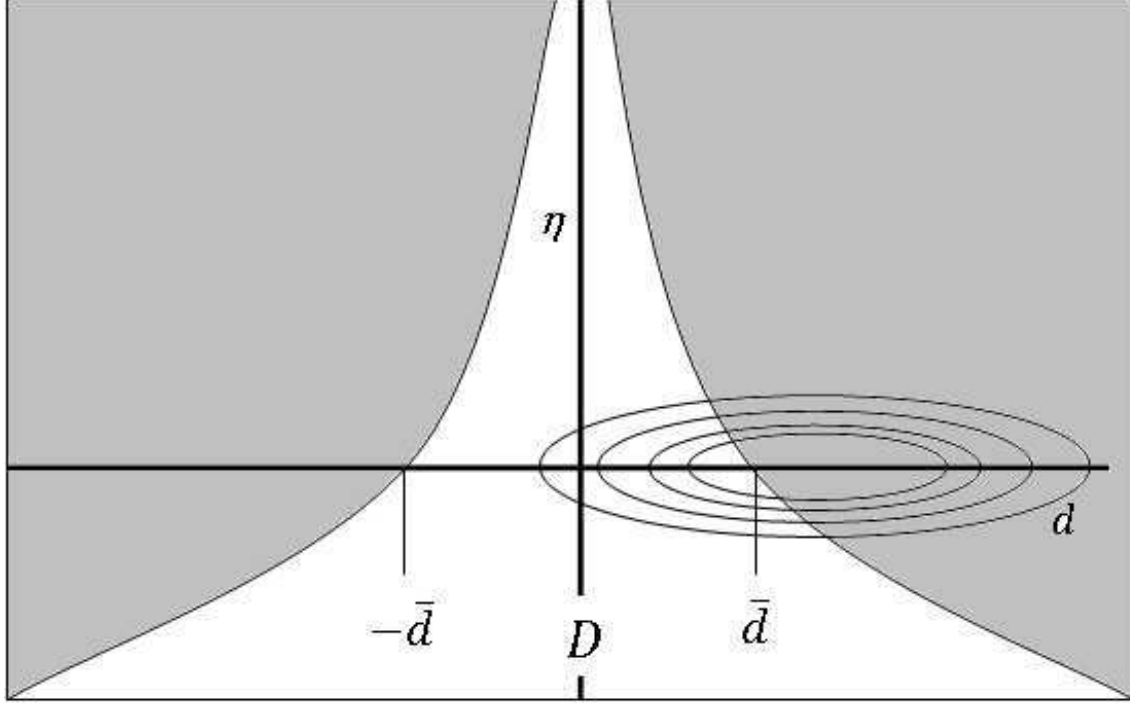


FIGURE 9.1 – Jointure entre η, d et ensemble $\mathcal{D} = \{(\eta, d) | z > \bar{z}\}$ qui définit la probabilité d'arrivée effective de l'information au prochain relais

En considérant l'indépendance entre η et d , et en appelant $\mathcal{D}(\eta)$ le sous-ensemble de $\mathcal{D}$ pour un η fixé, la forme se transforme en :

$$\begin{aligned} P_e &= 1 - \int_{-\infty}^{\infty} f_{\eta}(\eta) \left[\int_{\mathcal{D}(\eta)} f_d(d) dd \right] d\eta \\ &= 1 - \mathcal{E}_{\eta} \left[\int_{\mathcal{D}(\eta)} f_d(d) dd \right] \end{aligned} \quad (9.8)$$

où $\mathcal{E}_{\eta}$ signifie valeur moyenne attendue sachant η .

La seconde mesure considérée est la progression P_r qui est définie en termes de distance moyenne parcourue par le message, si tant est qu'il soit passé. Cela peut s'écrire également

ainsi :

$$\begin{aligned} P_r &= \frac{1}{1 - P_e} \int_{-\infty}^{\infty} f_{\eta}(\eta) \left[\int_{\mathcal{D}(\eta)} d \times f_d(d) dd \right] d\eta \\ &= \frac{1}{1 - P_e} \mathcal{E}_{\eta} \left[\int_{\mathcal{D}(\eta)} d \times f_d(d) dd \right] \end{aligned} \quad (9.9)$$

Notons le facteur $\frac{1}{1 - P_e}$ qui met à l'échelle dans l'intervall $[0, 1]$. Notons également la présence du facteur d (i.e. $d \times$) dans l'intégrale intérieure qui souligne que l'on prend la valeur moyenne de d .

Nous allons maintenant montrer comment s'utilisent ces mesures dans l'algorithme de choix du prochain saut, parmi l'ensemble des relais candidats. Pour ce faire, il nous faut définir un critère de décision.

9.3 Fonction de coût

Fixons tout d'abord :

- l'incertitude sur la distance entre 2 relais, caractérisée par son écart-type σ ;
- l'incertitude sur l'atténuation σ_{η} ;
- le bruit thermique N .

Dans ces conditions, il est clair que plus un relais est proche de l'émetteur, moins il y aura de possibilités d'erreur. Réciproquement, plus la probabilité d'erreur est petite, plus le relais est proche de l'émetteur. En d'autres termes, la probabilité d'erreur et la distance émetteur-récepteur évoluent à l'inverse l'une de l'autre. Nous choisissons une *fonction de coût* permettant de pondérer le rapport entre ces 2 mesures afin d'effectuer le meilleur choix du prochain relais :

$$J = P_e + \frac{\rho}{P_r} \quad (9.10)$$

où ρ est un scalaire positif.

Considérons dans un premier temps que la condition suivante est respectée : $\mu/\sigma > 3$. Dans ce cas, la *Rice distribution* peut s'écrire sous la forme d'une loi normale, $\mathcal{R}(\mu, \sigma^2) \approx \mathcal{N}(\sqrt{\mu^2 + \sigma^2}, \sigma^2)$. Par la suite, nous pourrions éventuellement nous intéresser au cas où cette condition n'est pas respectée. En prenant en compte l'indépendance des variables

aléatoires d , la distance et σ_η , l'incertitude de l'atténuation (*fading noise*), les mesures de l'Eq. 9.8 et l'Eq. 9.9, s'expriment ainsi :

$$P_e = 1 - \frac{1}{2\pi\sigma\sigma_\eta} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{\eta}{\sigma_\eta}\right)^2} \left[\int_{-e^{-\frac{a\eta+\bar{z}}{\gamma}}}^{e^{-\frac{a\eta+\bar{z}}{\gamma}}} e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2} dd \right] d\eta \quad (9.11)$$

$$P_r = \frac{1}{1-P_e} \times \frac{1}{2\pi\sigma\sigma_\eta} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{\eta}{\sigma_\eta}\right)^2} \left[\int_{-e^{-\frac{a\eta+\bar{z}}{\gamma}}}^{e^{-\frac{a\eta+\bar{z}}{\gamma}}} d \times e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2} dd \right] d\eta \quad (9.12)$$

notons l'intervalle de l'intégrale intérieure $\left[-e^{-\frac{a\eta+\bar{z}}{\gamma}}, e^{-\frac{a\eta+\bar{z}}{\gamma}} \right]$ qui est en conformité avec l'Eq. 9.7. Afin d'obtenir des équations tractables par les systèmes informatiques, nous devons de simplifier et d'obtenir une expression facilement calculable de nos 2 mesures. C'est l'objet du chapitre 10. Nous obtenons les expressions simplifiées suivantes :

$$P_e \approx \frac{1}{\sqrt{8\chi}\beta\sigma_\eta} \left(1 - \operatorname{erf} \left(\sqrt{\beta} \frac{\bar{d} - \bar{\mu}}{\sqrt{2\sigma}} \right) \right). \quad (9.13)$$

$$P_r \approx -e^{\frac{-\beta(\bar{d}-\bar{\mu})^2}{2\sigma^2}} \frac{\sigma}{2\sqrt{\pi\chi}\sigma_\eta(1-P_e)} + \bar{\mu}. \quad (9.14)$$

où :

$$\chi = \frac{1}{2\sigma_\eta^2} + \frac{a^2\bar{d}^2}{2\gamma^2\sigma^2} \quad (9.15)$$

$$\beta = 1 - \frac{a^2\bar{d}^2}{2\gamma^2\sigma^2\chi} \quad (9.16)$$

$$\bar{\mu} = \mu + \sigma \quad (9.17)$$

et la valeur tabulée $\operatorname{erf}(x) = (2/\sqrt{\pi}) \int_0^x e^{-t^2} dt$. Soulignons que les paramètres statistiques nécessaires aux calculs de ces expressions sont pour chaque relais : γ, σ_η, μ et σ , où γ et σ_η sont des paramètres du médium air et μ et σ sont obtenus à partir des statistiques géographiques.

9.4 Conclusion

Nous proposons dans ce chapitre une métrique de sélection du prochain saut ou relais pour les routages géographiques qui minimise une fonction de coût qui mesure la qualité du lien entre l'émetteur et le récepteur :

$$J = P_e + \frac{\rho}{P_r} \quad (9.18)$$

où ρ est un scalaire positif. Cette fonction de coût est minimale quand *error probability*, P_e est minimale et *progress of information*, P_r est maximale. En d'autres termes, $\arg \min_{(P_e, P_r)} J$, assure pour une valeur de ρ donnée, la progression la plus rapide du paquet transmis (en termes de distance parcourue) pour le minimum d'erreur. Ces 2 mesures sont calculées à partir de la capacité du médium air décrite par une variable aléatoire. Sont prises en compte les incertitudes sur l'atténuation (*fading noise*) et la position du relais ou nœud destination par rapport au nœud émetteur. Cela est associé au bruit thermique pour définir un contrat de qualité de service :

- l'incertitude sur la distance entre 2 nœuds, caractérisée par son écart-type σ ;
- l'incertitude sur l'atténuation σ_η ;
- le bruit thermique N .

Le choix d'une incertitude gaussienne plutôt que ricienne nous permet d'obtenir en peu d'étapes un modèle puissant et robuste. En revanche, nous sommes contraints pour ce faire à réduire le champ applicatif à $\mu/\sigma > 3$. Par ailleurs, les calculs sont lourds et demandent à être optimisés en vue d'une implémentation.

Dans le chapitre qui suit, nous présentons des éléments de réponse qui permettent de passer outre à la condition limitative, $\mu/\sigma > 3$, d'une part, et, d'autre part, d'optimiser les calculs en vue de l'implémentation.

9.4. CONCLUSION

Chapitre 10

Optimisation

10.1 Introduction

La résolution de l'Eq.9.11 et de l'Eq.9.12 est coûteuse en temps de calcul. Aussi, nous proposons une approche plus adaptée aux contraintes temps réel d'un protocole de routage. Nous verrons tout d'abord la résolution de l'Eq.9.11 où l'approche proposée consiste à linéariser les limites de l'intégration. Notons que la distribution conjointe est dense autour des coordonnées $(\mu, 0)$ comme on le voit sur la figure 1, dans un petit intervalle autour de η , donné par l'écart-type $\sigma_\eta \in [1, 3]dB$. Puis, une approche efficace consiste à linéariser les limites d'intégration autour de $\eta = 0$. L'erreur en utilisant cette linéarisation, augmente à mesure que nous nous éloignons des coordonnées $(\mu, 0)$, mais dans le même temps, la fonction de densité diminue rapidement, de sorte qu'en utilisant l'approximation, on peut s'attendre à une très petite erreur dans l'intégrale. Enfin, nous traiterons du cas où la condition $\mu/\sigma > 3$ n'est pas respectée.

10.2 Probabilité d'erreur

L'approximation linéaire est obtenue en prenant le terme linéaire du développement en série de Taylor, au point de coordonnées $\eta = 0$ soit :

$$d = e^{(-a\eta - \bar{z})/\gamma} \approx \bar{d}(1 - \frac{a}{\gamma}\eta) \quad (10.1)$$

Puis, en prenant en compte le fait que le côté gauche de la partie grise de Figure 1 ne contribue pas significativement à la probabilité d'erreur, l'Eq.9.11 est approximativement

égale à :

$$Pe \approx \frac{1}{2\pi\sigma\sigma_\eta} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{\eta}{\sigma_\eta}\right)^2} \int_{\bar{d}(1-\frac{a}{\gamma}\eta)}^{\infty} e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2} dd d\eta \quad (10.2)$$

$$Pe \approx 1 - \frac{1}{\sigma\sigma_\eta\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{\eta}{\sigma_\eta}\right)^2} \left[\int_0^{\bar{d}(1-\frac{a}{\gamma}\eta)} e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2} dd \right] d\eta \quad (10.3)$$

en appliquant le changement de variable suivant : $\tau = d - \bar{d}(1 - \frac{a}{\gamma}\eta)$, la probabilité d'erreur s'écrit :

$$Pe \approx \frac{1}{2\pi\sigma\sigma_\eta} \int_0^{\infty} \int_{-\infty}^{\infty} e^{-\chi\left(\eta - \frac{a_1}{2\chi}\right)^2 - a_2\beta} d\eta d\tau \quad (10.4)$$

où χ et β sont définis dans (9.15) (9.16), et

$$a_1 = \frac{\tau - \bar{\mu} + \bar{d}}{\gamma\sigma^2} a\bar{d}; \quad a_2 = \frac{(\tau - \bar{\mu} + \bar{d})^2}{2\sigma^2}$$

Puis, en prenant en compte le fait que :

$$\int_{-\infty}^{\infty} e^{-\chi\left(\eta - \frac{a_1}{2\chi}\right)^2} d\eta = \sqrt{\frac{\pi}{\chi}}, \quad (10.5)$$

la probabilité d'erreur se réduit à l'expression suivante :

$$Pe \approx \sqrt{\frac{\pi}{\chi}} \frac{1}{2\pi\sigma\sigma_\eta} \int_0^{\infty} e^{-a_2\beta} d\tau \quad (10.6)$$

par un nouveau changement de variable, à savoir : $t = \sqrt{a_2\beta}$ nous obtenons :

$$Pe \approx \frac{1}{\sqrt{2\chi\beta\pi}\sigma_\eta} \int_{\sqrt{\beta}\frac{\bar{d}-\bar{\mu}}{2\sigma^2}}^{\infty} e^{-t^2} dt \quad (10.7)$$

qui est équivalent à (9.13).

10.3 Probabilité de progression

En ce qui concerne la probabilité de progression, résolvons tout d'abord l'intégrale interne de l'Eq.9.12, en usant des mêmes arguments que pour Pe , concernant les limites de l'intégration. L'intégrale interne peut s'écrire :

$$\int_{-\bar{d}(1-\frac{a}{\gamma}\eta)}^{\bar{d}(1-\frac{a}{\gamma}\eta)} d e^{-\frac{1}{2}\left(\frac{d-\mu}{\sigma}\right)^2} dd \quad (10.8)$$

En dérivant la fonction exponentielle, nous obtenons :

$$\frac{d}{dd} \left(e^{-\frac{1}{2} \frac{(d-\bar{\mu})^2}{\sigma^2}} \right) = -\frac{(d-\bar{\mu})}{\sigma^2} e^{-\frac{1}{2} \frac{(d-\bar{\mu})^2}{\sigma^2}} \quad (10.9)$$

En arrangeant, nous trouvons :

$$de^{-\frac{1}{2} \frac{(d-\bar{\mu})^2}{\sigma^2}} = -\sigma^2 \frac{d}{dd} \left(e^{-\frac{1}{2} \frac{(d-\bar{\mu})^2}{\sigma^2}} \right) + \bar{\mu} e^{-\frac{1}{2} \frac{(d-\bar{\mu})^2}{\sigma^2}} \quad (10.10)$$

En multipliant par $1/(\sqrt{2\pi}\sigma)$ et en intégrant entre les extrêmes, nous trouvons que l'intégrale dans (10.8) est égale à :

$$\begin{aligned} & -\frac{\sigma}{\sqrt{2\pi}} \left[e^{-\frac{1}{2} \frac{(\bar{d}(1-\frac{a}{\gamma}\eta)-\bar{\mu})^2}{\sigma^2}} - e^{-\frac{1}{2} \frac{(\bar{d}(1-\frac{a}{\gamma}\eta)+\bar{\mu})^2}{\sigma^2}} \right] + \frac{\bar{\mu}}{2} \\ & \left[erf \left(\frac{\bar{d}(1-\frac{a}{\gamma}\eta) + \bar{\mu}}{\sqrt{2}\sigma} \right) + erf \left(\frac{\bar{d}(1-\frac{a}{\gamma}\eta) - \bar{\mu}}{\sqrt{2}\sigma} \right) \right] \end{aligned} \quad (10.11)$$

En prenant en compte l'écart-type de η , la seconde exponentielle ci-dessus est proche de zéro, tandis que la première fonction d'erreur est proche de l'unité, si bien que l'Eq.9.12 peut être approximée par :

$$\begin{aligned} Pr \approx & \frac{1}{\sqrt{2\pi}\sigma_\eta} \int_{-\infty}^{\infty} e^{-\frac{\eta^2}{2\sigma_\eta^2}} \left[-\frac{\sigma}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{(\bar{d}(1-\frac{a}{\gamma}\eta)-\bar{\mu})^2}{\sigma^2}} + \right. \\ & \left. \frac{\bar{\mu}}{2} \left(1 + erf \left(\frac{\bar{d}(1-\frac{a}{\gamma}\eta) - \bar{\mu}}{\sqrt{2}\sigma} \right) \right) \right] d\eta / (1 - Pe) \end{aligned} \quad (10.12)$$

Maintenant, l'intégral du premier terme est égale à l'intégrale interne de (10.4) mais en renommant a_1 et a_2 pour $\tau = 0$. Ainsi, l'intégrale interne est égale à :

$$\begin{aligned} & \frac{-\sigma}{2\pi\sigma_\eta} \int_{-\infty}^{\infty} e^{-\frac{\eta^2}{2\sigma_\eta^2}} e^{-\frac{1}{2} \frac{(\bar{d}(1-\frac{a}{\gamma}\eta)-\bar{\mu})^2}{\sigma^2}} d\eta = \\ & = \frac{-\sigma}{2\sigma_\eta\sqrt{\pi}\chi} e^{-\frac{1}{2} \frac{(\bar{d}-\bar{\mu})^2}{\sigma^2}} \beta \end{aligned} \quad (10.13)$$

Afin de résoudre le reste de l'intégrande dans (10.12) nous utilisons le fait que $erf(x) = 1 - (2/\sqrt{\pi}) \int_x^\infty e^{-t^2} dt$. Puis, en effectuant l'ultime changement de variable suivant : $\bar{t} = t\sqrt{2}\sigma + \bar{\mu}$, il vient :

$$\begin{aligned} & 1 + erf \left(\frac{\bar{d}(1-\frac{a}{\gamma}\eta) - \bar{\mu}}{\sqrt{2}\sigma} \right) = \\ & = 2 - \frac{2}{\sqrt{2\pi}\sigma} \int_{\bar{d}(1-\frac{a}{\gamma}\eta)}^{\infty} e^{-(\frac{\bar{t}-\bar{\mu}}{\sqrt{2}\sigma})^2} d\bar{t} \end{aligned} \quad (10.14)$$

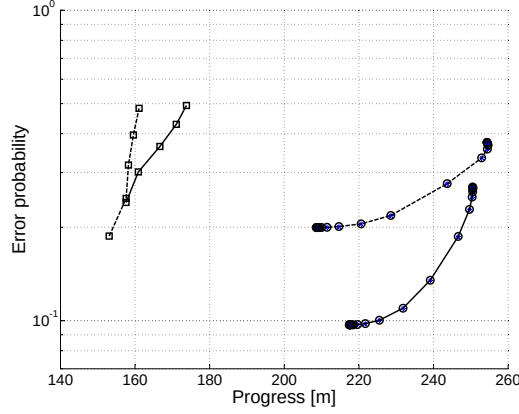


FIGURE 10.1 – Error probability versus progress for 5 relays. Dot-line for $\sigma_\eta = 3$, full-line for $\sigma_\eta = 1$. Squared for Deterministic decision and circle for stochastic one-hop decision. $\sigma = 210\text{ m} \Rightarrow \mu/\sigma < 3$

La seconde intégrale donne :

$$\frac{\bar{\mu}}{\sqrt{2\pi}\sigma_\eta} \int_{-\infty}^{\infty} e^{-\frac{\eta^2}{2\sigma_\eta^2}} \left(1 - \frac{1}{\sqrt{2\pi}\sigma} \int_{\bar{d}(1-\frac{a}{\gamma}\eta)}^{\infty} e^{-\left(\frac{\bar{t}-\bar{\mu}}{\sqrt{2}\sigma}\right)^2} d\bar{t}\right) d\eta \quad (10.15)$$

L'intégrande de la partie gauche est égale à $\bar{\mu}$ et la seconde peut être résolue en la rapprochant de (10.2) qui donne $\bar{\mu}Pe$. Puis, en ajoutant le terme (10.2) et divisant par $1 - Pe$ précédent, l'expression de l'Eq.9.12 est obtenue.

10.4 Cas où $\mu/\sigma \leq 3$

Il suffit de remplacer dans les Eq. 9.11 et 9.12, la normale par une ricienne. La simulation sous *Mathlab* démontre que les tendances obtenues, Fig. 10.1, sont les mêmes que celles obtenues avec une normale, Fig 10.2.

10.5 Conclusion

Dans ce chapitre, nous montrons que les informations posées au chapitre 11, sont suffisantes pour l'optimisation. Des expressions facilement calculables sont déduites. Les incertitudes sont *gaussiennes* car nous posons $\mu/\sigma > 3$. Cependant, une extension assez simple

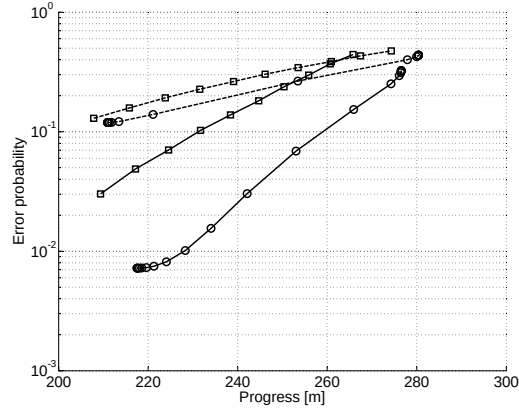


FIGURE 10.2 – Error probability versus progress for 10 relays. Dot-line for $\sigma_\eta = 3$, full-line for $\sigma_\eta = 1$. Square for deterministic decision and circle for stochastic one-hop-decision.

nous permet de démontrer que des comportements équivalents sont obtenus à partir d'une *Rice distribution* de la distance, pour le cas où $\mu/\sigma \leq 3$. Nous présentons donc des éléments de réponse qui permettent de passer outre à la condition limitative, d'une part, et, d'autre part, d'optimiser les calculs en vue de leur implémentation dans un langage de programmation. En l'occurrence, il s'agit de *C++* puisque nous avons choisi d'utiliser le cadre de travail *Network Simulator 2*, *NS2*, pour l'évaluation de notre modèle. C'est l'objet du chapitre qui suit.

10.5. CONCLUSION

Chapitre 11

Evaluation des mesures

11.1 Introduction

À titre d'exemple, nous appliquons les résultats obtenus aux chapitres 9 et 10, au protocole de routage géographique Dream, Camp [2003]. Ce dernier est décrit en détail dans la Sec. 5.4. Rappelons que ce protocole renferme un mécanisme de choix du relais suivant (*next hop*) qui utilise un cône d'émission $[-\alpha, +\alpha]$ autour de l'axe émetteur-destination θ . α est calculé en fonction de la vitesse de la destination, cf. Fig. 11.1.

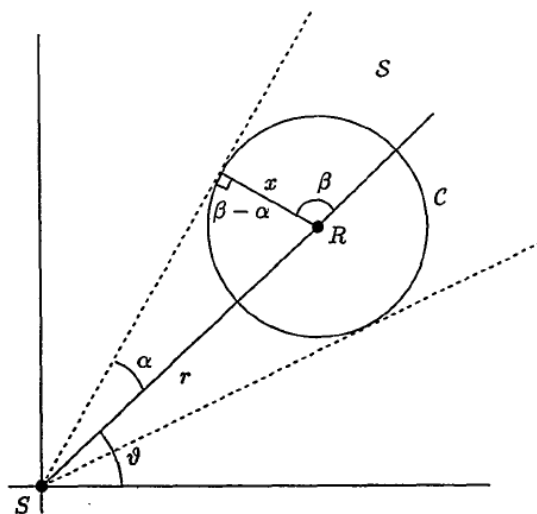


FIGURE 11.1 – DREAM : cône d'émission

Nous utilisons notre métrique afin de réduire encore le nombre d'émissions au relais du cône qui optimise notre fonction de coût. Soulignons qu'il ne s'agit que d'un début en ce

sens où des 3 paramètres de notre mesure :

- l’incertitude sur la distance entre 2 nœuds, caractérisée par son écart-type σ ;
- l’incertitude sur l’atténuation σ_η ;
- le bruit thermique N .

nous ne faisons varier que le premier : σ . Cela nous permet néanmoins de nous placer dans l’optique du *concepteur du réseau* dont l’objectif est de trouver le meilleur rapport Qualité/Prix parmi les différentes solutions qui répondent au problème qui lui est posé dans son cahier des charges, notamment du point de vue Qualité de Service dont la précision des mesures fait partie intégrante. Il est clair que la précision a un coût. Le fait d’offrir un aperçu des avantages et inconvénients d’une incertitude par rapport à une autre, est un atout majeur dans le choix de la solution finale qui sera alors retenue en connaissance de cause.

11.2 Configurations de test

Rappelons que le contrat de qualité de service se décline en :

- l’incertitude sur la distance entre 2 nœuds, caractérisée par son écart-type σ ;
- l’incertitude sur l’atténuation σ_η ;
- le bruit thermique N .

Dans nos tests, cependant, des 3 paramètres du contrat de qualité de service ci-dessus, nous ne faisons varier que le premier : σ . En effet, nous nous plaçons dans l’optique du *concepteur du réseau*, cf. 11.1. Les différentes valeurs de nos paramètres de simulation sont résumées en Table 11.1.

11.3 Résultats

Les Fig. 11.2 et 11.12 relatent les résultats de notre campagne de tests.

Les Fig. 11.2 à 11.7 présentent les fluctuations des différentes mesures en nombre de paquets de contrôle ou de données, émis ou reçus en fonction de l’incertitude sur la distance σ entre les nœuds émetteur et récepteur qu’il s’agisse des nœuds source et destination proprement dits ou des nœuds-relais situés entre ces derniers. La Fig. 11.2 montre que

TABLE 11.1 – Simulation

Simulateur	Network Simulator 2 (NS2)
Couche MAC	IEEE 802.11
Taux de transmission	100 m
Type de trafic	peer-to-peer 64 bytes packets
Destination	aléatoire
Caractéristiques physiques	914MHz Lucent WaveLAN DSSS
Ad hoc protocole	AODV
Vitesse des nœuds (m/s)	2 (piéton), 3 (bicyclette), 15 (véhicule)
Nombre de nœuds	50, 100
Aire de simulation	600 m $\times$ 300 m
Incertitude sur la distance σ	1 à 30

certaines moyens de transport sont plus sensibles à l'incertitude définie par le concepteur du réseau, que d'autres. En effet, sur cette figure, côté 50 nœuds, pour le moyen de transport « bicycle », mis à part l'incertitude de 19 m, le nombre de paquets de contrôle émis varie de moins d'1 %. Tandis que pour « pedestrian » et « car », les variations dépassent les 2 à 3 %. En passant de 50 à 100 nœuds, « pedestrian » présente une variation de l'ordre d'0,1 %. Tandis que pour « bicycle » et « car », les variations dépassent les 0,2 à 0,3 %.

Un résultat intéressant est donc le fait que :

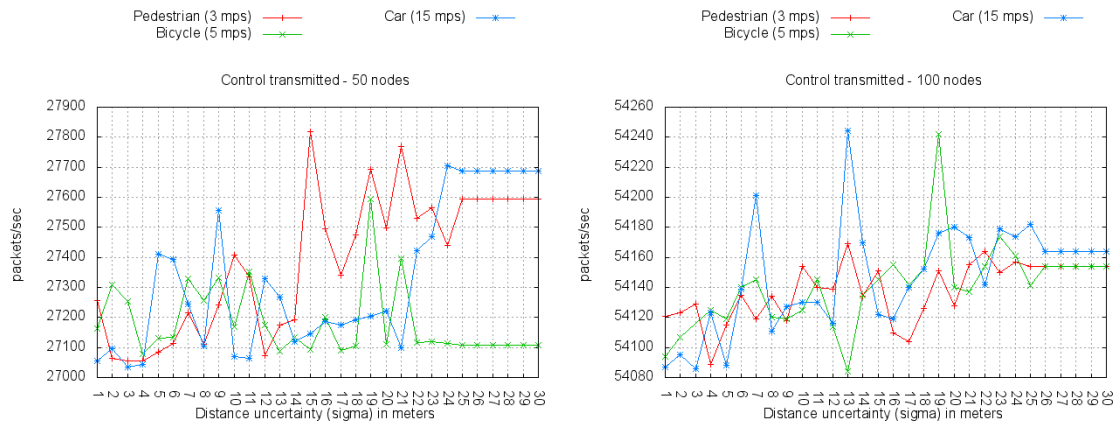


FIGURE 11.2 – Control transmitted - 50/100 nodes

la croissance de la densité limite l'impact de l'incertitude.

Par ailleurs, l'observation attentive de la Fig. 11.3 montre que pour une densité de

11.3. RÉSULTATS

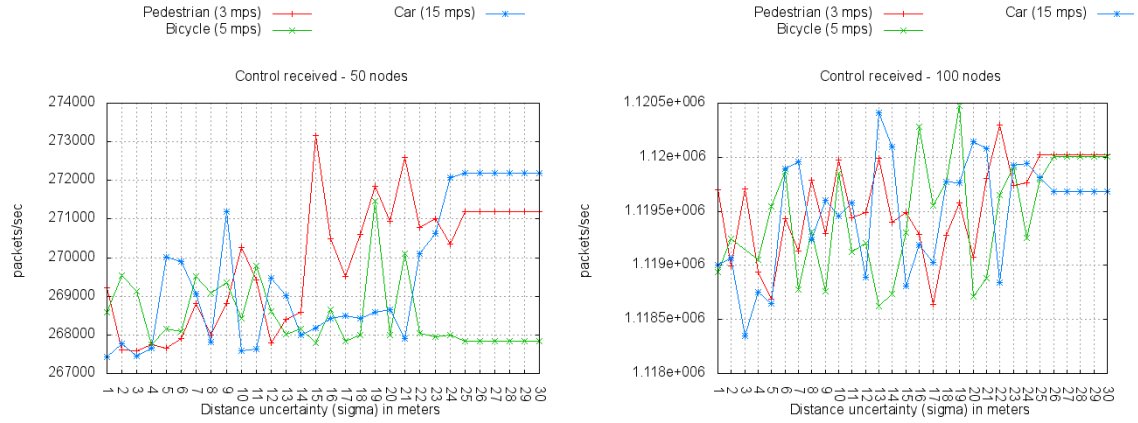


FIGURE 11.3 – Control received - 50/100 nodes

50 nœuds, le nombre de paquets de contrôle reçu est quasiment identique à celui émis. En revanche, pour une densité de 100 nœuds, le nombre de paquets de contrôle reçu est pratiquement 5 fois supérieur à celui émis. Un autre résultat intéressant est donc le fait que :

la croissance de la densité fait exploser le nombre de paquets de contrôle reçu et ce quelque soit l'incertitude.

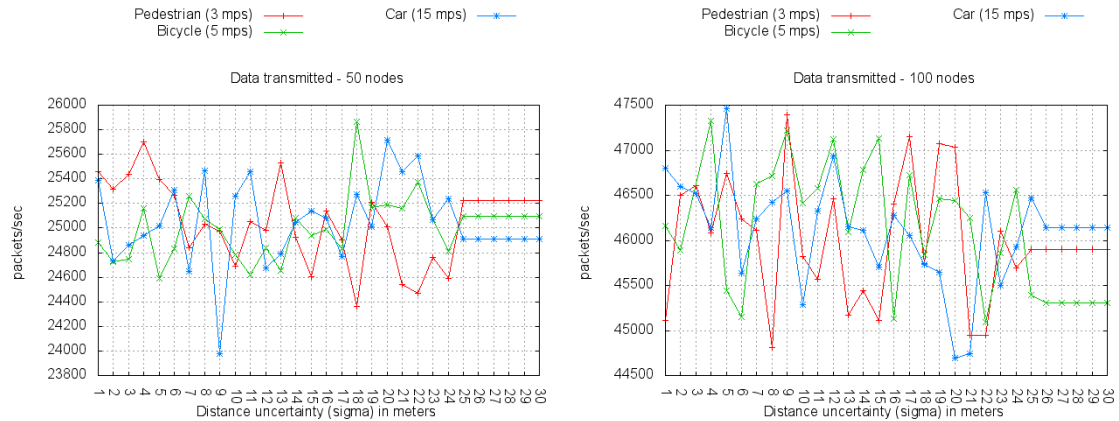


FIGURE 11.4 – Data transmitted - 50/100 nodes

La Fig. 11.5 confirme ces premiers résultats quoique les moyens de transport ne suivent pas exactement la même distribution selon l'incertitude. La Fig. 11.7 cumulant les paquets de contrôle et ceux de données, l'incertitude y joue un rôle qui est la moyenne des 2 premiers.

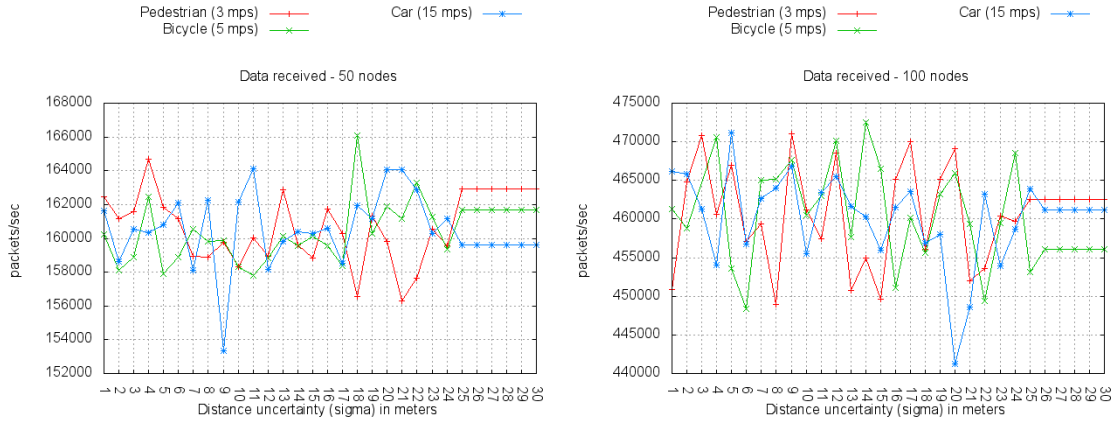


FIGURE 11.5 – Data received - 50/100 nodes

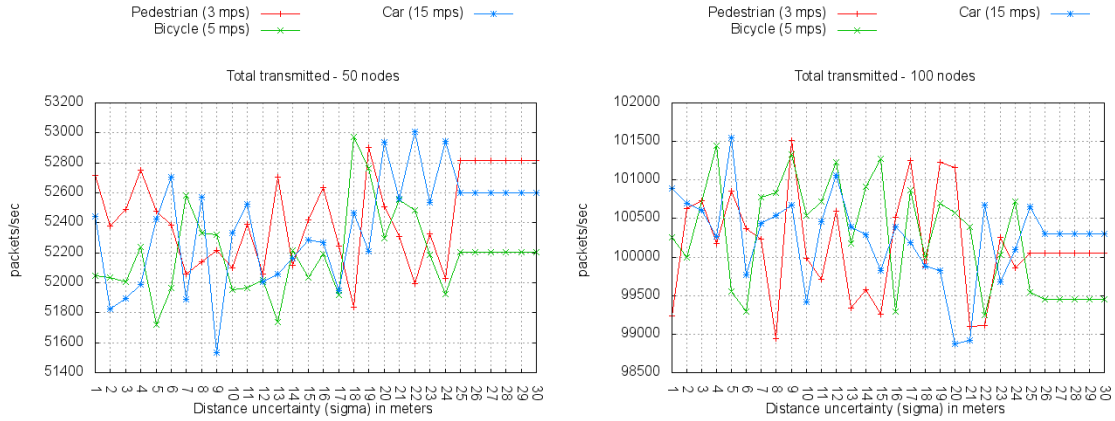


FIGURE 11.6 – Total transmitted - 50/100 nodes

La Fig. 11.9 donne le nombre de paquets de données mais cette fois-ci entre les nœuds source et destination proprement dits. La Fig. 11.10 donne le ratio du nombre de paquets de données reçus sur celui émis. L'incertitude y joue un rôle aussi important quelque soit la densité.

La Fig. 11.11 donnent le nombre de sauts moyen requis pour transmettre un paquet. La Fig. 11.12 donnent le nombre de sauts moyen requis pour transmettre un paquet. L'incertitude y joue également un rôle aussi important quelque soit la densité.

11.4. CONCLUSION

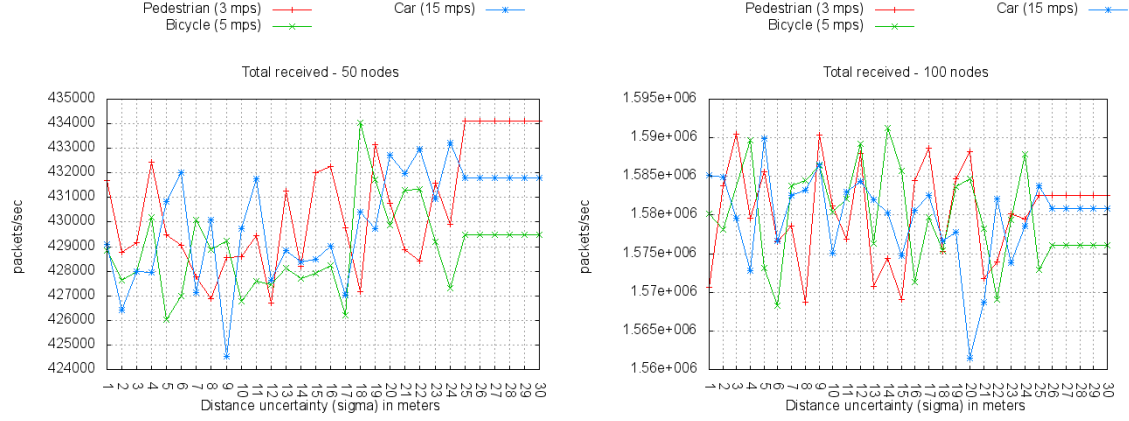


FIGURE 11.7 – Total received - 50/100 nodes

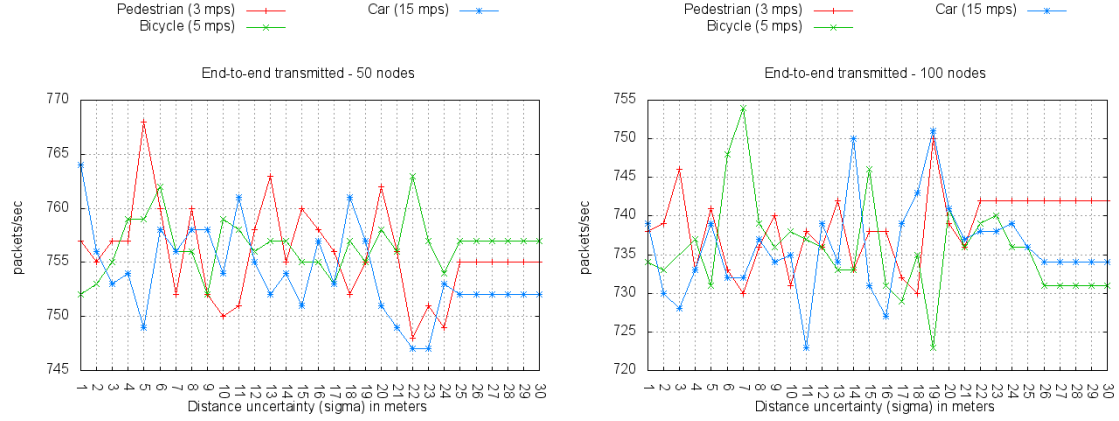


FIGURE 11.8 – End-to-end transmitted - 50/100 nodes

11.4 Conclusion

Dans ce chapitre, nous implémentons notre métrique dans le protocole *Dream* pour être celui que nous avons utilisé dans la Partie III et qu'il n'était pas justifié d'en chercher un autre. Dans nos tests, cependant, des 3 paramètres du contrat de qualité de service défini dans ce chapitre par :

- l'incertitude sur la distance entre 2 nœuds, caractérisée par son écart-type σ ;
- l'incertitude sur l'atténuation σ_η ;
- le bruit thermique N .

nous ne faisons varier que le premier : σ . Nous relevons que la croissance de la densité du réseau :

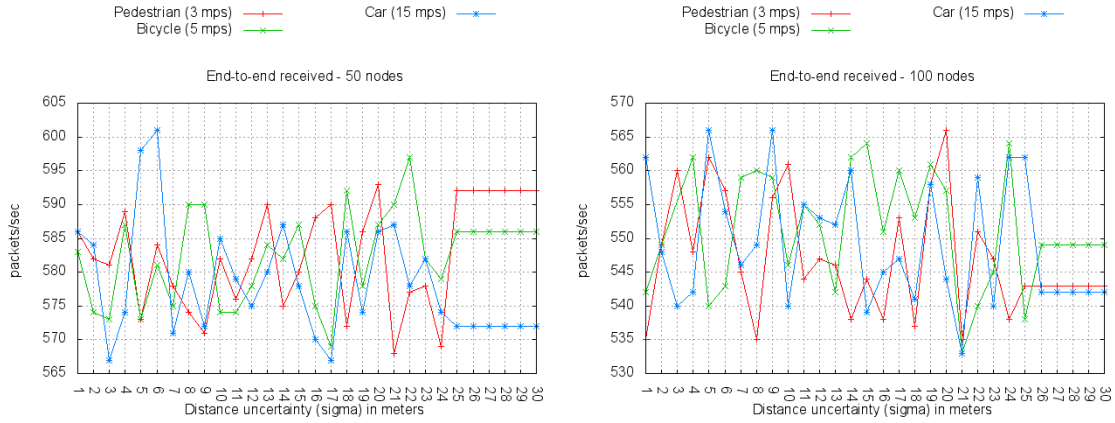


FIGURE 11.9 – End-to-end received - 50/100 nodes

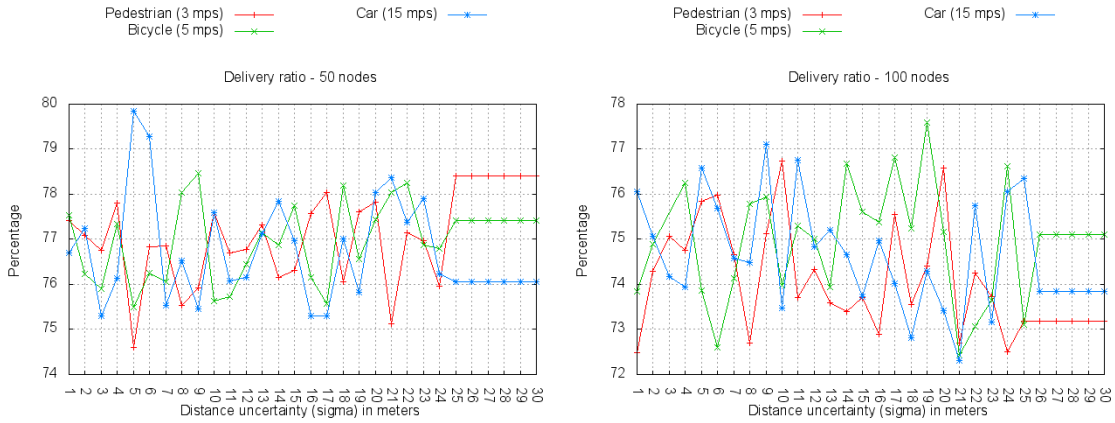


FIGURE 11.10 – Delivery ratio - 50/100 nodes

- limite l'impact de l'incertitude ;
- fait exploser le nombre de paquets de contrôle reçu et ce quelque soit l'incertitude.

En outre, le *concepteur de réseau* peut s'inspirer de nos courbes pour fixer l'incertitude qui présente le plus d'intérêt et éviter des dépenses consécutives au choix péremptoire d'un matériel offrant une précision supérieure pour une qualité moindre. En effet, toutes les mesures sans exception sont sensibles à l'incertitude.

Nous souhaitons poursuivre nos tests afin de prendre en compte également les variations des 2 autres paramètres σ_η et N . Enfin, il serait intéressant, pour les différents incertitudes et bruit choisis, de rapprocher le résultat de leurs tests à ceux de *Dream*.

11.4. CONCLUSION

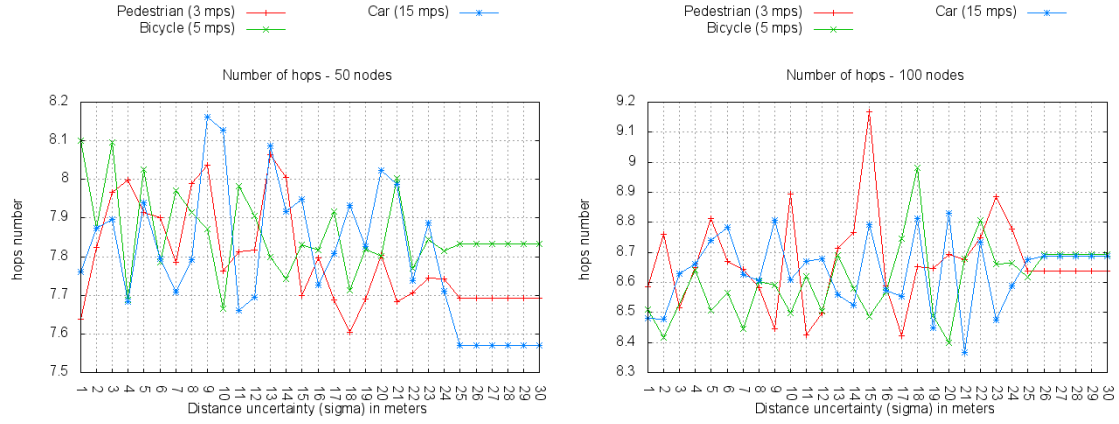


FIGURE 11.11 – Number of hops - 50/100 nodes

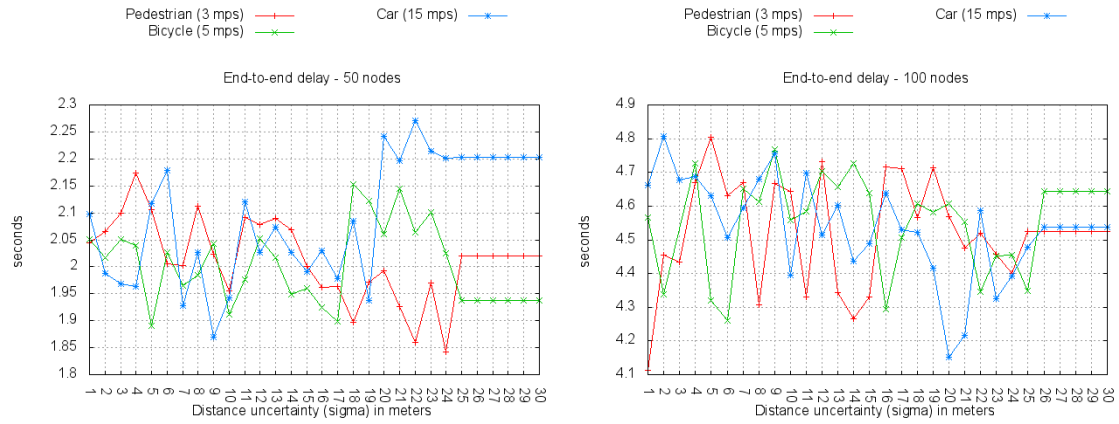


FIGURE 11.12 – End-to-end delay - 50/100 nodes

Cinquième partie

Conclusions

Chapitre 12

Conclusions de la thèse

12.1 Partie II : Mobilité

La notion de *Profil de Mobilité Utilisateur* y est introduit par quelques définitions. L'état de l'art qui suit, s'appuie sur des publications récentes dans ce domaine. Celles-ci font appel à des méthodes statistiques avancées mettant en œuvre :

- diverses *entropies* ;
- la *Maximisation de l'espérance* ;
- le *Filtrage* des données ;
- les *Chaînes de Markov à Temps Continu*.

Pour chacun de ses thèmes, nous avons donné leurs bases formelles accompagnées d'exemples.

Nous avons proposé une démarche nouvelle qui met en œuvre :

- les *Réseaux de Petri* spécialisés que sont les *Workflow-Nets* ;
- lesquels sont issus de l'analyse des traces de l'utilisateur par l'*Algorithme- α* ;
- et sur lesquels nous appliquons l'*algorithme de Lumpabilité*.

Nous proposons enfin une implémentation de notre démarche sous la forme d'un pseudo-code. Celui-ci pourra faire l'objet d'une implémentation dans un langage de programmation. Des campagnes de tests pourront alors être lancées afin d'asseoir notre démarche. Les traces utilisateur utilisées pourront être de différents profils. Nous aurons peut-être la chance d'utiliser des traces réelles. Toutefois, comme nous l'avons déjà publié dans un précédent travail (Costantini and Boumerdassi [2012]), les profils utilisateur issus de traces sont très limités dans l'espace et le temps. En revanche, définir un profil utilisateur en

termes de mixage de nos modèles de mobilité (cf. Sect. 4.5), nous apparaît plus pérenne.

Les modèles de mobilité suivent. En premier lieu, sont proposés 3 nouveaux modèles de mobilité sociaux. En second lieu, viennent leurs apports par rapport aux modèles existants, en soulignant le fait qu'en se rapprochant des comportements sociaux au travers de l'utilisation de la notion de Points d'attraction, ces modèles changent considérablement les résultats des tests de performances des protocoles de routage des réseaux mobiles. Les modèles de mobilité et le caractère incontournable de leur utilisation dans le domaine des réseaux mobiles sont présentés. L'objet de notre recherche est ensuite présenté, à savoir, l'importance de la prise en compte de la composante sociale dans l'étude et la modélisation des réseaux sans fil parce qu'en tout premier lieu, les appareils mobiles sont portés par des êtres humains. Afin d'asseoir ce point de vue, nous élaborons notre premier modèle de mobilité social, *Social Manhattan (SMN)* autour de la notion de points d'attraction. Un mécanisme qui puisse rendre compte du fait qu'un lieu perd de sa notoriété avec la raréfaction de ses visiteurs, trouve sa concrétisation en utilisant les Systèmes de Colonies de Fourmis (*Ant Colony System, ACS*) et leur mécanisme de phéromone. L'implémentation des modèles de mobilité a donné lieu à une charge de développement logiciel en Opnet Proto-C très importante. Une partie de ce travail est portée en annexe. Du point de vue recherche, ce travail a ouvert des voies qui sont au nombre des indices tangibles que nos tests ont mis en exergue : plus de 12% de moins au niveau des performances que ce qu'annonce les éditeurs de protocoles de routage. Tout d'abord du point de vue des modèles de mobilité sociaux : la preuve est faite qu'il y a là de quoi améliorer l'évaluation des protocoles des réseaux sans fil.

12.2 Partie III : Applications

Dans cette partie, la prise en compte de l'élément de la théorie du social que représente l'appartenance à un groupe communautaire fait l'objet de 2 implémentations dans 2 *Serveurs de localisation* :

- *Semi-Flooding Location Service, SFLS* ;
- *Simple Location Service, SLS*.

Le comportement de base d'*SFLS* est maintenu dans *Social SFLS*, *S-SFLS* à l'intérieur d'un groupe communautaire. À l'extérieur de celui-ci, le nombre de relaying de l'information de localisation est divisé par 2, par 3 ou par 4. Ce nouvel aspect est intégré au modèle. Il est ensuite implémenté en *C++*, sous *Network Simulator 2*, *NS2* en se basant sur le *service de localisation DREAM* qui fournit un cadre de travail simple et efficace pour le test des algorithmes d'inondation, *flooding*. Du point de vue des résultats, du côté du modèle, nous constatons que les pertes de gain occasionnées par la présence d'un groupe communautaire sont largement compensées par la réduction du nombre de relaying de l'information de localisation à l'extérieur du groupe communautaire. Du côté simulation, nous constatons qu'à nombre de paquets de données équivalents, nous avons un nombre de paquets de contrôle :

- moindre en passant de Dream à SFLS ;
- qui est proportionnel au diviseur de relaying.

Les résultats de l'évaluation du modèle, d'une part, et, d'autre part, ceux issus de la simulation vont donc dans le même sens : *Un gain évident de ressources est donc constaté avec un protocole qui est en définitive bien plus simple à mettre en œuvre que Dream*. Une évolution possible de notre *Social SFLS* consiste en le fait d'utiliser une courbe non linéaire pour décrire l'amoindrissement de la densité du réseau en fonction de l'éloignement du nœud i (le nœud qui transmet sa position) défini en nombre de sauts *hops*. Les pertes de gain d'*SFLS* sont en effet moindres dans ce cas. Nous souhaitons rappeler que les attraits de *SFLS* sont sa simplicité de faisabilité et mise en œuvre. Cela ne doit pas pour autant faire oublier que la raréfaction de l'actualisation des positions avec l'éloignement mutuel des nœuds peut être incompatible avec les contraintes de Qualité de Service, imposées par le cahier des charges du réseau sans fil. Néanmoins, il est clair que le partitionnement du réseau en communauté ne peut qu'être profitable à la Qualité de Service.

Ensuite, nous présentons la notion de *groupe d'appartenance* ajoutée au *Simple Location Service*, *SLS* qui, comme son nom l'indique, est un *service de localisation* élémentaire et simple à amender. Nous montrons que pour des petits nombres d'entrées dans *LP*, *SLS Social* offre de meilleures performances et pour des nombres d'entrées importants dans *LP*, *SLS Social* et *SLS* se valent. En conséquence, il est plus avantageux d'utiliser *SLS Social*.

Une évolution pourrait consister à, dans une optique probabiliste, prendre en compte le fait qu'il est clair que les nœuds d'un même groupe d'appartenance échangent davantage entre eux qu'avec l'extérieur de leur groupe. En conséquence, il serait envisageable de favoriser ces échanges en transmettant dans le groupe, les entrées du groupe présentent dans la table de localisation en priorité. Ainsi, nous gagnerions du point de vue de l'énergie pour peu de, voire aucune perte du point de vue de la communication. Enfin, nous souhaiterions analyser les statistiques du protocole de routage sous-jacent à Dream, SFLS et SLS à savoir SLC, notamment au niveau des délais d'acheminement des paquets.

12.3 Partie IV : Mesures

Dans cette partie, nous proposons une métrique de sélection du prochain saut ou relais pour les routages géographiques qui minimise une fonction de coût qui mesure la qualité du lien entre l'émetteur et le récepteur :

$$J = P_e + \frac{\rho}{P_r} \quad (12.1)$$

où ρ est un scalaire positif. Cette fonction de coût est minimale quand *error probability*, P_e est minimale et *progress of information*, P_r est maximale. En d'autres termes, $\arg \min_{(P_e, P_r)} J$, assure pour une valeur de ρ donnée, la progression la plus rapide du packet transmis (en termes de distance parcourue) pour le minimum d'erreur. Ces 2 mesures sont calculées à partir de la capacité du médium air prise en tant que variable aléatoire. Sont prises en compte les incertitudes sur l'atténuation (*fading noise*) et la position du relais ou nœud destination par rapport au nœud émetteur. Cela est associé au bruit thermique pour définir un contrat de qualité de service :

- l'incertitude sur la distance entre 2 nœuds, caractérisée par son écart-type σ ;
- l'incertitude sur l'atténuation σ_η ;
- le bruit thermique N .

Nous montrons que ses informations sont suffisantes pour l'optimisation. Des expressions facilement calculables sont déduites. Les incertitudes sont *gaussiennes*. Cependant, une extension assez simple nous permet de démontrer que des comportements équivalents sont obtenus à partir d'une *Rice distribution* de la distance. Nous implémentons notre métrique

dans le protocole *Dream* pour être celui que nous avons utilisé dans la Partie III et qu'il n'était pas justifié d'en chercher un autre. Dans nos tests, cependant, des 3 paramètres du contrat de qualité de service ci-dessus, nous ne faisons varier que le premier : σ . Nous relevons que la croissance de la densité du réseau :

- limite l'impact de l'incertitude ;
- fait exploser le nombre de paquets de contrôle reçu et ce quelque soit l'incertitude.

En outre, le *concepteur de réseau* peut s'inspirer de nos courbes pour fixer l'incertitude qui présente le plus d'intérêt et éviter des dépenses consécutives au choix péremptoire d'un matériel offrant une précision supérieure pour une qualité moindre. En effet, toutes les mesures sans exception sont sensibles à l'incertitude. Nous souhaitons poursuivre nos tests afin de prendre en compte également les variations des 2 autres paramètres σ_η et N . Enfin, il serait intéressant, pour les différents incertitudes et bruit choisis, de rapprocher le résultat de leurs tests à ceux de *Dream*.

12.4 Conclusions

Ce travail de thèse nous a apporté une ouverture inespérée sur des domaines de recherche aussi divers que les méthodes formelles, la mathématique du signal, la modélisation du social.

En outre, cela peut paraître un point de détail mais pour nous cela a été un véritable défi et donc une vraie fierté que cette thèse soit la première, sauf erreur ou omission de notre part, à mettre en contact les 2 mondes très cloisonnés d'Opnet Modeler et de Network Simulator 2, cf. 6.6.

Le nombre de retours en arrière et de remises en cause est incalculable et demeurera sans aucun doute pour nous l'apprentissage le meilleur que nous ayons jamais vécu en matière d'humilité et de patience.

Nous soutenons que le social va pénétrer encore davantage la sphère des sciences durs car cette voie est très prometteuse tant du point de vue des résultats déjà obtenus que de l'histoire que l'humanité vit actuellement en termes d'austérité. Il est clair à nos yeux que celle-ci ne peut que rapprocher les individus.

12.5 Perspectives

La mathématique¹ a toujours été pour nous, l'outil de formalisation le plus rigoureux et, pourquoi ne pas le dire, le plus esthétique.

Tout d'abord, la mathématique appliquée à la modélisation des déplacements humains a commencé notre formation de chercheur sous l'égide du Dr Selma Boumerdassi. Sa grande rigueur toute en douceur nous a conduit à produire la Partie III de cette thèse. Celle-ci va bien plus loin que la modélisation, comme le lecteur a pu le constater. Elle demeurera encore longtemps, nous l'espérons, le domaine de prédilection nos recherches futures.

Ensuite, la mathématique du traitement du signal et plus précisément celle de la couche PHY de notre domaine d'extraction, celui des réseaux sans fil, nous a été enseigné au plus haut niveau par le Pr Ruben Milocco et le travail de la Partie IV de cette thèse lui doit beaucoup en patience et application pour permettre le transfert de compétence depuis le signal vers les couches MAC et supérieures. Nous souhaitons malgré l'éloignement² avoir encore de nombreuses occasions de poursuivre cette collaboration.

Enfin, nous souhaitons vivement que les méthodes formelles deviennent notre domaine de spécialisation pour les années à venir. En effet, nous y avons pris goût lors de l'élaboration de la Partie II de ce travail de thèse, en travaillant en étroite collaboration avec le Pr Kamel Barkaoui, dont les conseils et la chaleur ne pouvaient que nous transmettre cette passion du domaine, cette vraie passion pour la Recherche : le plus beau de tous les rêves : rêver l'Homme !

1. En effet, nous tenons que les mathématiques ne font qu'une, à travers la topologie. Mais c'est une autre thèse !

2. Le Pr Ruben Milocco réside en Argentine.

Publications

Chapitre 13

Publications

Nos publications durant ce travail de thèse sont les suivantes :

- H. Costantini, S. Boumerdassi. *Social mobility models realism versus real traces*. Proceedings of the 2012 IEEE Wireless Communications and Networking Conference, WCNC 2012, Paris, France, April 1-4, 2012.
- H. Costantini, L. Harfouche, S. Boumerdassi. *Social Mobility Models using Ant Colony Systems*. Proceedings of the 22nd IEEE Personal Indoor Mobile Radio Communications (PIMRC'11 - WNHC). SEP. 2011.
- E. Renault, E. Amar, H. Costantini, S. Boumerdassi. *Semi-Flooding Location Service*. Proceedings of the 72nd IEEE Vehicular Technology Conference Fall (VTC 2010-Fall), 2010.
- E. Renault, E. Amar, H. Costantini, S. Boumerdassi. *Semi-Flooding Location Service : a cross-layer design*. Proceedings of the 7th ACM workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN '10), 2010.
- H. Costantini, E. Renault, S. Boumerdassi. *L'Internet du Futur dans les nuages*. Proceedings of the 2011 11th Annual International Conference on New Technologies of Distributed Systems (NOTERE 2011), mai, Paris, France, 2011.
- H. Costantini, E. Renault, S. Boumerdassi. *User-Centric in the Clouds to save our Future... Internet*. Proceedings of the 2011 International Conference on Networking

-
- and Future Internet (ICNFI 2011), Paris, France, 2011.
- H. Costantini, E. Renault, S. Boumerdassi. *L'Apport des Clouds à l'Internet du Futur*. Proceedings of the 2010 Congrès De Nouvelles Architectures pour la Communication (DNAC 2010), Paris, France, 2010.
 - H. Costantini, S. Boumerdassi. *802.11e standard in ad hoc network : QoS, disabling acknowledgment*. Proceedings of the 2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 2009), aug, Victoria, B.C., Canada, 2009.
 - H. Costantini, S. Boumerdassi, E. Renault. *Social Semi-Flooding Location Service*. Proceedings of the 2012 IFIP Wireless Days conference, November 21-23, Dublin, Ireland, 2012. (soumis)
 - H. Costantini, S. Boumerdassi. *OLSR community sensing*. Proceedings of the 2013 IEEE Wireless Communications and Networking Conference, WCNC 2013, Shanghai, China, April 7-10, 2012. (soumis)
 - H. Costantini, S. Boumerdassi. *Addition of ACS to Social Mobility Models and Implementation in Normal and Dense Networks*. Proceedings of the 2013 IEEE International Conference on Communications, ICC 2013, Budapest, Hungary, June 9-13. 2013 (soumis)

Bibliographie

Bibliographie

- R. Alhalimi and T. Kunz. Energy level accuracy in mobile ad-hoc networks using olsr. In *Proceedings of the 3rd international conference on Wireless internet, WICON '07*, pages 14 :1–14 :9, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN 978-963-9799-12-7. URL <http://portal.acm.org/citation.cfm?id=1460047.1460065>. 229
- J. An, Y. Ko, and D. Lee. A social relation aware routing protocol for mobile ad hoc networks. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1 –6, march 2009. doi : 10.1109/PERCOM.2009.4912879. 227
- S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '98*, pages 76–84, New York, NY, USA, 1998. ACM. ISBN 1-58113-035-X. doi : <http://doi.acm.org/10.1145/288235.288254>. URL <http://doi.acm.org/10.1145/288235.288254>. 67, 68, 86, 91
- M. Bayir, M. Demirbas, and N. Eagle. Discovering spatiotemporal mobility profiles of cellphone users. In *WOWMOM*, pages 1–9. IEEE, 2009. 4, 11, 15
- S. Boumerdassi, L. Harfouche, and E. Renault. Towards a social mobility model. In *20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2876–2880. IEEE Publisher, sep 2009. doi : 10.1109/PIMRC.2009.5450050. 43
- S. Boumerdassi, L. Harfouche, and E. Renault. Weighted social manhattan : Modeling and performance analysis of a mobility model. In *PIMRC'10*, pages 2019–2023, sep 2010. 46

- T. Camp. Location information services in mobile ad hoc networks. In *In Proceedings of the IEEE International Conference on Communications (ICC)*, pages 3318–3324, 2003. 119
- T. Camp, J. Boleng, and L. Wilcox. Location information services in mobile ad hoc networks. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 5, pages 3318 – 3324 vol.5, 2002. doi : 10.1109/ICC.2002.997446. 67, 69
- F. Chen, J. Liu, Z. Li, and Y. Wang. Routing with uncertainty in wireless mesh networks. In *Quality of Service (IWQoS), 2010 18th International Workshop on*, pages 1 –5, june 2010. doi : 10.1109/IWQoS.2010.5542719. 98
- Vittoria Colizza and Alessandro Vespignani. Epidemic modeling in metapopulation systems with heterogeneous coupling pattern : Theory and simulations. *Journal of Theoretical Biology*, 251 :450–467, 2008. 5, 12, 16
- P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 26(5) :748–760, June 2008. ISSN 0733-8716. doi : 10.1109/JSAC.2008.080602. URL <http://dx.doi.org/10.1109/JSAC.2008.080602>. 226, 227
- H. Costantini and S. Boumerdassi. Social mobility models realism versus real traces. In *WCNC*, pages 1841–1846. IEEE, 2012. ISBN 978-1-4673-0436-8. 40, 129
- H. Costantini, L. Harfouche, and S. Boumerdassi. Social mobility models using ant colony systems. In *22nd IEEE Personal Indoor Mobile Radio Communications (PIMRC'11 - WNHG)*, Toronto, Canada, sep 2011. 5, 12, 15
- E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '07, pages 32–40, New York, NY, USA, 2007a. ACM. ISBN 978-1-59593-684-4. doi : <http://doi.acm.org/10.1145/1288107.1288113>. URL <http://doi.acm.org/10.1145/1288107.1288113>. 226
- E.M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad*

- hoc networking and computing*, MobiHoc '07, pages 32–40, New York, NY, USA, 2007b. ACM. ISBN 978-1-59593-684-4. doi : <http://doi.acm.org/10.1145/1288107.1288113>. URL <http://doi.acm.org/10.1145/1288107.1288113>. 226
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1) :1–38, 1977. 20, 23
- S. Derisavi, H Hermanns, and W. H. Sanders. Optimal state-space lumping in markov chains. *Information Processing Letters*, 87(6) :309 – 315, 2003. ISSN 0020-0190. doi : 10.1016/S0020-0190(03)00343-0. URL <http://www.sciencedirect.com/science/article/pii/S0020019003003430>. 35
- B. Divecha, A. Abraham, C. Grosan, and S. Sanyal. Impact of node mobility on manet routing protocols models. 154
- M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992. 47
- G. Eason, B. Noble, and I. N. Sneddon. On certain integrals of lipschitz-hankel type involving products of bessel functions. *Philosophical Transactions of the Royal Society of London*, 247(935) :529–551, april 1955. doi : 10.1098/rsta.1955.0005. 42
- R. Fano. *Transmission of Information : A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961. 19
- B. Flury and A. Zoppe. Exercises in EM. 54(3) :207–??, August 2000. ISSN 0003-1305 (print), 1537-2731 (electronic). URL <http://www.amstat.org/publications/tas/Flury.htm>. 27
- N. Ghanem, S. Boumerdassi, and E. Renault. New energy saving mechanisms for mobile ad-hoc networks using olsr. In *Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, PE-WASUN '05, pages 273–274, New York, NY, USA, 2005. ACM. ISBN 1-59593-182-1. doi : <http://doi.acm.org/10.1145/1089803.1090006>. URL <http://doi.acm.org/10.1145/1089803.1090006>. 229, 230

- Network Working Group. Optimized link state routing protocol (olsr), oct 2003. 156
- H. Gudbjartsson and S. Patz. The rician distribution of noisy mri data. *Magnetic Resonance in Medicine*, 34(6) :910–914, 1995. URL <http://dx.doi.org/10.1002/mrm.1910340618>. 105
- M. Günes, U. Sorges, and I. Bouazizi. Ara - the ant-colony based routing algorithm for manets. In *International Conference on Parallel Processing Workshops, ICPPW*, page 79, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1680-7. doi : <http://dx.doi.org/10.1109/ICPPW.2002.1039715>. 47
- M. R. Gupta and Y. Chen. Theory and use of the em algorithm. *Found. Trends Signal Process.*, 4(3) :223–296, March 2011. ISSN 1932-8346. doi : 10.1561/20000000034. URL <http://dx.doi.org/10.1561/20000000034>. 20
- H.O. Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, 14(2) : 174–194, 1958. 20, 21, 23
- P.S. Heck and S. Ghosh. A study of synthetic creativity : Behavior modeling and simulation of an ant colony. *IEEE Intelligent Systems*, 15(6) :58–66, 2000. ISSN 1541-1672. doi : <http://dx.doi.org/10.1109/5254.895863>. 41
- M. Isaksson. *Applications of Gaussian Noise Stability in Inapproximability and Social Choice*. PhD thesis, Chalmers University of Technology and Göteborg University, 2009. 7, 99
- I. S. Jacobs and C. P. Bean. Fine particles, thin films and exchange anisotropy, 1963. 42
- A.P. Jardosh, E.M. Belding-Royer, K.C. Almeroth, and S. Suri. Real-world environment models for mobile network evaluation. *Selected Areas in Communications, IEEE Journal on*, 23(3) :622 – 632, 2005. ISSN 0733-8716. doi : 10.1109/JSAC.2004.842561. 41, 42
- G. Jayakumar and G. Gopinath. Performance comparison of manet protocol based on manhattan grid model. *Journal of Mobile Communications*, 2.1 :18–26, 2008. doi : <http://dx.doi.org/10.3923/jmcomm.2008.18.26>. URL <http://www.medwelljournals.com/abstract/?doi=jmcomm.2008.18.26>. 43, 154

- J.G. Kemeny and J.L. Snell. *Finite Markov Chains : With a New Appendix "Generalization of a Fundamental Matrix"*. Undergraduate Texts in Mathematics. Springer, 1976. ISBN 9780387901923. URL <http://books.google.fr/books?id=0bTK5uWzbYwC>. 32
- W. Kiess, H. Fussler, J. Widmer, and M. Mauve. Hierarchical location service for mobile ad-hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(4) :47–58, October 2004. ISSN 1559-1662. doi : 10.1145/1052871.1052875. URL <http://doi.acm.org/10.1145/1052871.1052875>. 67
- H. Kim and H. Y. Song. Daily life mobility of a student : From position data to human mobility model through expectation maximization clustering. In T.-H. Kim, H. Adeli, W. I. Grosky, N. Pissinou, T. Shih, E. J. Rothwell, B. H. Kang, and S.-J. Shin, editors, *FGIT-MulGraB (2)*, volume 263 of *Communications in Computer and Information Science*, pages 88–97. Springer, 2011. ISBN 978-3-642-27185-4. 18, 19
- H. Kim and H. Y. Song. Maximum likelihood estimation from incomplete data in a form of continuous time markov chain. *Procedia Computer Science*, 3(0) :1–7, 2012. 19, 31
- I. Kleinhans. *Économie & Finances Suite101*, 2010. URL <http://suite101.fr/article/la-personnalisation-des-produits-et-services-a8807>. 5, 12, 16
- A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7 :19–20, July 2003. ISSN 1559-1662. doi : <http://doi.acm.org/10.1145/961268.961272>. URL <http://doi.acm.org/10.1145/961268.961272>. 226
- Y. Ling, L. Zhang, and M. Gao. Design and implementation of process mining system based on a-algorithm. In *Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on*, page 308, june 2008. doi : 10.1109/ICICIC.2008.215. 36
- D. Liu, X. Jia, and I. Stojmenovic. Quorum and connected dominating sets based location service in wireless ad hoc, sensor and actuator networks. *Comput. Commun.*, 30(18) : 3627–3643, 2007. ISSN 0140-3664. doi : <http://dx.doi.org/10.1016/j.comcom.2007.07.001>. 67

- A. Marsan. *Modelling with generalized stochastic Petri nets*. Wiley series in parallel computing. Wiley, 1995. ISBN 9780471930594. URL <http://books.google.fr/books?id=ZN5QAAAAAAAJ>. 17
- B. Mathern, A. Mille, and T. Bellet. Rendre interactive la découverte d'automates à partir de traces d'activités. 2011. URL <http://liris.cnrs.fr/Documents/Liris-5131.pdf>. 37
- M. Mauve, A. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6) :30 –39, nov/dec 2001. ISSN 0890-8044. doi : 10.1109/65.967595. 64
- J. C. Maxwell. *A treatise on electricity and magnetism*. Clarendon Press, Oxford, 1873. 42
- P.G. More. *Statistical Methods, Applied to Experiments in Agriculture and Biology*. Biometrika Trust, 1952. URL <http://www.jstor.org/discover/10.2307/2334021?uid=3738016&uid=2129&uid=2134&uid=2&uid=70&uid=4&sid=47699090746787>. 21
- S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam. Mobisim : A framework for simulation of mobility models in mobile ad-hoc networks. *Wireless and Mobile Computing, Networking and Communication, IEEE International Conference on*, 0 :82, 2007. doi : <http://doi.ieeecomputersociety.org/10.1109/WIMOB.2007.52>. 41
- M. Musolesi and C. Mascolo. Mobility models for systems evaluation. *Middleware for Network Eccentric and Mobile Applications*, 3 :43–62, 2009. doi : 10.1007/978-3-540-89707-1_3. URL http://dx.doi.org/10.1007/978-3-540-89707-1_3. 41
- J. R. Norris. *Markov Chains (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, July 1998. URL <http://www.worldcat.org/isbn/0521633966>. 30
- A. Papoulis and S.U. Pillai. *Probability, random variables, and stochastic processes*. McGraw-Hill electrical and electronic engineering series. McGraw-Hill, 2002. ISBN 9780073660110. URL <http://books.google.fr/books?id=YYwQAQAIAAJ>. 105

- H. Peng, G. Kitagawa, M. Gan, and X. Chen. A new optimal portfolio selection strategy based on a quadratic form mean–variance model with transaction costs. *Optimal Control Applications and Methods*, 32(2) :127–138, 2011. ISSN 1099-1514. doi : 10.1002/oca.936. URL <http://dx.doi.org/10.1002/oca.936>. 105
- C. Perkins, E. Royer, and S. Das. Rfc 3561 ad hoc on-demand distance vector (aodv) routing. Technical report, IETF, 2003. URL <http://tools.ietf.org/html/rfc3561>. 156
- C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, SIGCOMM '94, pages 234–244, New York, NY, USA, 1994. ACM. ISBN 0-89791-682-4. doi : <http://doi.acm.org/10.1145/190314.190336>. URL <http://doi.acm.org/10.1145/190314.190336>. 227
- C. A. Petri. *Communication with automata*. PhD thesis, Universität Hamburg, 1966. 16, 31
- C. Rao. *Linear statistical inference and its applications*. Wiley series in probability and mathematical statistics. John Wiley and Sons, 1965. URL http://books.google.fr/books?id=zio_RAAACAAJ. 23
- E. Renault, E. Amar, H. Costantini, and S. Boumerdassi. Semi-flooding location service. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pages 1–5, sept. 2010a. doi : 10.1109/VETECF.2010.5594281. 71
- E Renault, H. Costantini, E. Amar, and S. Boumerdassi. Semi-flooding location service : a cross-layer design. In *Proceedings of the 7th ACM workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, PE-WASUN '10, pages 44–49, New York, NY, USA, 2010b. ACM. ISBN 978-1-4503-0276-0. doi : <http://doi.acm.org/10.1145/1868589.1868599>. URL <http://doi.acm.org/10.1145/1868589.1868599>. 71
- I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. In *INFOCOM*, pages 924–932. IEEE, 2008. URL <http://dblp.uni-trier.de/db/conf/infocom/infocom2008.html#RheeSHLC08>. 19

- K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *In ACM SenSys '04*, pages 108–121, 2004. 97
- G. W. Snedecor. *Statistical Methods, Applied to Experiments in Agriculture and Biology*. The Iowa State College Press, 1956. 21
- C. Song, Ze. Qu, N. Blumm, and A.-L. Barabasi. Limits of predictability in human mobility. *Science*, 327(5968) :1018–1021, February 2010. ISSN 1095-9203. doi : 10.1126/science.1177170. URL <http://dx.doi.org/10.1126/science.1177170>. 16, 18, 19
- H. Y. Song and H. G. Kim. Filtering erroneous positioning data with moving window approach. In *Proceedings of the 11th international conference on Telecommunications and Informatics, Proceedings of the 11th international conference on Signal Processing, SITE'12*, pages 47–52, Stevens Point, Wisconsin, USA, 2012. World Scientific and Engineering Academy and Society (WSEAS). ISBN 978-1-61804-081-7. URL <http://dl.acm.org/citation.cfm?id=2230716.2230727>. 19, 29
- L. Song, D. Kotz, R. Jain, and Xiaoning He. Evaluating location predictors with extensive wi-fi mobility data. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1414 – 1424 vol.2, march 2004. doi : 10.1109/INFCOM.2004.1357026. 98
- I. Stojmenovic, D. Liu, and X. Jia. A scalable quorum-based location service in ad hoc and sensor networks. *Int. J. Commun. Netw. Distrib. Syst.*, 1(1) :71–94, February 2008. ISSN 1754-3916. doi : 10.1504/IJCND.2008.017205. URL <http://dx.doi.org/10.1504/IJCND.2008.017205>. 67
- S.-F. Su. *The UMTS Air-Interface in RF Engineering : Design and Operation of Umts Networks*. McGraw-Hill, 2007. ISBN 9780071488662. URL <http://books.google.fr/books?id=0V50yMZi9F8C>. 104
- K. L. Thorpe. *Markov Chain Models of Social Mobility*. PhD thesis, Middlebury College, 2007. 31

- W. Van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011. ISBN 978-3-642-19344-6. 32
- W. Van Der Aalst and B. Van Dongen. Discovering workflow performance models from timed logs. In *Proceedings of the First International Conference on Engineering and Deployment of Cooperative Information Systems*, 2002. 16, 32, 37
- B. Van Dongen and W. Van der Aalst. Process mining tool. In *Proceedings of Applications and Theory of Petri Nets 2004 : 25th International Conference, ICATPN 2004, Bologna, Italy, June 21-25, 2004 — Volume 3099 of Lecture Notes in Computer Science / Cortadella, Reisig (Eds.)*, pages 454–463. Springer-Verlag, sep 2004. 36
- T. Vincenty. *Survey Review*, volume XXIII. Directorate of Overseas Surveys of the Ministry of Overseas Development, april 1975. URL http://www.ngs.noaa.gov/PUBS_LIB/inverse.pdf. 29
- Vodafone. Vodafone : Mobility user profiling, 2012. URL http://enterprise.vodafone.com/products_solutions/managed_services/mobility_user_profiling.jsp. 4, 15
- S. Wiehe, A. Carroll, G. Liu, K. Haberkorn, S. Hoch, J. Wilson, and J. Fortenberry. Using gps-enabled cell phones to track the travel patterns of adolescents. *International Journal of Health Geographics*, 7(1) :22+, May 2008. URL <http://www.hubmed.org/display.cgi?uids=18495025>. 5, 12, 16
- E. Yoneki, P. Hui, and J. Crowcroft. Bio-inspired computing and communication. chapter Wireless Epidemic Spread in Dynamic Human Networks, pages 116–132. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 978-3-540-92190-5. doi : http://dx.doi.org/10.1007/978-3-540-92191-2_11. URL http://dx.doi.org/10.1007/978-3-540-92191-2_11. 226
- A. M. Zakharin. Lumping the states of markov and semi-markov processes and the use of the lumping operation for the investigation of stochastic systems. *Cybernetics and Systems Analysis*, 8 :589–598, 1972. ISSN 1060-0396. URL <http://dx.doi.org/10.1007/BF01068280>. 10.1007/BF01068280. 35

BIBLIOGRAPHIE

Annexes

Annexe A

Implémentation de nos modèles de mobilité

A.1 SMN

A.1.1 Implémentation

Cette section est un résumé du travail d'implémentation décrit en détail à l'annexe B.

Afin d'évaluer nos modèles, nous devons choisir :

1. un cadre de travail parmi ceux existant en matière de mobilité ;
2. un outil de simulation.

Le cadre de travail que nous choisissons est celui du modèle de mobilité *Manhattan* car de très loin le plus répandu dans la littérature.

L'outil de simulation que nous choisissons est *Opnet Modeler* car il s'agit d'un outil commercial abouti tant du point de vue du fond que de la forme. Sur le fond, il suit de près les dernières évolutions des réseaux d'entreprise. Sur la forme, il propose un certain nombre d'éditeurs et notamment une interface homme-machine visuelle d'élaboration des modèles de simulation.

Dans la version d'*Opnet Modeler* destinée aux académies, seul le modèle *Random Way-point (RWP)* est disponible. Il nous faut donc développer nous-même le modèle *Manhattan*.

Initialement, nous pensons pouvoir prendre comme point de départ les principes de développement utilisés dans *RWP*.

Toutefois, nous sommes confrontés au *problème des intersections manquées* ce qui représente un verrou technique majeur.

Un exposé détaillé du problème et de son solutionnement est présent à l'annexe B.

Brièvement, *Opnet Modeler* traduit toutes les coordonnées dans son *système de coordonnées géographiques*.

Pour tous les modèles de mobilité aléatoires sans exception, les erreurs relatives aux conversions du *système de coordonnées cartésiennes* vers celui-ci passent tout à fait inaperçues.

En revanche, dès qu'on passe à un modèle de mobilité où les décisions sont prises aux intersections (continuer tout droit ou tourner ; se diriger vers un Point d'Attraction ou un autre...), le fait de « manquer » une intersection fausse tous les résultats ! C'est exactement le problème auquel nous faisons face.

Le solutionnement proposé consiste à ne pas s'appuyer sur l'infrastructure prévue par *Opnet* pour les modèles de mobilité, mais de créer nous-même une nouvelle infrastructure autour du *système de coordonnées cartésiennes*.

Nous commençons l'implémentation du modèle de mobilité *Freeway* [Divecha et al.], Fig. A.1c.

Le modèle *Freeway* est un modèle uni-directionnel (chaque ligne/rue ayant une *Orientat*ion). Périodiquement, un nœud choisit une vitesse et se déplace d'une étape correspondant à *Move Step* mètres. Sa vitesse est aléatoire, sauf quand il rentre dans la distance de sécurité de son prédécesseur, *Safety Distance*. Dans ce cas, la vitesse est divisée par 2. Aucun nœud ne peut dépasser un autre nœud.

Sur cette base, nous développons le modèle *Manhattan*, Fig. A.1d. Il suppose une grille représentant des routes bi-directionnelles (plus besoin d'*Orientat*ion de la ligne/colonne). L'aire de simulation est définie par ses coordonnées *min* et *max* sur le terrain (x_{min}, y_{min}), (x_{max}, y_{max}). Les lignes et les colonnes sont séparées par la longueur d'un pâté de maison, *Block Length*, en mètres. À une intersection, un nœud continue tout droit avec une probabilité 0,5, tourne à gauche ou à droite avec une probabilité de 0,25 pour chaque coté comme indiqué dans Jayakumar and Gopinath [2008].

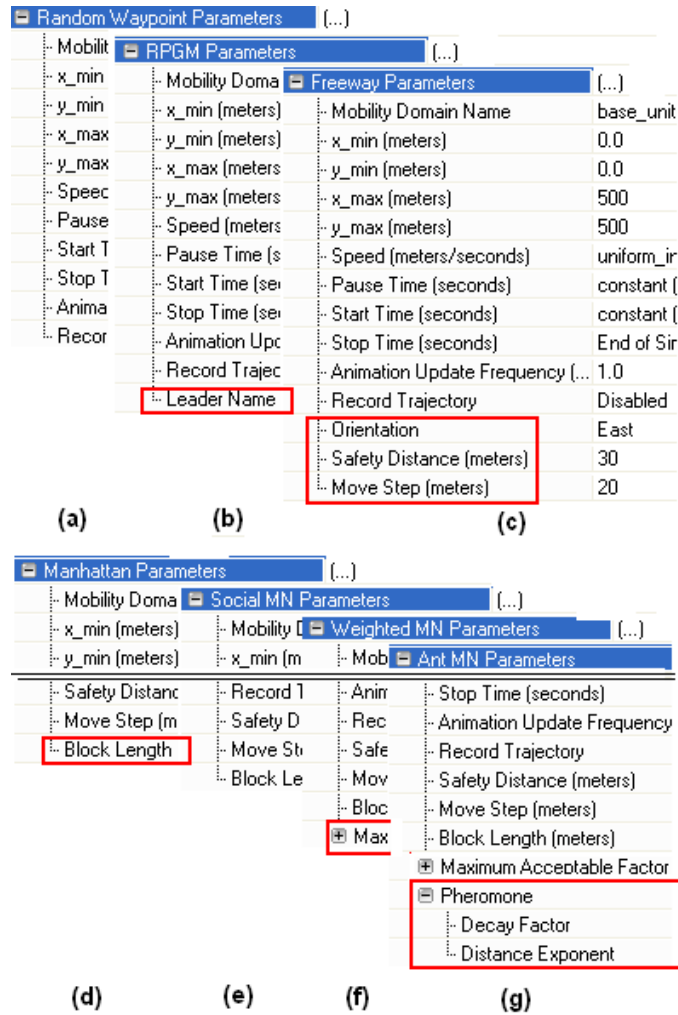


FIGURE A.1 – Attributs Mobility Config Ext

Puis vient l'implémentation de notre modèle *Social Manhattan*, voir Fig. A.1e, A.2b et A.3a.

Il ajoute des Points d'Attraction avec des *attractivités* fixes. Ceci est modélisé dans un nouveau composant *Attraction Point*, voir Fig. A.2b.

A.1.2 Test

Le caractère socio-culturel des relations humaines est-il un élément déterminant dans la modélisation du déplacement des humains? Dans les sections précédentes, nous avons apporté des éléments de réponse tant du point de vue de l'observation que de celui des

FIGURE A.2 – Nouveaux objets visuels *Opnet Modeler*

name	Attraction point 01	name	Station 05
model	manet_station_attraction_point	model	manet_station_social
x position	200	x position	200
y position	200	y position	320
Attractivity	5.0	Weight	5.0

(a)

(b)

FIGURE A.3 – Attributs Sociaux

mathématiques, qui nous ont permis de mesurer l'impact que certains caractères sociaux, représentés par nos Points d'Attraction, ont sur la modélisation du déplacement.

Notre premier modèle *Social Manhattan Mobility Model (SMN)* établi, a-t-il un réel intérêt pratique ? Sera-t-il à même de remettre en cause tout ou partie des résultats obtenus par la démarche usuelle, majoritairement aléatoire (*MN* ou *RWP*) ? Notre intuition nous porte à penser que la réponse est également affirmative. Pour confirmer cette intuition, ou l'infirmier, nous procédons à une série de tests qui nous permettent de répondre de proche en proche à nos questionnements issus directement ou indirectement de notre motivation première.

Notre objectif dans cette section est d'évaluer l'influence de notre modèle *SMN* sur les performances des protocoles réseaux parmi lesquels les protocole de routage constituent une branche très importante. Nous étudions ici les protocoles de routage les plus utilisés à savoir *Optimized Link State Routing Protocol (OLSR)* pour le cas proactif [Group 2003] et *Ad hoc On Demand Distance Vector (AODV)* pour le cas réactif [Perkins et al. 2003]. Rappelons ce que signifie les adjectifs réactif et proactif qui peuvent qualifier un protocole de routage réseau.

Protocoles de routage réactifs

Ces protocoles réagissent aux anomalies, par exemple, en cas de routes non fonction-

nelles. Ils sont moins exigeants en temps processeur. L'exemple le plus courant est le protocole de routage *AODV*, *Ad hoc On Demand Distance Vector*. Ce protocole est conçu pour les réseaux mobiles Ad Hoc. Il permet un routage dynamique, automatique et « multi-sauts » entre les ordinateurs. Il est défini dans la *RFC 3561*. Le projet *AODV@IETF* est rendu possible par la collaboration des laboratoires *MOMENT* et *NMSL* de *Santa Barbara* et de *Intel R&D*.

Protocoles de routage proactifs

Ces protocoles vérifient l'état des liens et mettent à jour leurs tableaux en anticipant toute demande de route. C'est pourquoi il est proactif : il agit par anticipation. Il est complexe et exigeant en temps processeur. Il est en revanche performant. L'exemple le plus courant est *OLSR*, *Optimized Link State Routing Protocol* qui fait l'objet de la *RFC 3626*. Il est conçu pour les réseaux mobiles Ad Hoc. Il est dit « dirigé par les tableaux » et utilise une technique appelée *MRP*, *multipoint relaying* pour acheminer les messages. C'est l'un des protocoles les plus stables et prometteurs.

Configurations de test

Nous déclinons nos tests selon les critères suivants :

- modèle de mobilité ;
- paramètres du modèle ;
- densité du réseau mobile ;
- nombre de Points d'Attraction.

Nous rapprochons notre modèle social (*SMN*) du modèle de mobilité le plus courant dans la communauté scientifique, à savoir Manhattan (*MN* ou *MAN*) pour mettre en perspective les apports de notre modèle.

Il va de soi que ce rapprochement nécessite un *cadre de test* commun. C'est la raison pour laquelle nous décidons de fixer certains paramètres de simulation. Il s'agit des paramètres suivants, nous utilisons ci-dessous leurs équivalents dans l'implémentation :

- *Safety Distance* : distance de sécurité routière ; valeur : 30 m, valeur moyenne en zone

urbaine ;

- *Move Step* : unité de mouvement d'un nœud mobile ; valeur : 25 m, pour ne pas alourdir les temps de simulation ; diviseur du *Block Length* afin que les nœuds « passent » par les intersections ;
- *Block Length* : distance entre 2 lignes ou 2 colonnes de la grille ; valeur : 50 m, largeur moyenne d'un pâté de maisons ;
- $x_min, y_min, x_max, y_max$: aire de simulation ; valeur : 1 km^2 , superficie moyenne d'un quartier.

Le paramètre *Attractivity*, popularité d'un Point d'Attraction n'est relatif qu'à notre modèle social *SMN*. Pour nos tests sa valeur est fixée à 5. Elle est issue des tests préliminaires qui ont mis en évidence les contraintes suivantes. Si la valeur est trop grande, la distance devient négligeable. Inversement, si elle est trop petite, c'est elle-même qui devient totalement négligeable par rapport à la distance. D'autres séries de tests seront nécessaires afin de déterminer comment faire en sorte que les Points d'Attraction aient des attractivités initiales différentes les unes des autres, en tenant compte de ces contraintes.

Les paramètres de simulation variables sont ceux qui nous permettent la mise en perspective de nos modèles. Les valeurs choisies nous apparaissent suffisantes dans un premier temps pour connaître l'influence des paramètres correspondants sur les modèles étudiés. Il s'agit de :

- protocole de routage ; valeurs : OLSR et AODV, les plus utilisés dans les réseaux sans fil et les plus représentatifs des grands ensembles que sont les protocoles réactifs et les protocoles proactifs, respectivement ;
- débit ; valeurs : 11, 24 et 54 Mbit/s, un débit faible, un autre moyen, un débit plus important ;
- vitesse *maximale* des nœuds ; valeurs : 2, 3 et 15 m/s soit 7 km/h pour les piétons, 11 km/h pour les bicyclettes et 52 km/h pour les véhicules urbains ;
- nombre de nœuds ; valeurs : 20 nœuds, un groupe moyen (classe d'école...) et 40 nœuds, un groupe plus important (groupe de touristes, conférence...) ;
- nombre de Points d'Attraction ; valeurs : 4, quartier périphérique et 8, centre ville ;

Les différentes valeurs de nos paramètres de simulation sont résumés en Table A.1.

TABLE A.1 – Simulation

Simulateur	OPNET Modeler 16.0
Couche MAC	IEEE 802.11
Taux de transmission	250 m
Type de trafic	UDP
Destination	aléatoire
Caractéristiques physiques	Extended Rate PHY (802.11g)
Protocoles de routage	OLSR, AODV
Débit (Mbit/s)	11, 24, 54
Vitesse des nœuds (m/s)	2 (piéton), 3 (bicyclette), 15 (véhicule)
Nombre de nœuds	20, 40
Aire de simulation	1000 m $\times$ 1000 m
Block length	50 m
Safety distance	30 m
Move step	25 m
Nombre de Points d'Attraction	4, 8
Attractivity	5
Modèle de mobilité	Manhattan (MN), Social Manhattan (SMN)

A.1.3 Résultats

Dans cette section, nous relatons les tests de simulation effectués selon les prérogatives de la section précédente. Les modèles *Manhattan* (*MN* ou *MAN*) et *Social Manhattan* (*SMN*) sont rapprochés afin de mettre en évidence leurs différences.

Les mesures reportées sont :

- *Load (Mbps)* : moyenne des *Mbit/s* envoyés par les couches hautes vers la couche *MAC* pour l'ensemble du réseau ;
- *Delay (s)* ou *Dlay* : moyenne en secondes des délais d'acheminement des paquets depuis la couche *MAC* du nœud émetteur jusqu'à la couche *MAC* du nœud récepteur ;
- *Throughput (Mbps)* ou *Thru* : moyenne des *Mbit/s* reçus par la couche *MAC* des nœuds en provenance des couches basses ;
- *Data dropped (Retry Threshold Exceeded) (Mbps)* ou *DrpR* : moyenne des *Mbit/s* abandonnés par la couche *MAC* pour cause de bourrage à l'émission (médium radio non disponible).

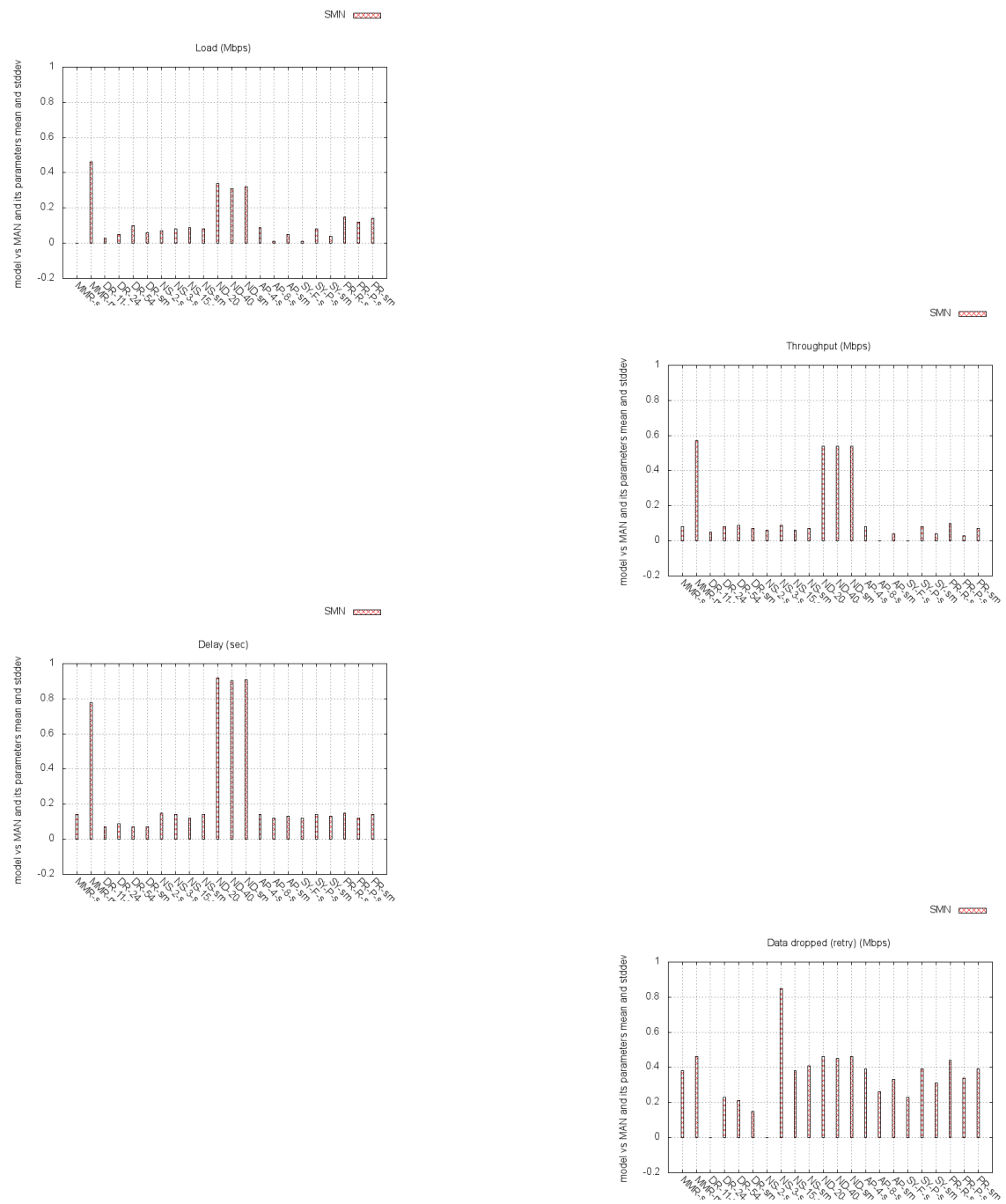


FIGURE A.4 – SMN - Variables statistiques réseau

La Fig. A.4 rassemble les graphiques suivants : Load, Throughput (Thru), Delay (Dlay), Data dropped (Retry Threshold Exceeded, DrpR).

La Table suivante en donne la légende.

MMR	"Model me mean" to "MAN me mean" Ratio with Model in {SMN,WMN,AM1,AM2,AM3} and me in {Load, Thru, Dlay, DrpR}
MMR-s	stddev of MMR of ALL scenarios
MMR-m	mean of MMR of ALL scenarios
DR-n	Data Rate of n Mbps with n in {11,24,54}
NS-n	Node Speed in range 0-n mps with n in {2,3,15}
ND-n	Network Density of n nodes with n in {20,40}
AP-n	Number of Attraction Points with n in {4,8}
SY-t	Sociability Type with t in {(F)ull,(P)artial}
PR-t	Protocol Type with t in {(R)eactive,(P)roactive}
X	MMR of All scenarios covered by X with X in {DR,NS,ND,AP,SY,PR}
X-n	with n in {relative parameter value, see above}
X-n-s	stddev of X-n
X-sm	mean of all X-n-s

Un graphique comporte 22 histogrammes. Notons tout d'abord *MMR*, le rapport ou *ratio* du modèle considéré sur le modèle Manhattan de la variable statistique considérée. Un histogramme donne les moyenne et écarts des MMR, d'une part, de façon globale à tous les scénarios de test, puis, d'autre part, par paramètre, pour chacune de ses valeurs, pour les scénarios concernés. C'est cette dernière « moyenne d'écarts » qui dénote l' « influence » du paramètre sur l'ensemble de la mesure. Une précision concernant la Sociabilité s'impose : le caractère entièrement social ((*F*)*ull*) signifie que tous les nœuds sont respectueux du modèle de mobilité en vigueur ; tandis que la présence d'éléments en « rébellion » ou asociaux ((*P*)*artial*) est modélisée par environ 15 % des nœuds qui manquent totalement de respect à cet ordre préétabli.

Notons en premier lieu que :

La moyenne globale de Load lors du passage de Manhattan (MAN) à Social Manhattan (SMN) est de : -0,13. Cela signifie que Manhattan est surévalué en l'absence de prise en compte de l'aspect social.

La Table suivante résume l'« influence » des paramètres selon les prérogatives suivantes concernant l'écart global d'un paramètre (dénnoté par X-s, conférer la Table légende ci-dessus et l'axe des abscisses des graphiques) :

- inférieur à 0,2 : influence nulle ;
- entre 0,2 et 0,6 : influence faible ;
- entre 0,6 et 1 : influence moyenne ;
- supérieur à 1 influence forte.

	Load	Throughput	Delay	Data dropped
Debit	nulle	moyenne	forte	forte
Vitesse	faible	moyenne	forte	forte
Densité	faible	moyenne	forte	forte
Nombre de Points d'Attraction	nulle	faible	forte	forte
Sociabilité	nulle	faible	forte	forte
Réactif / Proactif	nulle	moyenne	forte	forte

À noter que le cas extrême où il n'y a pas de Point d'Attraction, à savoir Manhattan (MN ou MAN) peut être assimilé au cas où il y a une infinité de Points d'Attraction. En effet, dans ce cas, les nœuds sont distribués aléatoirement sans prise en compte de leurs éventuels liens sociaux.

Quand le nombre de Points d'Attraction diminue, la concentration des nœuds augmente. Cela a pour effet de réduire le délai de transmission de bout en bout. En effet, il n'y a pratiquement plus besoin de nœuds intermédiaires. En revanche, nombre de données

sont perdues pour cause de bourrage du médium air (Data dropped). L'explication est la suivante : la bande de fréquence étant fixe, l'augmentation de la densité géographique des nœuds résulte en une augmentation du nombre de collisions.

A.2 Réseaux pondérés

A.2.1 Implémentation

Nous développons le modèle *Weighted Social Manhattan Mobility Model (WMN)* sur la base de notre modèle *Social Manhattan Mobility Model (SMN)*. Le nouveau modèle *WMN* ajoute :

- Le *weight*, i.e. poids social de l'utilisateur, au nœud mobile, ce poids est implémenté par le composant *Social Manet Station* qui est une extension du composant *Opnet Manet Station* ;
- Le *Maximum Acceptable Factor*, qui limite l'accroissement de l'attractivité d'un Point d'Attraction.

Notre objectif dans ce qui suit est d'évaluer l'influence de notre modèle *WMN* sur les performances des protocoles de routage et le comparer avec *SMN* et *MN* de base.

Les déclinaisons de nos tests sont les même que ceux décrit en section A.1.2, p. 155 :

Dans ce *cadre de test* commun, nous avons les paramètres suivants qui sont spécifiques à notre modèle *WMN* :

- *weight* : poids social ; influence du nœud mobile sur l'attractivité ;
- *Attractivity* : popularité d'un Point d'Attraction ; devient changeante en fonction de la fréquentation ;
- *Maximum Acceptable Factor* : fonction $f(t)$; limite de la croissance de l'attractivité ; valeur *constante* ;

A.2.2 Résultats

Introduction

Dans cette section, nous relatons les tests de simulation effectués selon les préroga-

tives de la section précédente. Les modèles *Manhattan* (*MN/MAN*) et *Weighted Manhattan* (*WMN*) sont rapprochés afin de mettre en évidence leurs différences. Le(s) rapprochement(s) présenté(s) au(x) chapitre(s) précédent(s) est(sont) également conservé(s) afin de le(s) mettre en perspective avec le présent rapprochement.

Configurations de test

Nous déclinons nos tests selon les critères suivants :

- modèle de mobilité ;
- paramètres du modèle ;
- densité du réseau mobile ;
- nombre de Points d’Attraction.

Nous rapprochons notre modèle *Weighted Manhattan* (*WMN*) du modèle de mobilité de base, à savoir *Manhattan* (*MN* ou *MAN*) pour mettre en perspective les apports de notre modèle.

Il va de soi que ce rapprochement nécessite un *cadre de test* commun. C’est la raison pour laquelle nous décidons de fixer certains paramètres de simulation. (cf. Table A.2 pour les paramètres de simulation). Il s’agit des paramètres suivants, nous utilisons ci-dessous leurs équivalents dans l’implémentation :

- *Safety Distance* : distance de sécurité routière ; valeur : 30 m, valeur moyenne en zone urbaine ;
- *Move Step* : unité de mouvement d’un nœud mobile ; valeur : 25 m, pour ne pas alourdir les temps de simulation ; diviseur du *Block Length* afin que les nœuds « passent » par les intersections ;
- *Block Length* : distance entre 2 lignes ou 2 colonnes de la grille ; valeur : 50 m, largeur moyenne d’un pâté de maisons ;
- $x_min, y_min, x_max, y_max$: aire de simulation ; valeur : 1 km^2 , superficie moyenne d’un quartier.

Pour les paramètres de simulation variables, les valeurs choisies nous apparaissent suffisantes dans un premier temps pour connaître l’influence des paramètres correspondants sur les modèles étudiés. Il s’agit de :

TABLE A.2 – Simulation

Simulateur	OPNET Modeler 16.0
Couche MAC	IEEE 802.11
Taux de transmission	250 m
Type de trafic	UDP
Destination	aléatoire
Caractéristiques physiques	Extended Rate PHY (802.11g)
Protocoles de routage	OLSR, AODV
Débit (Mbit/s)	11, 24, 54
Vitesse des nœuds (m/s)	2 (7 km/h, piéton, <i>pedestrian</i> , <i>ped</i>) 3 (11 km/h, bicyclette, <i>bicycle</i> , <i>bic</i>) 15 (52 km/h, véhicule, <i>car</i>)
Nombre de nœuds	20, 40
Aire de simulation	1000 m × 1000 m
Block length	50 m
Safety distance	30 m
Move step	25 m
Nombre de Points d'Attraction	4, 8
Attractivity	5
Modèle de mobilité	Manhattan (MN) Social Manhattan (SMN) Weighted Social Manhattan (WMN)

- protocole de routage ; valeurs : OLSR et AODV, les plus utilisés dans les réseaux sans fil et les plus représentatifs des grands ensembles que sont les protocoles réactifs et les protocoles proactifs, respectivement ;
 - débit ; valeurs : 11, 24 et 54 Mbit/s, un débit faible, un autre moyen, un débit plus important ;
 - vitesse *maximale* des nœuds ; valeurs : 2, 3 et 15 m/s soit 7 km/h pour les piétons, 11 km/h pour les bicyclettes et 52 km/h pour les véhicules urbains ;
 - nombre de nœuds ; valeurs : 20 nœuds, un groupe moyen (classe d'école...) et 40 nœuds, un groupe plus important (groupe de touristes, conférence...) ;
 - nombre de Points d'Attraction ; valeurs : 4, quartier périphérique et 8, centre ville ;
- Les différentes valeurs de nos paramètres de simulation sont résumés en Table A.2.

A.2. RÉSEAUX PONDÉRÉS

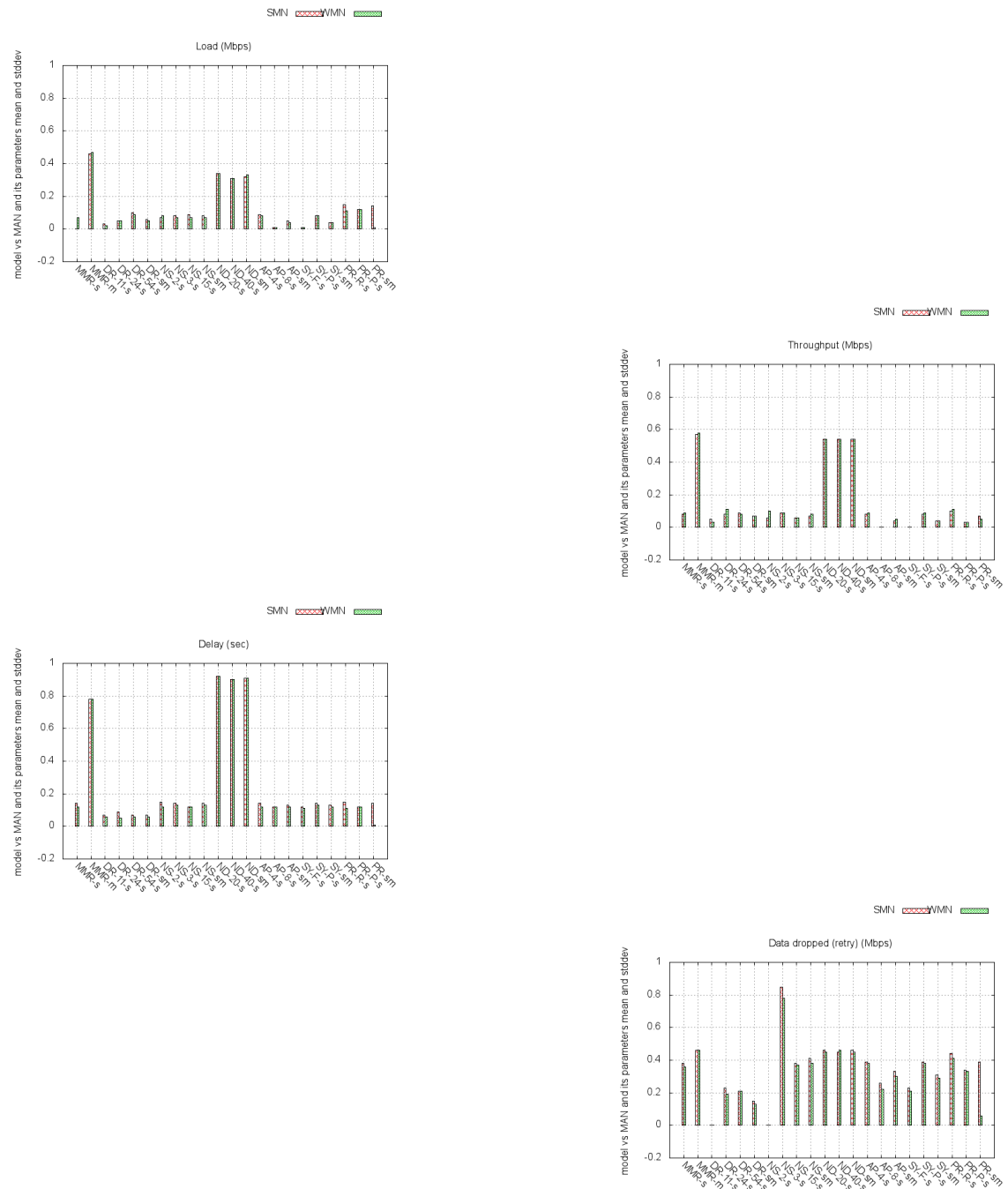


FIGURE A.5 – WMN - Variables statistiques réseau

La Fig. A.5 rassemble les graphiques suivants : Load, Throughput (Thru), Delay (Dlay), Data dropped (Retry Threshold Exceeded, DrpR).

La Table suivante en donne la légende.

MMR	"Model me mean" to "MAN me mean" Ratio with Model in {SMN,WMN,AM1,AM2,AM3} and me in {Load, Thru, Dlay, DrpR}
MMR-s	stddev of MMR of ALL scenarios
MMR-m	mean of MMR of ALL scenarios
DR-n	Data Rate of n Mbps with n in {11,24,54}
NS-n	Node Speed in range 0-n mps with n in {2,3,15}
ND-n	Network Density of n nodes with n in {20,40}
AP-n	Number of Attraction Points with n in {4,8}
SY-t	Sociability Type with t in {(F)ull,(P)artial}
PR-t	Protocol Type with t in {(R)eactive,(P)roactive}
X	MMR of All scenarios covered by X with X in {DR,NS,ND,AP,SY,PR}
X-n	with n in {relative parameter value, see above}
X-n-s	stddev of X-n
X-sm	mean of all X-n-s

Un graphique comporte 22 histogrammes. Notons tout d'abord *MMR*, le rapport ou *ratio* du modèle considéré sur le modèle Manhattan de la variable statistique considérée. Un histogramme donne les moyenne et écart des MMR, d'une part, de façon globale à tous les scénarios de test, puis, d'autre part, par paramètre, pour chacune de ses valeurs, pour les scénarios concernés. C'est cette dernière « moyenne d'écarts » qui dénote l' « influence » du paramètre sur l'ensemble de la mesure. Une précision concernant la Sociabilité s'impose : le caractère entièrement social ((*F*)*ull*) signifie que tous les nœuds sont respectueux du modèle de mobilité en vigueur ; tandis que la présence d'éléments en « rébellion » ou asociaux ((*P*)*artial*) est modélisée par environ 15 % des nœuds qui manquent totalement de respect à cet ordre préétabli.

Notons en premier lieu que :

La moyenne globale de Load lors du passage de Manhattan (MAN) à Weighted Manhattan (WMN) est de : -0,10. Cela signifie que Manhattan est surévalué en l'absence de prise en compte de l'aspect social.

A.2. RÉSEAUX PONDÉRÉS

La Table suivante résume l'« influence » des paramètres selon les prérogatives suivantes concernant l'écart global d'un paramètre (dénnoté par X-s, conférer la Table légende ci-dessus et l'axe des abscisses des graphiques) :

- inférieur à 0,2 : influence nulle ;
- entre 0,2 et 0,6 : influence faible ;
- entre 0,6 et 1 : influence moyenne ;
- supérieur à 1 influence forte.

	Load	Throughput	Delay	Data dropped
Debit	faible	forte	forte	forte
Vitesse	faible	forte	forte	forte
Densité	faible	forte	forte	forte
Nombre de Points d'Attraction	nulle	moyenne	forte	forte
Sociabilité	nulle	moyenne	forte	forte
Réactif / Proactif	nulle	forte	forte	forte

Comme c'était déjà le cas dans *SMN*, quand le nombre de Points d'Attraction diminue, la concentration des nœuds augmente. Cela a pour effet de réduire le délai de transmission de bout en bout. En effet, il n'y a pratiquement plus besoin de nœuds intermédiaires. En revanche, nombre de données sont perdues pour cause de bourrage du médium air (Data dropped). L'explication est la suivante : la bande de fréquence étant fixe, l'augmentation de la densité géographique des nœuds résulte en une augmentation du nombre de collisions.

Les variables statistiques réseau *Delay* et *Data dropped* qui représentent le délai entre

couches MAC des émetteur et récepteur, d'une part, et, d'autre part, les données abandonnées au niveau de la couche MAC à l'émission pour cause de bourrage du médium-air, ont rendu, pour la totalité des paramètres (débit, vitesse, densité, nombre de Points d'Attraction, Sociabilité et Type de protocole réactif ou proactif), des écarts considérables, très proches de ceux de *SMN* mais sans les atteindre cependant. Pour les autres variables statistiques, Load et Throughput, la tendance de leurs influences a augmenté, par rapport à *SMN*, en devenant « moyenne » voire « forte », et presque jamais « nulle ». Cela nous permet de conclure que la prise en compte de l'aspect social évolutif des Points d'Attraction, c'est-à-dire, le fait de faire croître leurs attractivités en fonction des visites, doit faire l'objet d'attention dans la conception des protocoles de communications dans les réseaux sans fil, au même titre que *SMN*.

A.3 Les systèmes de colonies de fourmis

A.3.1 Résultats

Introduction

Dans cette section, nous relatons les tests de simulation effectués selon les prérogatives de la section précédente. Les modèles *Manhattan (MN/MAN)* et *Ant Manhattan (AMN)* sont rapprochés afin de mettre en évidence leurs différences. Le(s) rapprochement(s) présenté(s) au(x) chapitre(s) précédent(s) est(sont) également conservé(s) afin de le(s) mettre en perspective avec le présent rapprochement.

Configurations de test

Pour récapituler, les nouveaux paramètres de notre modèle sont :

- *Pheromone Decay factor* : α ;
- *Pheromone Distance Exponent* : β .

Nous déclinons nos tests selon les critères suivants :

- modèle de mobilité ;
- paramètres du modèle ;

- densité du réseau mobile ;
- nombre de Points d’Attraction.

Nous rapprochons notre modèle fourmis (*AMN*) du modèle Manhattan de base (*MN* ou *MAN*) pour mettre en perspective les apports de notre modèle.

Il va de soi que ce rapprochement nécessite un *cadre de test* commun. C’est la raison pour laquelle nous décidons de fixer certains paramètres de simulation. Il s’agit des paramètres suivants, nous utilisons ci-dessous leurs équivalents dans l’implémentation :

- *Safety Distance* : distance de sécurité routière ; valeur : 30 m, valeur moyenne en zone urbaine ;
- *Move Step* : unité de mouvement d’un nœud mobile ; valeur : 25 m, pour ne pas alourdir les temps de simulation ; diviseur du *Block Length* afin que les nœuds « passent » par les intersections ;
- *Block Length* : distance entre 2 lignes ou 2 colonnes de la grille ; valeur : 50 m, largeur moyenne d’un pâté de maisons ;
- $x_min, y_min, x_max, y_max$: aire de simulation ; valeur : 1 km^2 , superficie moyenne d’un quartier.

Pour les paramètres de simulation variables, les valeurs choisies nous apparaissent suffisantes dans un premier temps pour connaître l’influence des paramètres correspondants sur les modèles étudiés. Il s’agit de :

- protocole de routage ; valeurs : OLSR et AODV, les plus utilisés dans les réseaux sans fil et les plus représentatifs des grands ensembles que sont les protocoles réactifs et les protocoles proactifs, respectivement ;
- débit ; valeurs : 11, 24 et 54 Mbit/s, un débit faible, un autre moyen, un débit plus important ;
- vitesse *maximale* des nœuds ; valeurs : 2, 3 et 15 m/s soit 7 km/h pour les piétons, 11 km/h pour les bicyclettes et 52 km/h pour les véhicules urbains ;
- nombre de nœuds ; valeurs : 20 nœuds, un groupe moyen (classe d’école...) et 40 nœuds, un groupe plus important (groupe de touristes, conférence...) ;
- nombre de Points d’Attraction ; valeurs : 4, quartier périphérique et 8, centre ville ;

Les différentes valeurs de nos paramètres de simulation sont résumés en Table A.3.

TABLE A.3 – Simulation

Simulateur	OPNET Modeler 16.0
Couche MAC	IEEE 802.11
Taux de transmission	250 m
Type de trafic	UDP
Destination	aléatoire
Caractéristiques physiques	Extended Rate PHY (802.11g)
Protocoles de routage	OLSR, AODV
Débit (Mbit/s)	11, 24, 54
Vitesse des nœuds (m/s)	2 (7 km/h, piéton, <i>pedestrian</i> , <i>ped</i>) 3 (11 km/h, bicyclette, <i>bicycle</i> , <i>bic</i>) 15 (52 km/h, véhicule, <i>car</i>)
Nombre de nœuds	20, 40
Aire de simulation	1000 m × 1000 m
Block length	50 m
Safety distance	30 m
Move step	25 m
Nombre de Points d'Attraction	4, 8
Attractivity	5
Modèle de mobilité	Manhattan (MN) Social Manhattan (SMN) Weighted Social Manhattan (WMN) Ant Manhattan (AMN)
(α, β)	AMN1 : 0.005 : norm. decay, 1.5 : norm. dist. AMN2 : 0.002 : weak decay, 2 : heavy dist. AMN3 : 0.01 : heavy decay , 0.5 : weak dist.

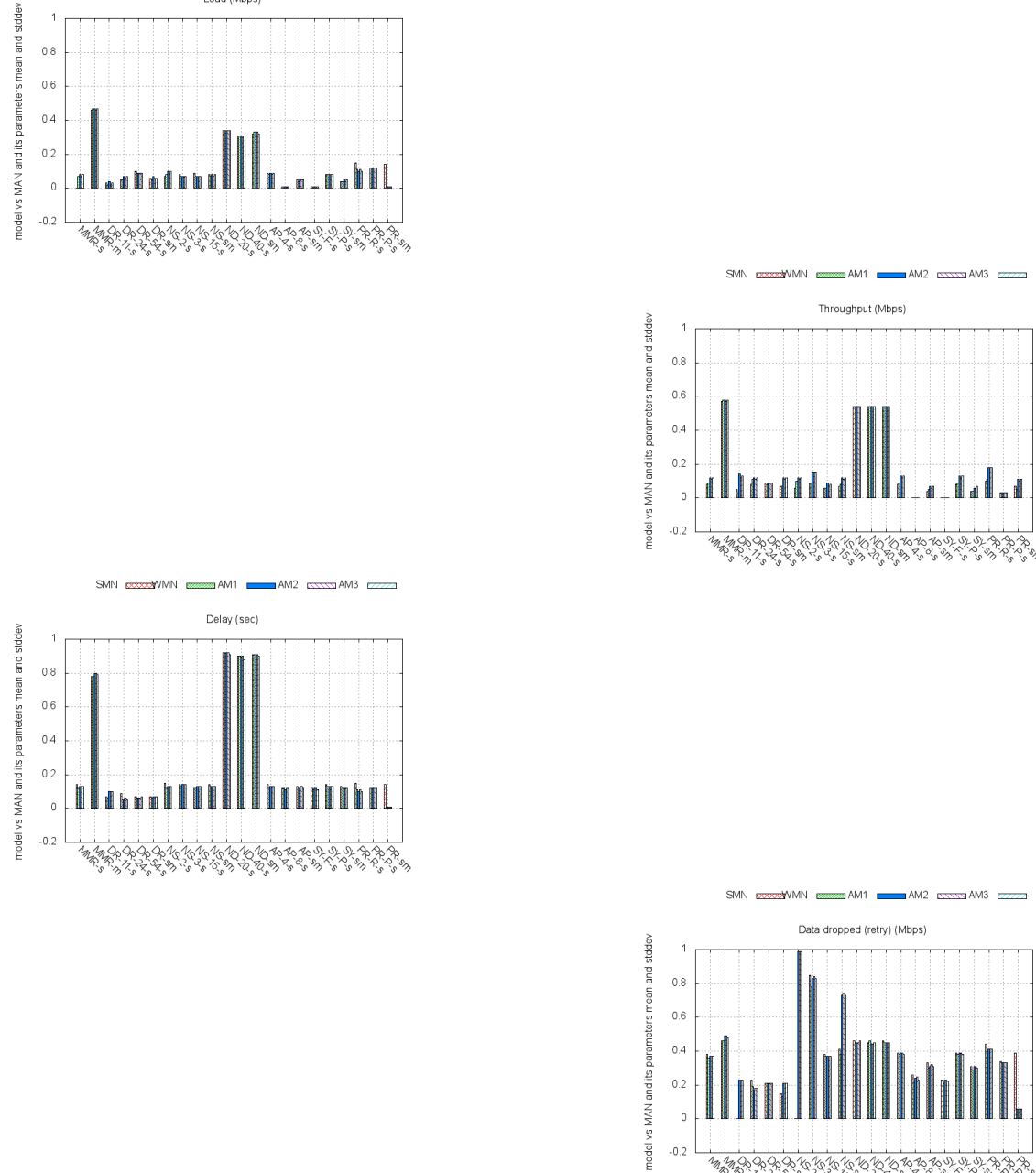


FIGURE A.6 – AMN - Variables statistiques réseau

La Fig. A.6 rassemble les graphiques suivants : Load, Throughput (Thru), Delay (Dlay), Data dropped (Retry Threshold Exceeded, DrpR).

La Table suivante en donne la légende.

MMR	"Model me mean" to "MAN me mean" Ratio with Model in {SMN,WMN,AM1,AM2,AM3} and me in {Load, Thru, Dlay, DrpR}
MMR-s	stddev of MMR of ALL scenarios
MMR-m	mean of MMR of ALL scenarios
DR-n	Data Rate of n Mbps with n in {11,24,54}
NS-n	Node Speed in range 0-n mps with n in {2,3,15}
ND-n	Network Density of n nodes with n in {20,40}
AP-n	Number of Attraction Points with n in {4,8}
SY-t	Sociability Type with t in {(F)ull,(P)artial}
PR-t	Protocol Type with t in {(R)eactive,(P)roactive}
X	MMR of All scenarios covered by X with X in {DR,NS,ND,AP,SY,PR}
X-n	with n in {relative parameter value, see above}
X-n-s	stddev of X-n
X-sm	mean of all X-n-s

Un graphique comporte 22 histogrammes. Notons tout d'abord *MMR*, le rapport ou *ratio* du modèle considéré sur le modèle Manhattan de la variable statistique considérée. Un histogramme donne les moyenne et écart des MMR, d'une part, de façon globale à tous les scénarios de test, puis, d'autre part, par paramètre, pour chacune de ses valeurs, pour les scénarios concernés. C'est cette dernière « moyenne d'écarts » qui dénote l' « influence » du paramètre sur l'ensemble de la mesure. Une précision concernant la Sociabilité s'impose : le caractère entièrement social ((*F*)*ull*) signifie que tous les nœuds sont respectueux du modèle de mobilité en vigueur ; tandis que la présence d'éléments en « rébellion » ou asociaux ((*P*)*artial*) est modélisée par environ 15 % des nœuds qui manquent totalement de respect à cet ordre préétabli.

Notons en premier lieu que :

La moyenne globale de Load lors du passage de Manhattan (MAN) à Ant Manhattan (AMN) est de : -0,10. Cela signifie que Manhattan est surévalué en l'absence de prise en compte de l'aspect social.

A.3. LES SYSTÈMES DE COLONIES DE FOURMIS

La Table suivante résume l'« influence » des paramètres selon les prérogatives suivantes concernant l'écart global d'un paramètre (dénnoté par X-s, conférer la Table légende ci-dessus et l'axe des abscisses des graphiques) :

- inférieur à 0,2 : influence nulle ;
- entre 0,2 et 0,6 : influence faible ;
- entre 0,6 et 1 : influence moyenne ;
- supérieur à 1 influence forte.

	Load	Throughput	Delay	Data dropped
Debit	faible	forte	forte	forte
Vitesse	faible	forte	forte	forte
Densité	faible	forte	forte	forte
Nombre de Points d'Attraction	nulle	moyenne	forte	forte
Sociabilité	nulle	moyenne	forte	forte
Réactif / Proactif	faible	forte	forte	forte

À nouveau, quand le nombre de Points d'Attraction diminue, la concentration des nœuds augmente. Cela a pour effet de réduire le délai de transmission de bout en bout. En effet, il n'y a pratiquement plus besoin de nœuds intermédiaires. En revanche, nombre de données sont perdues pour cause de bourrage du médium air (Data dropped). L'explication est la suivante : la bande de fréquence étant fixe, l'augmentation de la densité géographique des nœuds résulte en une augmentation du nombre de collisions.

Les variables statistiques réseau *Delay* et *Data dropped* qui représentent le délai entre

couches MAC des émetteur et récepteur, d'une part, et, d'autre part, les données abandonnées au niveau de la couche MAC à l'émission pour cause de bourrage du médium-air, ont rendu, pour la totalité des paramètres (débit, vitesse, densité, nombre de Points d'Attraction, Sociabilité et Type de protocole réactif ou proactif), des écarts considérables, compris entre ceux de *SMN* et ceux de *WMN*. Pour les autres variables statistiques, Load et Throughput, la tendance de leurs influences a augmenté, par rapport à *SMN*, comme pour *WMN*, en devenant « moyenne » voire « forte », et presque jamais « nulle ». Notons toutefois, le passage d'une influence « forte » à une influence « plus forte » du choix du Type de protocole de routage (Réactif/Proactif) pour la variable statistique réseau *Throughput*. Cela nous permet de conclure que la prise en compte de l'aspect social évolutif des Points d'Attraction, c'est-à-dire, le fait de faire croître leurs attractivités en fonction des visites MAIS AUSSI, et c'est là la spécificité d'*AMN*, le fait de la faire décroître en fonction de la raréfaction des visites, doit faire l'objet d'attention dans la conception des protocoles de communications dans les réseaux sans fil, au même titre qu'*SMN* et *WMN*.

Annexe B

Choix relatifs à la simulation et implémentation de nos modèles

B.1 Introduction

Afin d'évaluer nos modèles, nous devons choisir :

1. un cadre de travail parmi ceux existant en matière de mobilité ;
2. un outil de simulation.

Le cadre de travail que nous choisissons est celui du modèle de mobilité *Manhattan* car de très loin le plus répandu dans la littérature.

L'outil de simulation que nous choisissons est *Opnet Modeler* pour les raisons exposées à la section suivante qui présente cet outil.

Ensuite, nous relatons notre étude de l'existant en matière de modèle de mobilité, i.e. ce qui existe dans la version d'*Opnet Modeler* destinée aux académies.

Puis, nos premiers développements du modèle de mobilité *Manhattan* sont exposés.

Vient ensuite le problème des intersections manquées qui représente le verrou technique majeur de cette thèse. La cause de ce problème puis la solution choisie pour lever ce verrou qui littéralement débloque l'ensemble de notre travail, font l'objet de sections séparées.

Les sections suivantes sont ensuite consacrées au mode de développement d'un objet nouveau dans la palette d'objets de l'outil. La découverte de ce mode de développement représente en effet un investissement en temps considérable. Celui-ci trouve toute sa jus-

tification dans la simplification drastique que ces nouveaux objets visuels entraînent dans l'implémentation de nos modèles qui comprennent parfois plus d'une soixantaine d'instance de ses nouveaux objets !

L'implémentation proprement dite de nos modèles fera l'objet d'une section dédiée dans les chapitres correspondants.

Enfin et surtout nos résultats de simulation sont donnés en dernière section desdits chapitres. Ceux-ci sont le but de l'ensemble du processus de développement : démontrer l'intérêt de nos modèles !

L'implémentation des modèles des chapitres suivants s'appuie, répétons-le, sur les sections qui vont suivre immédiatement. Le lecteur ne s'étonnera donc pas de la longueur de l'exposé compte tenu du fait qu'il sert de base à l'ensemble de l'implémentation des outils de simulations utilisés tout au long de ce travail de thèse.

B.2 Présentation d'*Opnet Modeler*

Au Centre de recherche en informatique et communication, au laboratoire *CEDRIC* du Conservatoire National des Arts et Métiers de Paris, *CNAM de Paris*, nous disposons de 14 licences d'utilisation du produit commercial *Opnet Modeler* (Fig. B.1). La version dont nous disposons pour ce travail de thèse est donnée en Fig. B.2. Les mentions légales concernant les académies s'affichent au lancement de l'outil et sont reproduites en Fig. B.3.

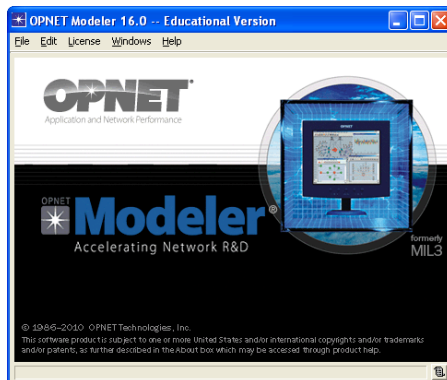


FIGURE B.1 – OPNET Modeler

Il s'agit d'un outil de simulation qui, par rapport à *Network Simulator 2* qui est le

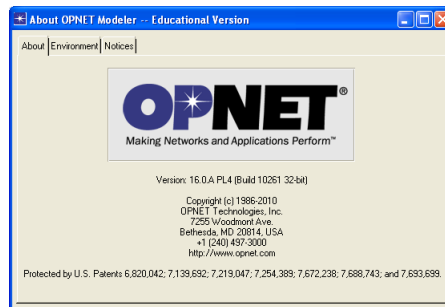


FIGURE B.2 – Notre version d'OPNET Modeler

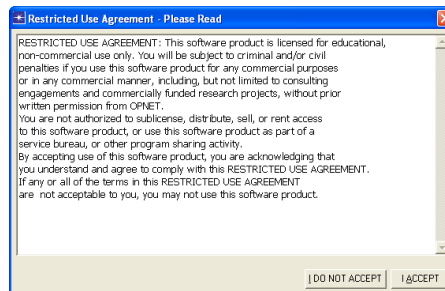


FIGURE B.3 – Mentions légales OPNET Modeler à destination des Académies

simulateur le plus utilisé dans les laboratoires de recherche, dispose d'au minimum 2 points forts :

- une interface homme-machine graphique évoluée orientée objet ;
- des objets ou nœuds de simulation qui suivent de près les besoins du marché.

L'interface homme-machine graphique évoluée orientée objet permet à l'utilisateur de décrire un scénario de simulation en choisissant les éléments du réseau simulé, dans une palette d'objets, Fig. B.4A. Ceux-ci sont déposés sur un espace représentant le lieu sur lequel s'étend le réseau simulé. Ensuite, chacun de ses objets instanciés peut faire l'objet d'un paramétrage spécifique via un éditeur d'attributs, Fig. B.4B.

Ces objets de simulation suivent de près les besoins du marché, comme en témoigne la présence de :

- La norme 802.11g implémentée au milieu de la première décennie 2000, en même temps que son adoption dans l'entreprise, Fig. B.4C ;
- La norme 802.11n MIMO implémentée fin de cette même décennie via la possibilité

B.3. ÉTUDE DE L'EXISTANT

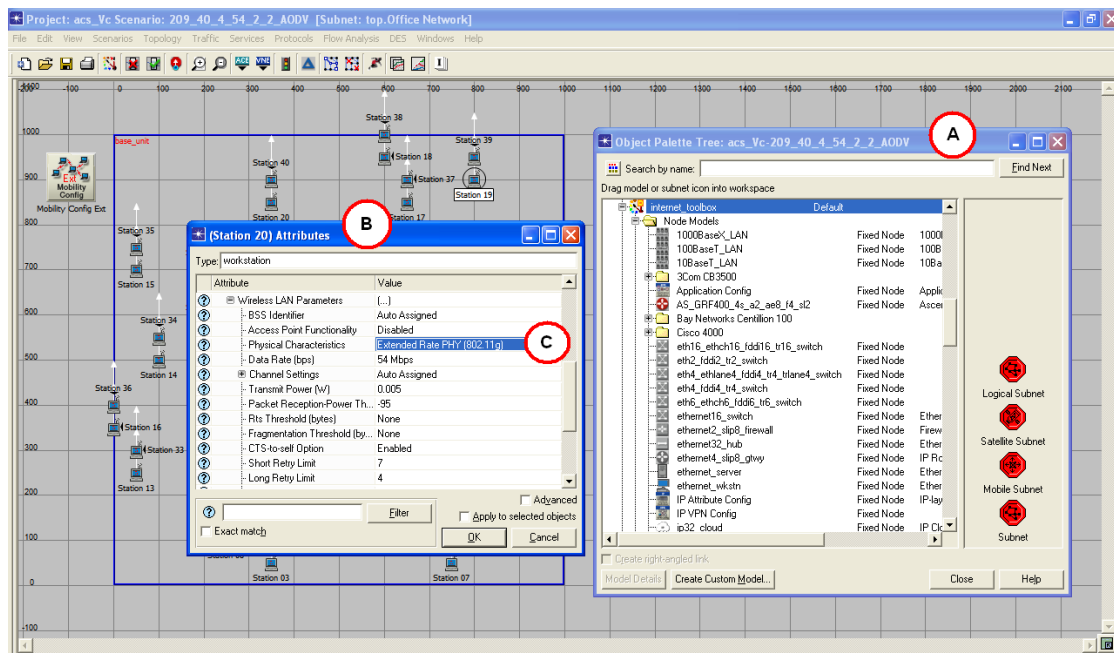


FIGURE B.4 – L'Interface Homme-Machine (IHM) d'OPNET Modeler

de définir le nombre d'antennes en émission comme en réception dont la simulation consiste en un accroissement sensible du rapport signal sur bruit (SNR) ;

- L'évolution à long terme ou LTE qui concerne tant la Super 3G que l'actuelle 4G en cours de normalisation.

En conséquence, *Opnet Modeler* est à notre connaissance le plus à jour à ce niveau.

Suite aux différentes présentations de la société *Opnet France* dans les locaux du *CNAM*, nous avons investi de nombreux mois de notre recherche à acquérir une solide expérience du *Modeler* que nous maîtrisons pour les aspects *802.11 WiFi* dont nous dispensons depuis plusieurs années des cours de formation dans le cadre des cursus Ingénieur et Mastère de la branche Réseaux du *CNAM Paris*.

B.3 Étude de l'existant

Sauf erreur ou omission de notre part, depuis le début de nos travaux de recherche dans le cadre de la présente thèse et jusqu'à la rédaction de celle-ci, l'unique modèle de mobilité présent dans la version mise à disposition par la société *Opnet* dans son *Modeler*

à destination des académies est : *Random Waypoint*.

Cette section se propose d'étudier cet unique « existant ».

Un projet d'exemple va nous y aider :

1. lancer « Opnet Modeler » ;
2. cliquer « File | Open... » ;
3. constater l'affichage de la boîte de dialogue standard « Open » ;
4. sélectionner le type de fichier « Project File (*.prj) » ;
5. se déplacer jusqu'au dossier suivant :

D:\Program Files\OPNET\16.0.A\models\std\example_networks\

où D représente la lettre du lecteur sur lequel est installé *Opnet Modeler* ;

6. ouvrir le dossier « Random_Mobility.project » ;
7. double-cliquer le fichier « Random_Mobility » qui a pour extension « .prj » ;
8. constater l'ouverture du projet d'exemple, Fig. B.5 ;



9. constater la présence de l'objet Mobility Config : c'est lui qui renferme le paramétrage du (ou des) modèle(s) de mobilité utilisé(s) dans le scénario courant nommé « Baseline » ;
10. cliquer droit sur cet objet et dans le menu contextuel qui s'affiche sélectionner « Edit Attributes » ;
11. constater l'affichage de l'éditeur d'attributs de l'objet, voir Fig. B.6 ;

Décrivons les attributs de l'objet « Mobility Config » qui apparaissent dans la Fig. B.6, du premier au dernier :

- name = Mobility Profile Definition : nom donné à l'instance ;
- Mobility Modeling Status = Enabled : les modèles de mobilité de l'instance sont actifs ;
- Random Mobility Profiles : ensemble de modèles de mobilité :
 - Number of Rows = 1 : nombre de modèles de mobilité ;

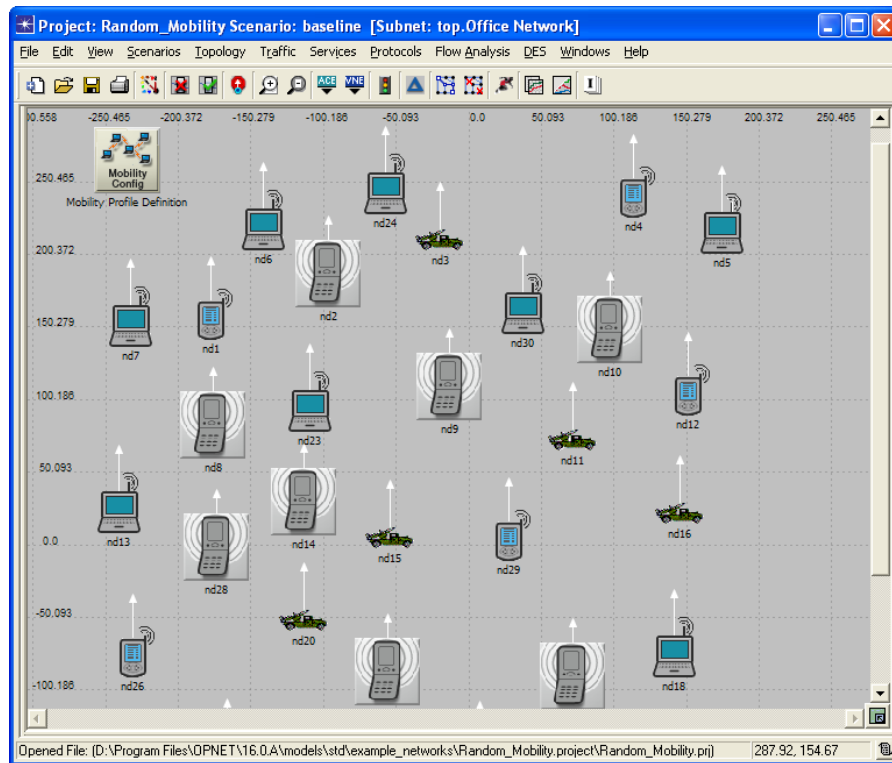


FIGURE B.5 – Projet Exemple Random_mobility d'Opnet Modeler

- Wanderer : nom du modèle de mobilité n° 1
- Profile Name = Wanderer : idem
- Mobility Model = Random Waypoint : type du modèle de mobilité
- Random Waypoint Parameters :

Attribut	Fonction
Mobility Domain Model	aire de déplacement : déf. visuelle
x_min, y_min, x_max, y_max	aire de déplacement : déf. non visuelle
Speed	vitesse des nœuds en m/s
Pause Time	temps de pause entre 2 déplacements
Start Time	délai en s entre début simu et déplac.
Stop Time	durée de l'ensemble des déplacements
Animation Update Frequency	période animation
Record Trajectory	enregistrement trajectoires des nœuds

Le seul modèle de mobilité existant est exclusivement « Random Waypoint » comme en témoigne la liste de choix ouverte qui ne contient que cet élément.

L'Éditeur d'attributs comporte une case à cocher intitulée « Advanced ». Celle-ci en la

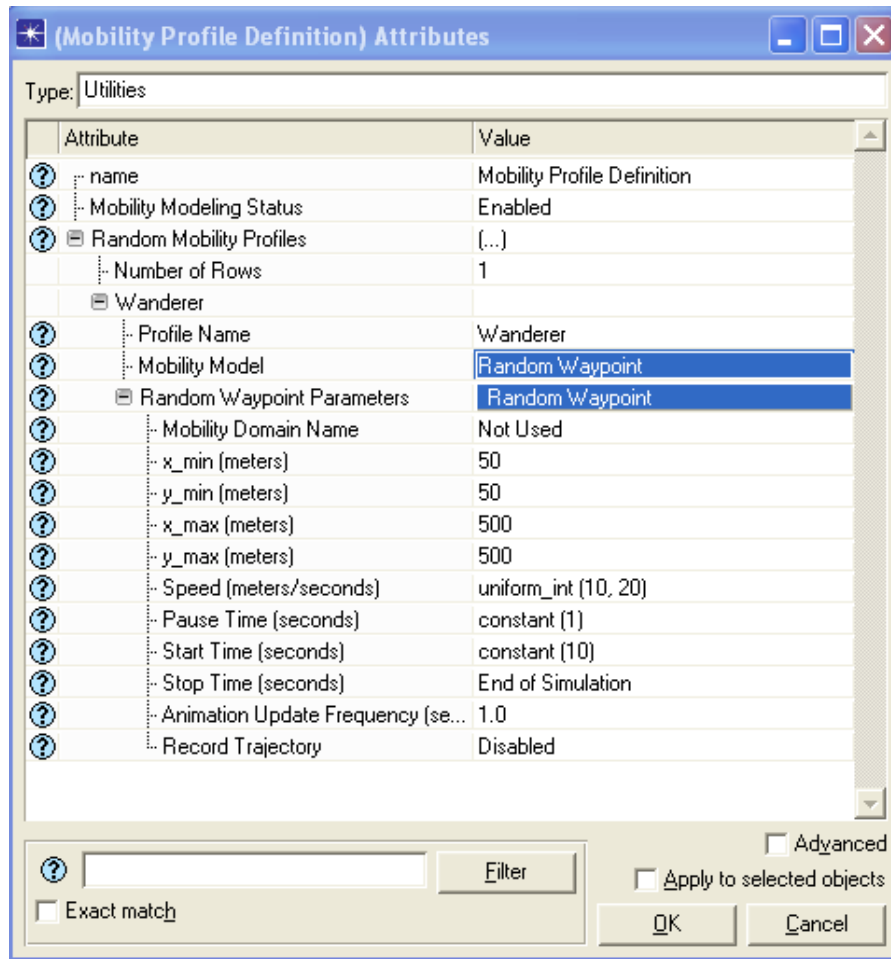


FIGURE B.6 – Editeur d'attributs de l'objet Mobility Config

cochant permet d'afficher des attributs supplémentaires, voir Fig. B.7.

Notamment, l'attribut « model » nous permet d'accéder aux organes internes de l'objet. En cliquant sur ce dernier, nous recevons la mise en garde de la Fig. B.7, qui stipule qu'il est possible de créer un modèle de sauvegarde du nœud considéré via *Device Creator* afin de pouvoir importer ses données en cas de changement du modèle d'objet. Recherches faites dans l'aide en ligne, les seuls objets concernés par cette fonctionnalité sont donnés ci-dessous :

Groupe	Objet
Suite page suivante	

Groupe Suite ...	Objet
Bridge	Ethernet (10BaseT, 100BaseT, 1000BaseX) FDDI Token Ring
Cloud	ATM Frame Relay SLIP Multiprotocol
Fibre	Channel Devices Hub Switch Endnode Multihomed Endnode
Firewall	ATM Ethernet (10BaseT, 100BaseT, 1000BaseX), Ethernet LANE, and EtherChannel FDDI Frame Relay SLIP Token Ring and TR LANE
Hub	Ethernet (10BaseT, 100BaseT, 1000BaseX) FDDI Fibre Channel Token Ring
LAN Model	Shared Switched
Mainframes	ATM Ethernet (10BaseT, 100BaseT, 1000BaseX), Ethernet LANE, and EtherChannel
Suite page suivante	

Groupe Suite ...	Objet FDDI Frame Relay Fibre Channel SLIP Token Ring and TR LANE Wireless LAN
Multihomed	Host Client Server
Multi-Service Switch	Circuit Switched Ethernet (10BaseT, 100BaseT, 1000BaseX) and EtherChannel FDDI Frame Relay SLIP Token Ring Voice over ATM
Router	ATM Ethernet (10BaseT, 100BaseT, 1000BaseX), Ethernet LANE, and EtherChannel FDDI Frame Relay SLIP Token Ring and TR LANE Wireless LAN
Switch	ATM Ethernet (10BaseT, 100BaseT, 1000BaseX) and EtherChannel FDDI Fibre Channel Frame Relay LANE
Suite page suivante	

Groupe Suite ...	Objet Multiprotocol Switch SSP Token Ring UMTS RNC
Vendor Device	3Com Avici Systems Bay Networks Cabletron Cisco eXtreme Net- works Fore Systems Foundry Net- works Hewlett-Packard Juniper Networks Lucent (Ascend) Newbridge NEC Nortel Networks
C'est fini	

Ces objets concernés par *Device Creator* ne comprennent ni « Mobility Config » ni « manet_station » qui sont les 2 objets sur lesquels nous basons nos développements.

Du coup, nous répondons *Yes* la mise en garde de la Fig. B.8 pour signifier au système que nous avons bien compris le message !

La liste de choix pour l'attribut « model est affichée comme suit :

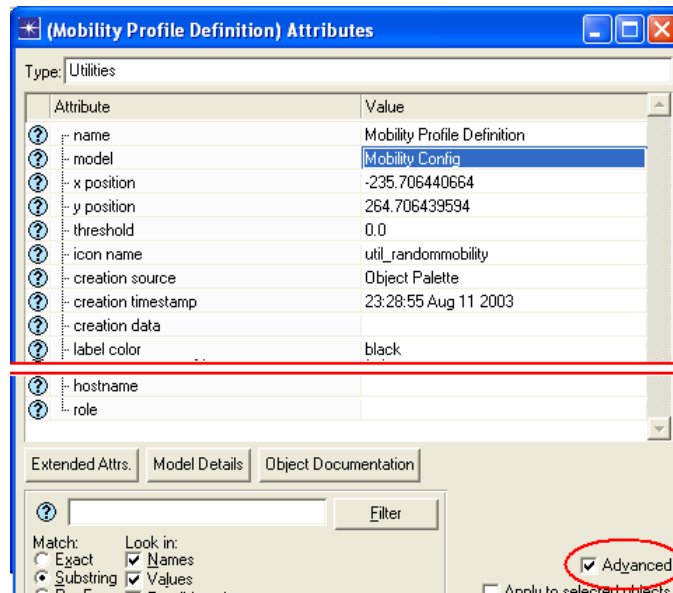
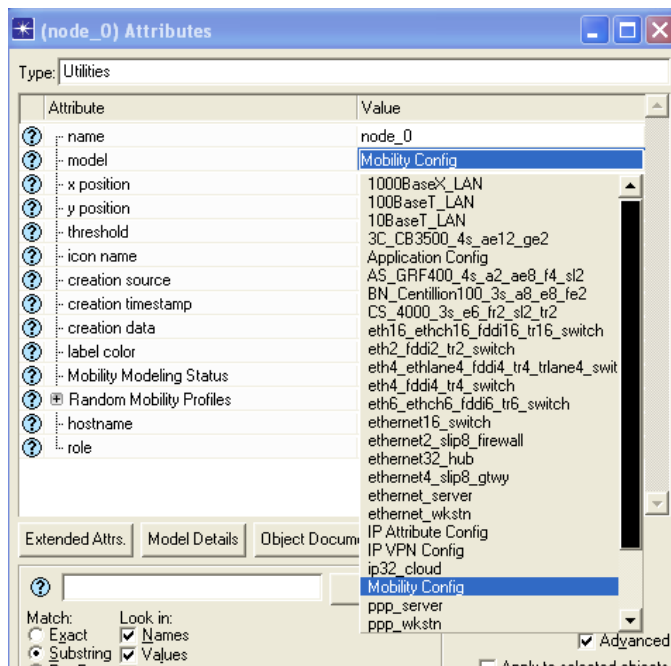


FIGURE B.7 – Éditeur d'attributs avancé de l'objet Mobility Config



Mais, nous en tenons pas compte car notre but est d'étudier l'existant et non pas de le modifier. Nous cliquons donc à nouveau la valeur actuelle de l'attribut « model » à savoir Mobility Config et ouvrons par là même, la boîte de dialogue ci-dessous :

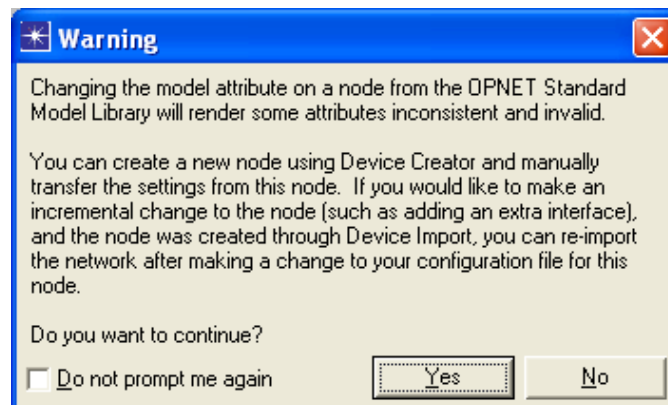
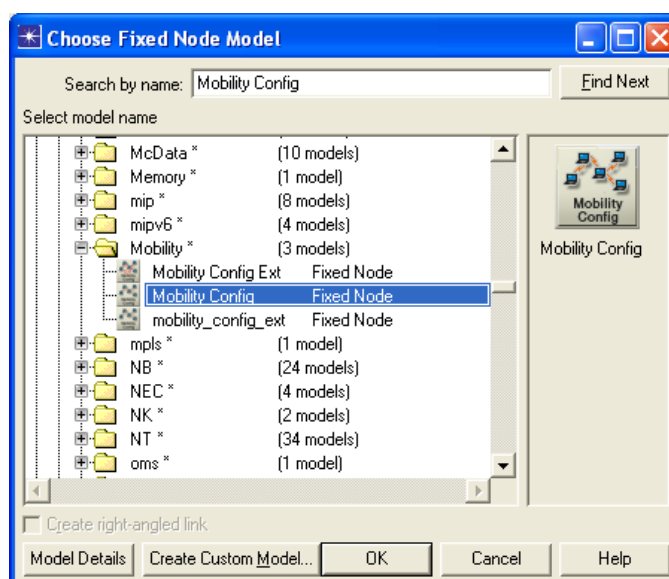
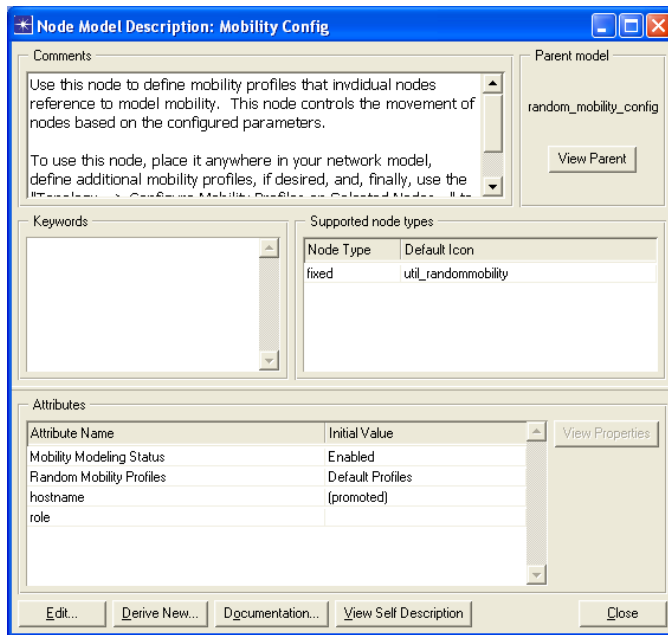


FIGURE B.8 – Mise en garde concernant les modifications de modèle d'objet *Opnet*

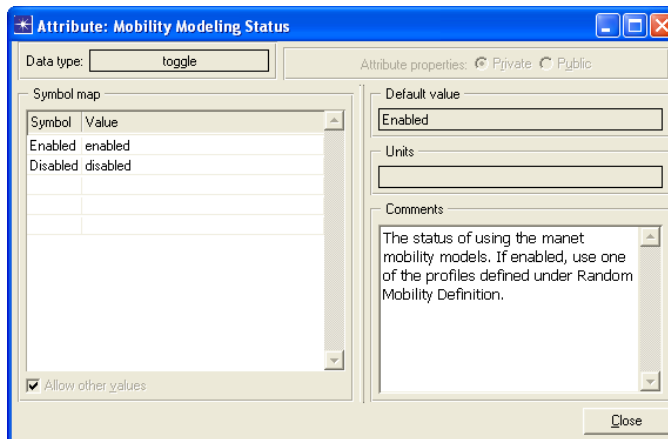


Nous cliquons droit sur l'icône en marge droite et faisons apparaître ceci :



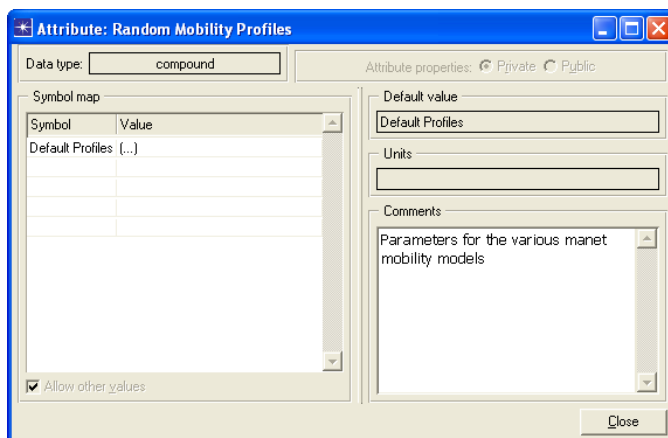
Il s'agit de la description du modèle de nœud de l'objet « Mobility Config ». Un volet en haut à gauche donne un court commentaire sur la fonction du nœud. En haut à droite, nous trouvons le modèle parent qui est « random_mobility_config ». Au centre sur l'axe horizontal, nous trouvons un volet « Keywords » vide et un autre « Supported node types » en forme de tableau qui pour chaque type de nœud (fixed, mobile, satellite) donne l'icône associée. En l'occurrence, seule la ligne « fixed » est renseignée avec la valeur « util_randommobility ». Enfin en bas, le volet « Attributes » est un tableau qui liste les attributs et leurs valeurs initiales. Nous avons déjà rencontré plus haut, « Mobility Modeling Status » et « Random Mobility Profiles », qui était dans l'éditeur d'attributs simple. Nous y trouvons également « hostname » qui était présent dans l'éditeur d'attributs avancé. Nous constatons que la valeur initiale de cet attribut est « (promoted) », entre parenthèse donc : c'est sans doute ce qui fait qu'il ne s'affiche que dans l'éditeur d'attributs avancé. À droite de ce tableau, nous trouvons un bouton inactif (grisé) nommé « View Attribute ».

Nous sélectionnons l'attribut « Mobility Modeling Status », puis cliquons sur ce bouton « View Attribute » qui ouvre la fenêtre suivante :



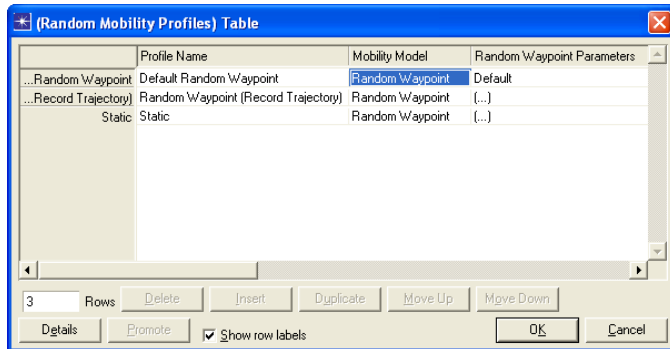
Elle nous donne : le type de l'attribut qui est « Toggle » (interrupteur à bascule) ; les valeurs possibles de cet attribut qui sont données dans un tableau à 2 colonnes « Symbol » (étiquette) et « Value » et qui sont l'étiquette « Enabled » de valeur « enabled » qui signifie que les modèles de mobilité définis dans l'objet « Mobility Config » sont actifs, et l'étiquette « Disabled » de valeur « disabled » qui signifie que ces modèles sont inactifs ; sa valeur par défaut qui est « Enabled » ainsi qu'un commentaire succinct qui nous dit que cette bascule contrôle l'utilisation des modèles (ou profiles) de mobilité par les stations du réseau. Nous cliquons « OK » afin de revenir à la fenêtre précédente.

Nous sélectionnons maintenant l'attribut « Random Mobility Profiles » et cliquons le bouton « View Properties » qui ouvre la fenêtre suivante :



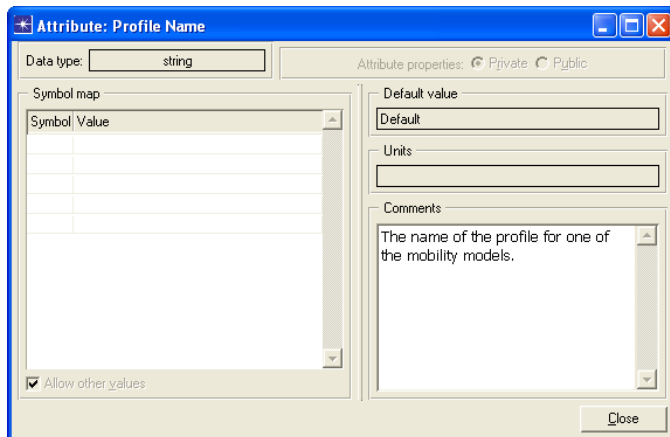
Elle nous donne : le type de l'attribut qui est « Compound » (composé) ; les valeurs possibles de cet attribut qui sont une à savoir l'étiquette « Default Profiles » et sa valeur « (...) » ; un commentaire qui nous informe du fait que cet attribut renferme l'ensemble

des modèles (profiles) de mobilité. Nous cliquons la valeur « (...) » qui ouvre la fenêtre suivante :



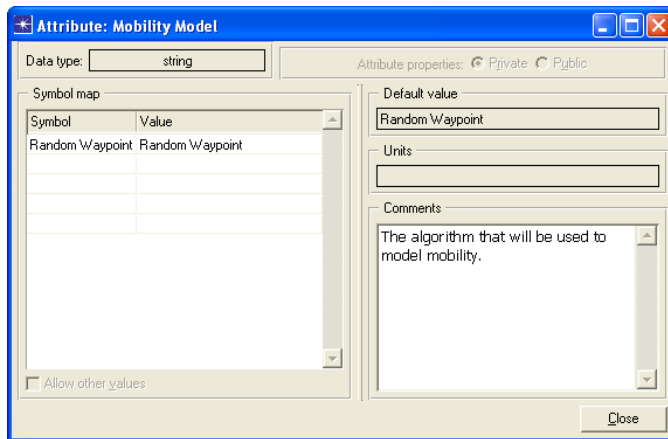
Elle est un tableau de 3 colonnes qui sont « Profile Name », « Mobility Model » et « Random Waypoint Parameters » que nous avons déjà rencontrés Fig. B.6, p. 183, et 3 lignes qui sont les 3 modèles (profiles) de mobilité définis par défaut : « Default Random Waypoint », « Random Waypoint (Record Trajectory) » et « Static ».

Nous sélectionnons « Default Random Waypoint » et cliquons le bouton « Details » qui ouvre la fenêtre suivante :



Elle nous donne le type « string » (chaîne de caractères) ; la valeur par défaut « Default » ainsi qu'un commentaire qui nous apprend que l'attribut « Profile Name » n'est autre que le nom du modèle (profile) de mobilité considéré. Nous cliquons « OK » pour revenir à la fenêtre précédente.

Nous sélectionnons « Random Waypoint » et cliquons le bouton « Details » qui ouvre la fenêtre suivante :



Elle nous donne : le type « string » ; l'étiquette « Random Waypoint » de valeur « Random Waypoint » ; la valeur par défaut « Random Waypoint » ainsi qu'un commentaire nous indiquant qu'il s'agit de l'algorithme qui implémente le modèle de mobilité. Nous cliquons « OK » pour revenir à la fenêtre précédente.

Nous sélectionnons « Random Waypoint Parameters » et cliquons le bouton « Details » qui ouvre la fenêtre suivante :

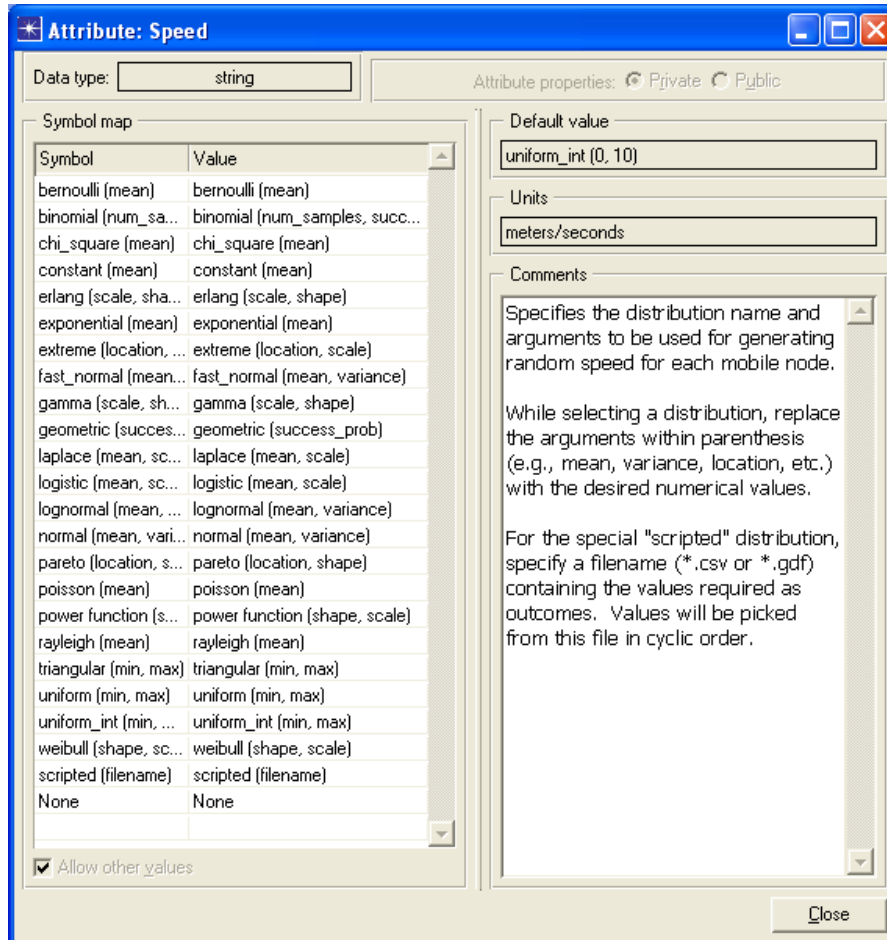
Attribute	Value
Mobility Domain Name	Not Used
x_min (meters)	0.0
y_min (meters)	0.0
x_max (meters)	500
y_max (meters)	500
Speed (meters/seconds)	uniform_int (0, 10)
Pause Time (seconds)	constant (100)
Start Time (seconds)	constant (10)
Stop Time (seconds)	End of Simulation
Animation Update Frequency (seconds)	1.0
Record Trajectory	Disabled

Elle nous donne un tableau à 2 colonnes « Attribute » et « Value » qui nous donne pour chaque paramètre du modèle de mobilité déjà rencontré Fig. B.6, p. 183, sa valeur.

Nous passons rapidement sur les premiers attributs de la liste qui sont soit des « string » (chaîne de caractères), soit des « double » (valeur numérique en virgule flottante).

Nous sélectionnons maintenant « Speed (meters/seconds) » et cliquons le bouton « De-

tails » qui ouvre la fenêtre suivante :



Elle illustre combien *Opnet Modeler* est complet d'un point de vue probabiliste : pas moins de 23 choix différents ! La valeur par défaut est « uniform_int(0, 10) » qui signifie « distribution uniforme en 0 et 10 m/s ». On remarque l'unité donnée dans la zone « Units ». Et bien sûr, le commentaire qui explique que l'on peut choisir dans la liste la distribution qui convient ainsi que ses arguments qu'il faut prendre soin de remplacer par les valeurs réelles. À noter la présence du type « scripted » où les valeurs de la distribution peuvent être lues dans un fichier externe du type *.csv, par exemple.

Les attributs « Pause (seconds) » et « Time (seconds) » dispose d'un choix identique et leurs valeurs par défaut sont respectivement « constant(100) » (pour la valeur constante 100) et « constant(10) ».

Nous sélectionnons maintenant « Stop Time (seconds) » et cliquons le bouton « Details »

qui ouvre la fenêtre suivante :

Symbol	Value
End of Simulation	-1.0

Elle nous donne : le type « double » ; l'étiquette « End of Simulation » de valeur « -1 » ; la valeur par défaut « End of Simulation » ainsi qu'un commentaire nous indiquant qu'il s'agit du temps de terminaison pour tous les nœuds qui implémentent le modèle de mobilité. Nous cliquons « OK » pour revenir à la fenêtre précédente.

Nous cliquons « OK » pour fermer la fenêtre « (Random Waypoint Parameters) Table » et revenir à la fenêtre « (Random Mobility Profiles) Table ».

Nous cliquons sur la deuxième ligne sur « (...) » afin de constater que la seule différence entre « Default Random Waypoint » et « Random Waypoint (Record Trajectory) » est justement le fait que l'attribut « Record Trajectory » est à « Enabled » dans ce dernier.

Quant au troisième modèle de mobilité par défaut, « Static », il a son attribut « Start Time (secondes) » à la valeur « None » qui signifie que la mobilité ne démarre jamais, résultant en un état statique des nœuds l'implémentant.

Nous cliquons « OK » pour fermer la fenêtre « (Random Mobility Profiles) Table » et revenir à la fenêtre « Attribute : Random Mobility Profiles ».

Nous cliquons « Close » pour fermer la fenêtre « Attribute : Random Mobility Profiles » et revenir à la fenêtre « Node Model Description : Mobility Config » que nous fermons par

un clic sur « Close » pour revenir à la fenêtre « Choose Fixed Node Model » que l'on ferme par « Cancel » qui nous fait revenir à l'éditeur d'attribut de « Mobility Config ».

Avec cela se termine notre étude de l'existant.

B.4 Nos premiers développements

Nous développons nos modèles mathématiques décrits dans les chapitres précédents, en proto-C, le langage de programmation d'*Opnet Modeler* et en amont, nous développons l'interface utilisateur dédiée qui facilite grandement la mise en œuvre des scénarios de simulation de nos modèles, par l'utilisation d'objets visuels. Mais avant de décrire ces développements, nous présentons dans les section suivantes, le verrou technique majeur que nous avons rencontré, le problème des intersections manquées, sa cause et sa solution.

B.5 Le problème des intersections manquées

Nous essayons donc naturellement de décliner ce modèle afin d'implémenter directement le modèle *Manhattan*, cf. la section 4.2.4 et la Fig. B.9.

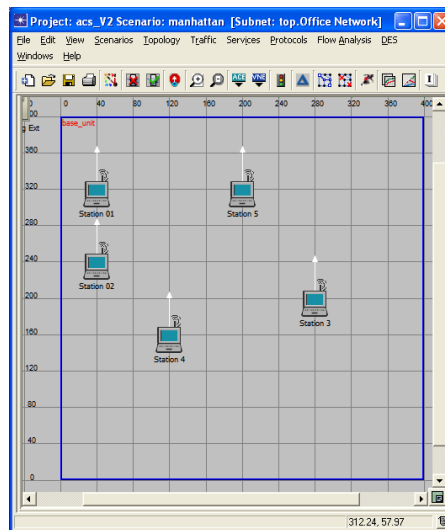


FIGURE B.9 – Implementation de Manhattan dans Opnet Modeler

Nous prenons bien soin de :

- placer chaque nœud sur une intersection, à l'initialisation du scénario ;
- définir l'incrément c'est-à-dire le pas de progression d'un nœud de sorte à ce qu'il soit bien un diviseur de la longueur standard d'un bloc d'immeuble.

Toutefois, nous constatons que de façon apparemment aléatoire, ce qui en soit est toujours très troublant, les nœuds de notre réseau ne parviennent pas exactement à certains points d'intersection de notre grille Manhattan ! En d'autres termes :

Certains nœuds manquent certaines intersections.

Après de nombreux essais très coûteux en temps, nous finissons par entrevoir la cause et trouver la solution. Afin de les exposer, il nous faut d'abord comprendre les différents systèmes de coordonnées *Opnet Modeler*. C'est l'objet de la section suivante.

B.6 Les 3 systèmes de coordonnées d'*Opnet Modeler*

En premier lieu, nous présentons dans cette section les 3 systèmes de coordonnées définis dans *Opnet Modeler* :

- le *système de coordonnées global* ;
- le *système de coordonnées relatif au sous-réseau* ;
- le *système de coordonnées géocentrique*.

B.6.1 Système de coordonnées global

Dans le *système de coordonnées global*, la position d'un objet du réseau est décrite par ses latitude, longitude et altitude. La latitude précise l'éloignement de l'objet vers le nord ou vers le sud, par rapport à l'équateur. Les latitudes sont positives dans l'hémisphère nord et négatives dans l'hémisphère sud. La longitude précise l'éloignement de l'objet vers l'est ou l'ouest, par rapport au méridien d'origine. Les longitudes positives, de 0° à +180°, sont dans l'hémisphère oriental et les longitudes négatives, de 0° à -180°, sont dans l'hémisphère occidentale. L'altitude est la hauteur de l'objet à la verticale de son lieu. L'altitude est mesurée en mètres au-dessus du niveau des mers. Voir la Fig. B.10.

Pour les sous-réseaux et les nœuds contenus dans un sous-réseau dont les coordonnées sont données en degrés, les attributs liés à la position de ces objets sont relatifs au *système*

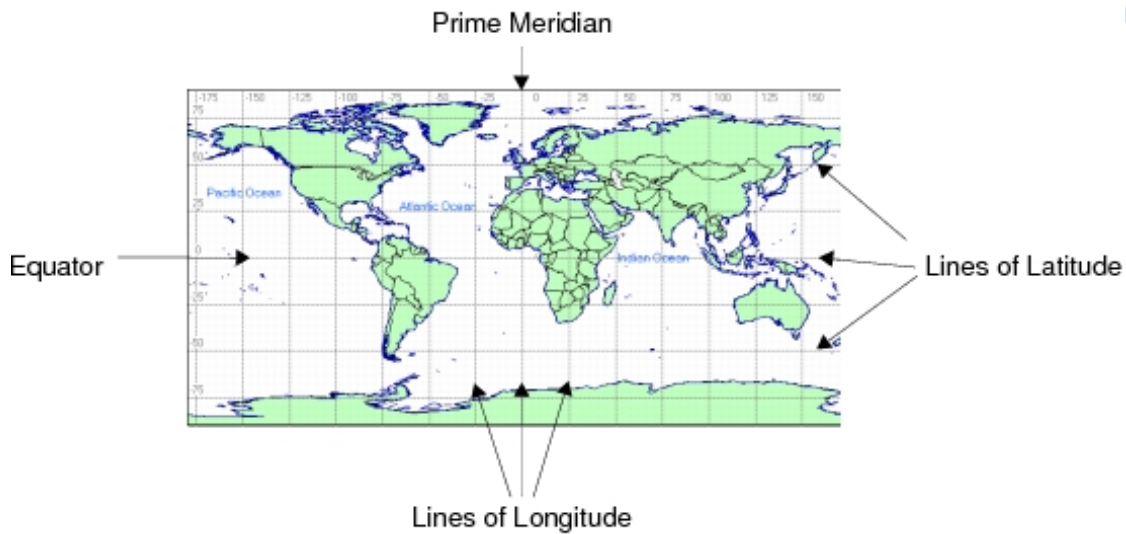


FIGURE B.10 – Le système de coordonnées global d’Opnet Modeler

de coordonnées global. Par exemple, dans le sous-réseau racine, l’attribut « position x » d’un nœud est sa longitude, tandis que son attribut « position y » est sa latitude ; enfin, son attribut « altitude » est son altitude en mètres.

Quel que soit le sous-réseau dans lequel un nœud se trouve, sa position globale peut être obtenue lors de la simulation par un appel à `op_ima_obj_pos_get()` qui est une fonction de base (*Kernel Procedure*). Si tous les parents d’un nœud sont des sous-réseaux fixes, les valeurs retournées pour un nœud fixe ne changent pas au fil du temps. Pour les nœuds mobiles ou satellitaires, les valeurs retournées reflètent la position du nœud à l’instant précis de l’appel de la fonction de base, dans la simulation.

B.6.2 Système de coordonnées relatif au sous-réseau

Dans le *système de coordonnées relatif au sous-réseau*, la position d’un objet du réseau est déterminée par son emplacement dans son sous-réseau. Si l’unité de ce dernier n’est pas le degré, les coordonnées de l’objet représentent son décalage par rapport à l’origine de son sous-réseau, qui est son coin supérieur gauche.

Un site possède les attributs « position x », « position y » et (à l’exception des sous-réseaux fixes) « altitude » qui définissent sa localisation. Dans le *système de coordonnées*

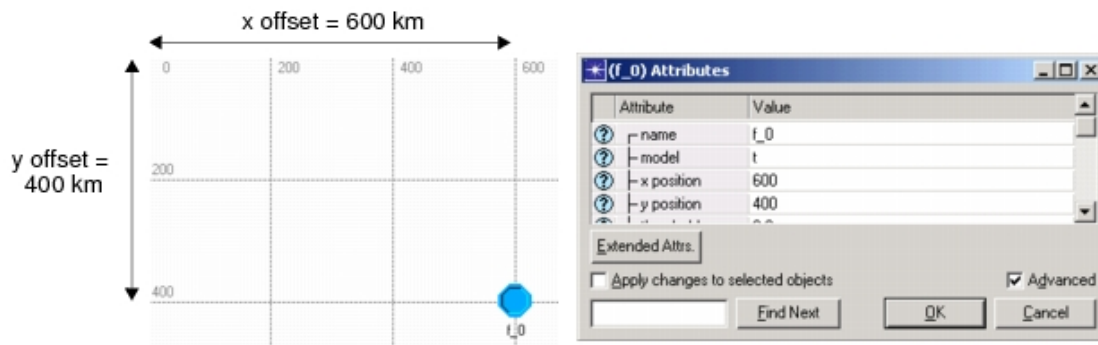


FIGURE B.11 – Le système de coordonnées relatif au sous-réseau d'Opnet Modeler

relatif au sous-réseau, les valeurs d'attribut sont dans l'unité du sous-réseau. L'attribut « position x » spécifie le décalage en abscisse et l'attribut « position y » spécifie le décalage en ordonnée par rapport à l'origine du sous-réseau, i.e. son coin supérieur gauche. L'attribut « altitude » indique l'altitude relative à son sous-réseau. L'élévation du sous-réseau lui-même est sa distance absolue au-dessus du niveau des mers. Les sous-réseaux fixes ont une altitude implicite de 0. À mesure qu'un site mobile ou satellitaire se déplace lors de la simulation, la valeur de ses attributs « position x », « position y » et « altitude » changent en conséquence. Voir la Fig. B.11.

Dans le *système de coordonnées relatif au sous-réseau*, déterminer la position globale d'un objet du réseau peut impliquer un ensemble de calculs récursifs. Si l'étendu de son sous-réseau est spécifié dans le *système de coordonnées global*, l'obtention de la position globale d'un objet ne nécessite que le calcul de son décalage en degrés depuis le coin supérieur gauche du sous-réseau. En revanche, si le sous-réseau est également dans un *système de coordonnées relatif au sous-réseau*, alors la position globale de son étendue, aussi, doit être calculée. Par conséquent, une série de calculs peut être nécessaire, afin que la position globale du sous-réseau soit connue. Tous ces calculs sont gérés par le noyau de simulation.

B.6.3 Système de coordonnées géocentrique

Le *système de coordonnées géocentrique* est une grille cartésienne tri-dimensionnelle dont l'origine est le centre de la terre. Il comporte trois axes (x, y et z) et a pour unité le

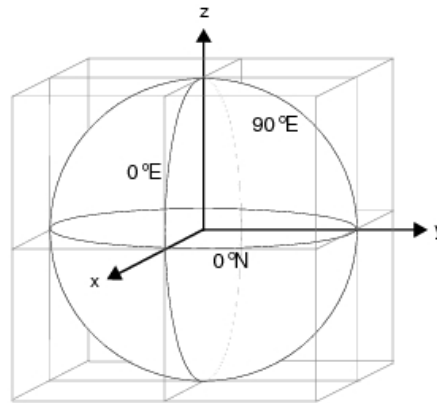


FIGURE B.12 – Le système de coordonnées géocentrique d’Opnet Modeler

mètre. Chacun des axes est défini par un vecteur qui part du centre de la terre et croise un point de la surface terrestre. Voir la Fig. B.12.

B.7 La cause du problème

Quoiqu’*Opnet Modeler* aient effectivement 3 systèmes de coordonnées différents, objet de la section précédente, de façon interne à *Opnet Modeler*, le seul système de coordonnées utilisé est celui des coordonnées géographiques.

Or, cette façon interne de procéder ne répond pas aux besoins de précision de nos modèles de mobilité. En effet, la conversion interne des coordonnées cartésiennes en coordonnées géographiques, n’est pas suffisamment exacte. Nous le constatons en traçant au pas à pas le code source du premier modèle que nous implémentons : *Manhattan*.

C’est ce qui explique que parfois un nœud du réseau « manque » une intersection.

Nous n’en croyions pas nos yeux malgré les tests innombrables que nous avons joués et rejoués ! La cause s’est effectivement avérée être cette imprécision relative de la conversion des coordonnées cartésiennes vers celles géographiques, voir Fig. B.13 où le décalage (et donc le saut de l’intersection) est multiplié par 1000 environ afin de rendre visible l’erreur de conversion.

Nous renonçons donc à prendre comme point de départ *Random Waypoint* et sa représentation interne des coordonnées.

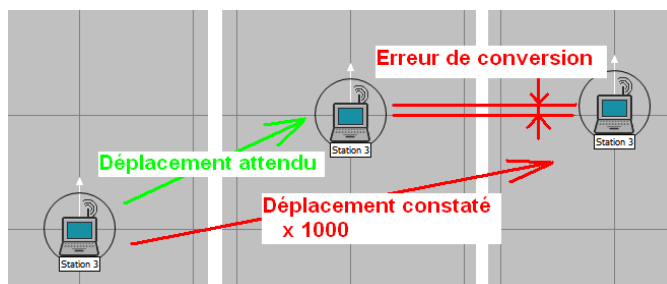


FIGURE B.13 – Erreur de conversion dans Opnet Modeler X 1000

B.8 La solution du problème

La solution du problème des intersections manquées est donc, malgré le fait que cela allait nous causer un effort de développement logiciel de grosse envergure, comme nous allons le voir, de développer notre modèle *Manhattan* à partir de zéro, afin d'être en coordonnées cartésiennes au niveau de l'Interface-Homme-Machine et au niveau interne. Pour commencer, nous devons créer nos propres objets logiciels d'interface visuelle *Opnet Modeler*, c'est ce dont traite la section suivante.

B.9 Création d'objet

Dans *Opnet Modeler*, nous disposons de 2 méthodes de création d'un objet à partir d'un objet existant :

- **la méthode par dérivation d'objet** depuis la palette des objets. Cette méthode ne permet pas de changer les traitements de l'objet père dans l'objet dérivé ou objet fils. En d'autres termes, le père et le fils partagent le même code. Une *dérivation Opnet Modeler* ne sert donc qu'à réaliser des changements dans la partie visuelle du nouvel objet ;
- **la méthode par duplication d'objet** depuis le gestionnaire de fichiers du système d'exploitation, par copier/coller du fichier de traitement de l'objet, à savoir son modèle de processus dont l'extension est *pr.m* pour *process model*, puis par son intégration dans la palette des objets. Dans ce cas, non seulement l'interface visuelle du nouvel objet est modifiable mais également ses traitements.

B.9.1 Par duplication

Le développement d'un nouvel objet visuel n'étant pas particulièrement bien documenté, nous en recherchons le mode de développement par nous même. Nous l'exposons ci-après en commençant par le premier objet que nous développons pour implémenter *Manhattan* à savoir une extension de l'objet *Mobility Config*.

Il convient ici d'ouvrir une parenthèse. Le nom de fichier du modèle de processus de l'objet visuel *Mobility Config* est : « random_mobility_mgr.pr.m ». On aurait pu s'attendre à ce que le nom de fichier du modèle de processus de l'objet visuel *Mobility Config* soit justement « mobility_config.pr.m » mais c'est malgré tout « random_mobility_mgr.pr.m » qui a été choisi. Cela semble indiquer que les développeurs d'*Opnet Modeler* ne pensaient a priori pas implémenter de modèle de mobilité qui soit non aléatoire du point de vue du choix de l'azimut. Nous fermons la parenthèse.

Dans notre cas, nous changeons le comportement de l'objet père. Nous devons donc mettre en œuvre la *méthode par duplication d'objet*, voir la section B.9.

En conséquence, nous commençons par copier/coller le fichier de modèle de processus de l'objet visuel *Mobility Config*, à savoir « random_mobility_mgr.pr.m » en « mobility_config_ext_BIS.pr.m ». L'ajout de « ext » signifie « extension » et le « BIS » exige le *nota* qui suit.

nota

Pour rédiger les procédures de mise en œuvre *Opnet Modeler* de cette thèse, nous les avons entièrement rejouées. Et, afin de disposer de nos nouveaux objets tout en les recréant, nous avons choisi de les recréer en suffixant leurs noms de BIS.

Nous choisissons d'abandonner l'adjectif *random* car nos modèles ne sont pas aléatoires du point de vue de l'azimut comme c'est le cas du *Random Walk*. Par ailleurs, *mgr* qui signifie *manager* c'est-à-dire *gestionnaire* n'apporte pas suffisamment d'information pour le préférer à *conf* qui a l'avantage d'être proche du nom de son objet visuel.

Nous l'intégrons à la palette d'objets comme suit :

1. lancer *Opnet Modeler* ;
2. ouvrir un projet existant ou en créer un nouveau sans changer les valeurs par défaut, l'objectif étant d'accéder à la palette d'objets ;
3. cliquer « Topology | Open Object Palette » afin d'ouvrir la palette d'objets si elle ne s'est pas ouverte automatiquement ;
4. cliquer le bouton  en haut à gauche, afin de passer en « Icon View », voir Fig. B.14 ;

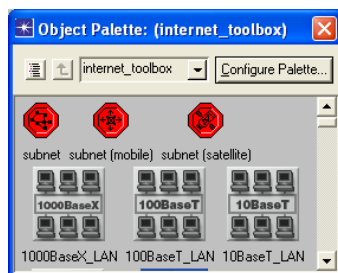


FIGURE B.14 – Boîte de dialogue Icon View d'Opnet Modeler

5. cliquer le bouton « Configure Palette » en haut à droite ;
6. constater l'affichage de la boîte de dialogue « Configure Palette », voir Fig. B.15 ;

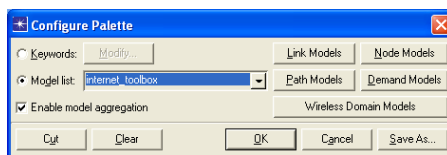


FIGURE B.15 – Boîte de dialogue Configure Palette d'Opnet Modeler

7. cliquer le bouton « Clear » en bas à gauche ;
8. constater que seuls 3 objets demeurent dans la palette, à savoir *subnet*, *subnet (mobile)*, *subnet (satellite)* ;
9. cliquer le bouton « Node Models » ;
10. constater l'affichage de la boîte de dialogue « Select Included Entries » ;
11. se déplacer jusqu'à la ligne « mobility_config_ext » ;

12. cliquer dans la colonne « Status » sur cette ligne afin de basculer de « not included » à « included », voir Fig. B.16 ;

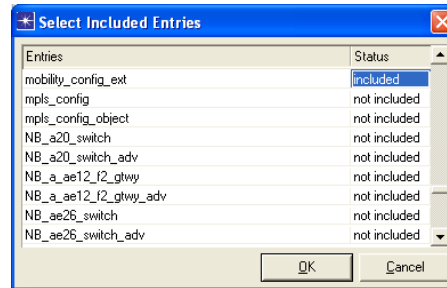


FIGURE B.16 – Boîte de dialogue Select Included Entries d’Opnet Modeler

13. cliquer le bouton « OK » ;
14. constater le retour sur la boîte de dialogue « Configure Palette » ;
15. cliquer le bouton « Save as... » ;
16. constater l’affichage de la boîte de dialogue standard « Save As » ;
17. se déplacer jusqu’au dossier suivant :
`D:\Program Files\OPNET\16.0.A\models\std\manet`
où D représente la lettre du lecteur sur lequel est installé *Opnet Modeler* ;
18. nommer le fichier « MANET_EXT » qui aura par défaut l’extension « .cml » pour
« Custom Model List » ;
19. constater l’affichage de « MANET_EXT » dans la liste de choix « Model List » ;
20. constater l’affichage de l’icône de l’objet « Mobility Config » à la suite des 3 objets
sus-cités ;
21. constater l’affichage de « mobility_config_ext » sous cette icône, voir Fig. B.17 ;
22. cliquer le bouton « OK » ;
23. constater le retour sur la palette d’objets ;
24. cliquer le bouton « OK » pour la fermer.

À ce stade, notre nouvel objet est ajouté dans la palette d’objets dans la nouvelle
branche « MANET_EXT ».

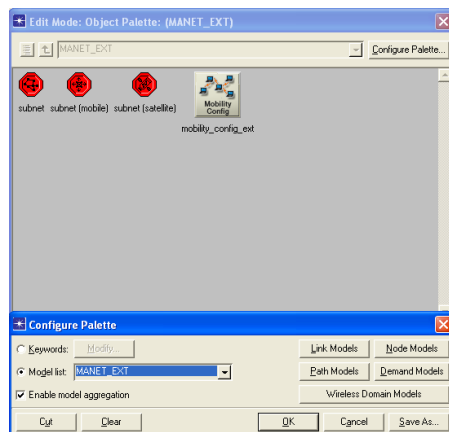


FIGURE B.17 – Configure Palette après ajout de l'objet `mobility_config_ext`

Il nous faut cependant procéder à 2 mises au point sur notre nouvel objet, l'une facultative mais de bon aloi afin de lui donner un nom qui suive le standard des noms des objets dans la palette, l'autre impérative pour distinguer de façon visuelle notre objet de son parent, à savoir lui donner une icône distinctive.

B.9.2 Par dérivation

Afin qu'un nom d'objet suive le standard de désignation des objets dans la palette, son nom doit être formé de mots écrits en lettres minuscules sauf la lettre initiale qui est en majuscule. Ces mots doivent être espacés par des espaces et non des sous-tirets, comme c'est le cas actuellement. En d'autres termes, « `mobility_config_ext` » doit devenir « Mobility Config Ext ». Pour ce faire, nous déroulons les étapes complémentaires suivantes qui correspondent à la première méthode de création d'objet, à savoir, la *méthode par dérivation d'objet*, voir la section B.9 :

1. cliquer « Topology | Open Objet Palette » ;
2. constater l'affichage de la palette d'objets ;
3. se déplacer jusqu'à la branche « MANET_EXT » et l'ouvrir puis ouvrir « Node Models » ;
4. cliquer droit sur notre nouvel objet « `mobility_config_ext` » et choisir « View Model Details » ;

5. constater l’affichage de la boîte de dialogue « Node Model Description : mobility_config_ext », voir Fig. B.18 ;

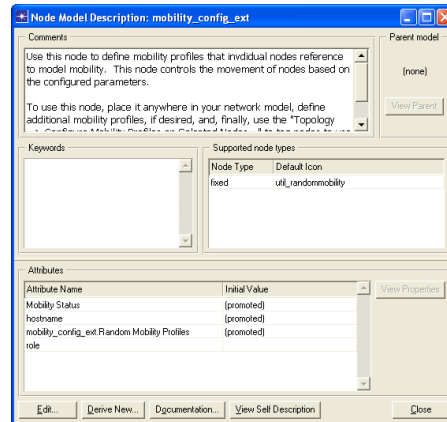


FIGURE B.18 – Configure Palette après ajout de l’objet mobility_config_ext

6. cliquer le bouton « Derive New... » ;
7. constater l’affichage de la boîte de dialogue « Derive New Node Model », voir Fig. B.19 ;

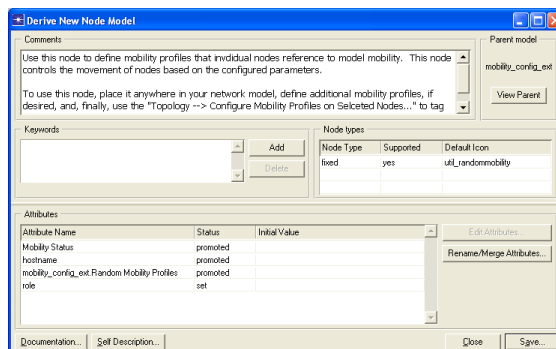


FIGURE B.19 – Boîte de dialogue Derive New Node Model d’Opnet Modeler

8. cliquer le bouton « Save... » ;
9. constater l’affichage de la boîte de dialogue standard « Save As » ;
10. se déplacer jusqu’au dossier suivant :
`D:\Program Files\OPNET\16.0.A\models\std\wireless`
où D représente la lettre du lecteur sur lequel est installé *Opnet Modeler* ;
11. nommer le fichier « Mobility Config Ext » qui aura par défaut l’extension « .nd.m » pour « Node Model » et cliquer « OK » ;

12. constater que l'affichage de la boîte de dialogue « Derive New Node Model » a été supplanté par celui de la boîte de dialogue « Edit Derived Model : Mobility Config Ext », voir Fig. B.20 ;

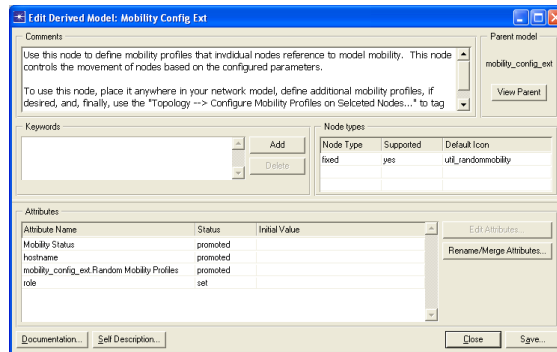


FIGURE B.20 – Boîte de dialogue Edit Derived Model d'Opnet Modeler

13. relever le nom de l'icône (qui est par défaut celle de l'objet père) qui est situé dans le tableau « Node Types » dans la colonne « Default Icon », à savoir : *util_randommobility* ;
14. cliquer sur ce nom ;
15. constater l'affichage de la boîte de dialogue « Icon Palette », voir Fig. B.21 ;

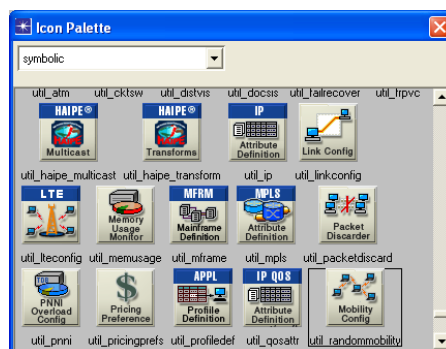


FIGURE B.21 – Boîte de dialogue Icon Palette d'Opnet Modeler

16. relever le nom situé dans la boîte de liste sous le titre de la boîte de dialogue : il s'agit de la base de données où l'icône est situé qui est donc : *symbolic* ;
17. cliquer sur la case de fermeture (la croix blanche sur un carré rouge) en haut à droite de la boîte de dialogue afin de la fermer ;

18. cliquer le bouton « Close » afin de fermer la boîte de dialogue « Edit Derived Model : Mobility Config Ext » ;
19. cliquer le bouton « Close » afin de fermer la palette d'objets ;

Le nom de l'icône de l'objet père est donc :

util_randommobility.

Tandis que sa base de données d'icônes est :

symbolic.

Nous nous servons de ceux-ci à la section suivante.

B.9.3 Création d'icône

Afin de dupliquer l'icône de l'objet père et la modifier pour l'attribuer à notre nouvel objet, nous avons besoin de connaître le nom de l'icône de l'objet père. Par ailleurs, il nous faut connaître également le lieu où cette icône se trouve dans l'arborescence des fichiers d'*Opnet Modeler*. Fort heureusement, nous les avons relevés à la section précédente, il s'agit respectivement de : *util_randommobility* et *symbolic*.

Procédons maintenant à la duplication et à la modification de l'icône parente :

1. cliquer « File | Open... » ;
2. constater l'affichage de la boîte de dialogue standard « Open » ;
3. sélectionner le type de fichier « Icon Database File (*.icons) » ;
4. se déplacer jusqu'au dossier suivant :
`D:\Program Files\OPNET\16.0.A\models\std\icons`
où D représente la lettre du lecteur sur lequel est installé *Opnet Modeler* ;
5. sélectionner le fichier « symbolic » qui a pour extension « .icons » ;
6. rechercher visuellement (il n'y a pas d'autre moyen... il s'agit en gros de la 800^{ième} icône...), l'icône *util_randommobility* ;
7. cliquer droit sur l'icône et choisir « Copy » afin de copier l'icône dans le presse-papier ;
8. cliquer dans le menu principal « Edit | Paste » afin de coller la nouvelle icône en cliquant dans l'espace libre de la base d'icône ;

9. cliquer droit sur la nouvelle icône et choisir « Set Name » et la nommer en : *util_randommobility_ext* ;
10. cliquer droit sur la nouvelle icône et choisir « Edit » ;
11. constater l'affichage de la boîte de dialogue « Image Editor », voir Fig. B.22 ;

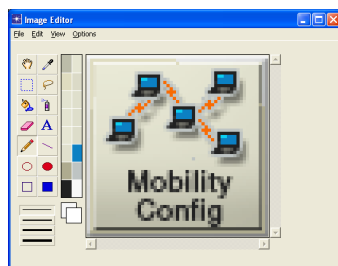


FIGURE B.22 – Boîte de dialogue Image Editor d'Opnet Modeler

12. cliquer « Options | Edit Colors... » afin de changer de palette de couleurs ;
13. constater l'affichage de la boîte de dialogue « Color Palette Editor », voir Fig. B.23 ;

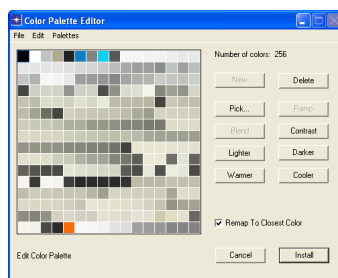


FIGURE B.23 – Boîte de dialogue Color Palette Editor d'Opnet Modeler

14. cliquer « Palettes | Palette couleurs... » afin de passer à la palette couleurs ;
15. cliquer le bouton « Install » ;
16. constater que la couleur rouge est maintenant disponible par les couleur de l'« Image Editor » ;
17. cliquer l'outil texte (lettre A), cliquer la couleur rouge et écrire « Ext » au milieu de la nouvelle icône, voir Fig. B.24 ;
18. cliquer sur « File | Save » afin d'enregistrer la modification de l'icône ;
19. cliquer sur « File | Close » pour fermer l'éditeur d'icône et revenir à la base d'icône ;
20. cliquer sur « File | Save » afin d'enregistrer la modification de la base d'icônes ;

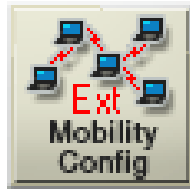


FIGURE B.24 – Icône distinctive de notre nouvel objet *Mobility Config Ext*

21. cliquer sur « File | Close » pour fermer la base d'icône ;

B.9.4 Intégration d'icône et désignation standard

De retour sur la fenêtre projet, nous insérons l'objet possédant la bonne désignation dans la palette d'objet et retirons par là même l'objet intermédiaire « mal nommé ». Enfin, nous lui attribuons sa nouvelle icône. Ci-après la procédure de mise en œuvre :

1. cliquer « Topology | Open Object Palette » ;
2. constater l'affichage de la palette d'objets ;
3. cliquer le bouton en haut à gauche afin de passer en « Icon View », voir Fig. B.14, p. 202 ;
4. cliquer le bouton « Configure Palette » en haut à droite ;
5. constater l'affichage de la boîte de dialogue « Configure Palette » ;
6. cliquer le bouton « Clear » en bas à gauche ;
7. constater que seuls 3 objets demeurent dans la palette, à savoir *subnet*, *subnet (mobile)*, *subnet (satellite)*, voir Fig. B.15, p. 202 ;
8. cliquer le bouton « Node Models » ;
9. constater l'affichage de la boîte de dialogue « Select Included Entries » ;
10. se déplacer jusqu'à la ligne « Mobility Config Ext » ;
11. cliquer dans la colonne « Status » sur cette ligne afin de basculer de « not included » à « included » ;
12. cliquer le bouton « OK » ;
13. constater le retour sur la boîte de dialogue « Configure Palette » ;

14. cliquer le bouton « Save as... » ;
15. constater l'affichage de la boîte de dialogue standard « Save As » ;
16. se déplacer jusqu'au dossier suivant :
`D:\Program Files\OPNET\16.0.A\models\std\manet`
où D représente la lettre du lecteur sur lequel est installé *Opnet Modeler* ;
17. nommer le fichier « MANET_EXT » et répondre « OK » à la confirmation de remplacement dudit fichier ;
18. constater l'affichage de « MANET_EXT » dans la liste de choix « Model List » ;
19. constater l'affichage de l'icône de l'objet « Mobility Config » à la suite des 3 objets sus-cités ;
20. constater l'affichage de « Mobility Config Ext » sous cette icône ;
21. cliquer le bouton « OK » ;
22. constater le retour sur la palette d'objets ;
23. cliquer droit sur notre nouvel objet et choisir « View Model Details » ;
24. constater l'affichage de la boîte de dialogue « Node Model Description : Mobility Config Ext » ;
25. cliquer le bouton « Edit... » ;
26. constater que l'affichage de la boîte de dialogue « Edit Derived Model : Mobility Config Ext », voir Fig. B.20, p. 206 ;
27. cliquer sur le nom de l'icône situé dans le tableau « Node Types » dans la colonne « Default Icon », à savoir : *util_randommobility* ;
28. constater l'affichage de la boîte de dialogue « Icon Palette » ;
29. choisir la nouvelle icône, à savoir : *util_randommobility_ext* ;
30. cliquer le bouton « Save... » ;
31. constater l'affichage de la boîte de dialogue standard « Save As » ;
32. se déplacer jusqu'au dossier suivant :
`D:\Program Files\OPNET\16.0.A\models\std\wireless`

- où D représente la lettre du lecteur sur lequel est installé *Opnet Modeler* ;
33. nommer le fichier « Mobility Config Ext » et cliquer « OK » : pas de demande de confirmation de remplacement dudit fichier (...);
 34. cliquer le bouton « Close » afin de fermer la boîte de dialogue « Edit Derived Model : Mobility Config Ext » ;
 35. cliquer le bouton « Close » afin de fermer la palette d'objets ;

B.10 Attribut d'objet

À ce stade, nous disposons d'un nouvel objet avec un nom qui suit la désignation standard, une icône dérivée de l'icône mère mais spécifique au nouvel objet. Toutefois, les attributs de notre nouvel objet visuel sont en tous points identiques à ceux de l'objet père.

B.10.1 Nom d'attribut

L'étape suivante consiste donc à enrichir l'interface visuelle de notre nouvel objet afin qu'il propose parmi ses attributs, le modèle de mobilité *Manhattan*.

Cela demande non seulement d'ajouter des attributs à notre objet mais aussi et d'abord de modifier certains noms d'attribut afin de les mettre en conformité avec ce qui va être ajouté.

Par exemple, nous avons déjà mentionné au début de la section B.9.1, dans une parenthèse explicite, que le nom de fichier du modèle de processus de l'objet visuel *Mobility Config* était : « random_mobility_mgr.pr.m » et qu'on aurait pu s'attendre à ce que le nom de fichier du modèle de processus de l'objet visuel *Mobility Config* soit justement « mobility_config.pr.m » mais c'est malgré tout « random_mobility_mgr.pr.m » qui a été choisi. Nous avons proposé d'interpréter cela comme semblant indiquer que les développeurs d'*Opnet Modeler* ne pensaient a priori pas implémenter de modèle de mobilité qui soit non aléatoire du point de vue du choix de l'azimut. En conséquence, lors du copier/coller du fichier de modèle de processus de l'objet visuel *Mobility Config*, à savoir « random_mobility_mgr.pr.m » nous le renommâmes « mobility_config_ext.pr.m ». Nous choisissons d'abandonner l'adjectif *random* car nos modèles ne sont pas aléatoires

du point de vue de l'azimut. En sus, *mgr* qui signifie *manager* c'est-à-dire *gestionnaire* n'apportait pas suffisamment d'information pour le préférer à *conf* qui a l'avantage d'être proche du nom de son objet visuel.

Il en va de même pour les noms d'attribut ! À cela près que les noms d'attribut sont définissables et re-définissables. C'est-à-dire qu'un même attribut peut avoir un *nom original* lequel est utilisé dans le code des traitements de cet objet, et un *nom visuel* lequel est vu par l'utilisateur. Si de plus, cet objet a un fils, celui-ci pourra, s'il a été créé par la *méthode de dérivation d'objet* (cf. section B.9.2, p. 204), en changer le *nom visuel*, et s'il a été créé par la *méthode de duplication d'objet* ((cf. section B.9.1, p. 201), en changer le *nom original* ! En d'autres termes, le nombre maximal de noms différents que peut posséder un même attribut est :

$$n = 2^d$$

où d est le nombre de générations : $d = 1$ père seul, $d = 2$ père+fil, $d = 3$ père+fil+petit-fil etc..

NOTE IMPORTANTE

Bien qu'il soit toujours théoriquement possible de changer le *nom original* ou le *nom visuel* d'un attribut d'objet, si cet objet a déjà été instancié dans un scénario de projet *Opnet Modeler*, le changement aura pour effet de bord, le retour aux valeurs par défaut de tous les attributs de l'instance ! Il est clair que les conséquences sont catastrophiques, s'il s'agit d'un projet de plusieurs centaines de scénario, comme c'est le cas dans notre thèse : tous les attributs de toutes les instances de l'objet en question seraient à redéfinir dans tous les scénarios !

B.10.2 Modification

À titre d'exemple, affichons les noms originaux des attributs de premier niveau de notre nouvel objet visuel :

1. cliquer « Topology | Open Objet Palette » ;
2. constater l’affichage de la palette d’objets ;
3. se déplacer jusqu’à la branche « MANET_EXT » et l’ouvrir puis ouvrir « Node Models » ;
4. cliquer droit sur notre nouvel objet « Mobility Config Ext » et choisir « View Model Details » ;
5. constater l’affichage de la boîte de dialogue « Node Model Description : Mobility Config Ext » ;
6. cliquer le bouton « View Parent » en haut à droite ; en effet les attributs sont rattachés à l’objet « mobility_config_ext » et non à l’objet « Mobility Config Ext » qui n’est qu’une enveloppe visuelle comme nous l’avons vu à la section précédente ;
7. cliquer le bouton « Edit... » ;
8. constater l’affichage de la fenêtre « Node Model : mobility_config_ext », voir Fig. B.25 ;

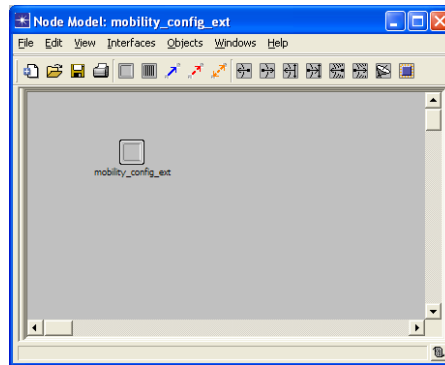


FIGURE B.25 – Fenêtre de l’éditeur de modèle de nœud d’Opnet

9. double-cliquer sur l’unique nœud de la zone cliente de la fenêtre ;
10. constater l’affichage de la fenêtre « Process Model : mobility_config_ext », voir Fig. B.26 ;
11. cliquer dans le menu « Interfaces | Model Attributes »
12. constater l’affichage de la boîte de dialogue « Model Attributes : mobility_config_ext », voir Fig. B.27.

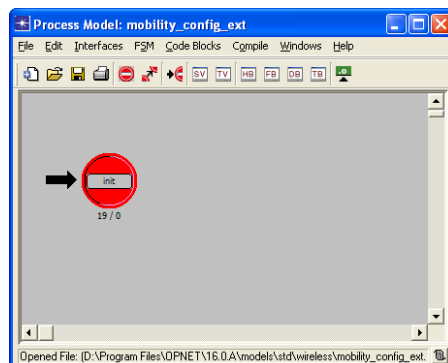


FIGURE B.26 – Fenêtre de l'éditeur de processus d'Opnet

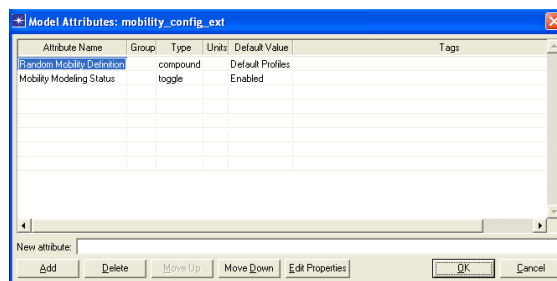


FIGURE B.27 – Boîte de dialogue Model Attributes Initiale de mobility_config_ext

Les sections qui suivent montre les modes opératoires des :

- Modification du nom original d'un attribut d'objet ;
- Modification du nom visuel d'un attribut d'objet.

Modification du *nom original* d'un attribut d'objet

Elle se fait dans la boîte de dialogue « Model Attributes : mobility_config_ext », voir Fig. B.27.

Nous choisissons de changer le nom « Mobility Modeling Status » en « Mobility Status » car c'est lors de la simulation que cette donnée est testée par les traitements, et à ce moment ça n'est pas de modélisation dont il s'agit mais bien de simulation. En outre, « Random Mobility Definitions » devient « Random Mobility Profiles », afin de rapprocher le *nom original* du *nom visuel* et nous simplifier la lecture des traitements.

La boîte de dialogue ci-dessus devient semblable à la Fig. B.28

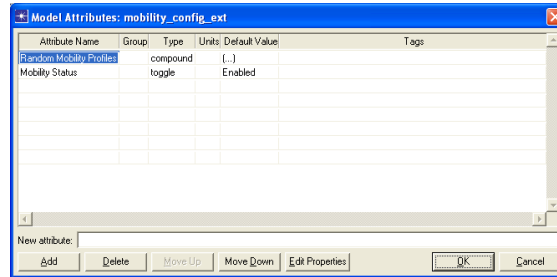


FIGURE B.28 – Boîte de dialogue Model Attributes de Mobility Config Ext

où *Mobility Modeling Status* est devenu *Mobility Status*.

Ensuite, comme nous l'avons déjà mentionné, il nous faut modifier les traitements qui adressent le *nom original*, et qui sont dans le fichier du modèle de processus associé : *mobility_config_ext.pr.m*. Pour ce faire, nous déroulons les actions suivantes (ici pour un seul des 2 attributs ci-dessus) :

1. cliquer le bouton « OK » pour fermer la boîte de dialogue « Model Attributes : mobility_config_ext » et revenir à la fenêtre « Process Model : mobility_config_ext », voir Fig. B.26, p. 214 ;
2. double-cliquer sur la partie *supérieure* du disque rouge de l'état « init » et dans la fenêtre qui s'ouvre remplacer *Mobility Modeling Status* par *Mobility Status*, voir Fig. B.29 ;

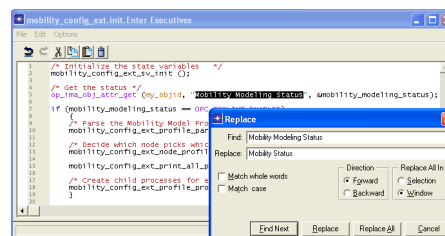


FIGURE B.29 – Modification du code de l'état Init du processus Mobility Config Ext

3. passer en revue les unes après les autres les commandes du menu « Code Blocks » et constater par une recherche de *Mobility Modeling Status* que ce texte n'apparaît dans aucun de ces blocks.
4. cliquer dans le menu « Compile | Compile code » ;

- constater l'affichage de la boîte de dialogue « compilation réussie », voir Fig. B.30 ;

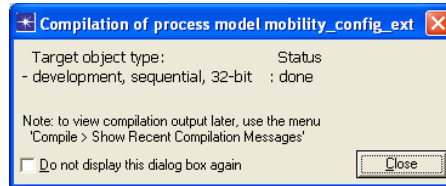


FIGURE B.30 – Boîte de dialogue de compilation réussie

- cliquer le bouton « OK » pour fermer cette boîte de dialogue ;
- cliquer le menu « File | Save » puis « File | Close » pour d'abord sauvegarder le modèle de processus puis fermer cette fenêtre ;
- constater l'affichage de la fenêtre « Node Model : mobility_config_ext », voir Fig. B.25 ;
- cliquer le menu « File | Save » puis « File | Close » pour d'abord sauvegarder le modèle de processus puis fermer cette fenêtre ;
- constater que l'affichage de la boîte de dialogue « Edit Derived Model : Mobility Config Ext », voir Fig. B.20, p. 206 ;
- cliquer le bouton « Close » pour fermer cette fenêtre ;
- constater l'affichage de la palette d'objets ;
- cliquer le bouton « Close » pour la fermer ;

Modification du *nom visuel* d'un attribut d'objet

Elle se fait comme suit :

- cliquer « Topology | Open Objet Palette » ;
- constater l'affichage de la palette d'objets ;
- se déplacer jusqu'à la branche « MANET_EXT » et l'ouvrir puis ouvrir « Node Models » ;
- cliquer droit sur notre nouvel objet « Mobility Config Ext » et choisir « View Model Details » ;

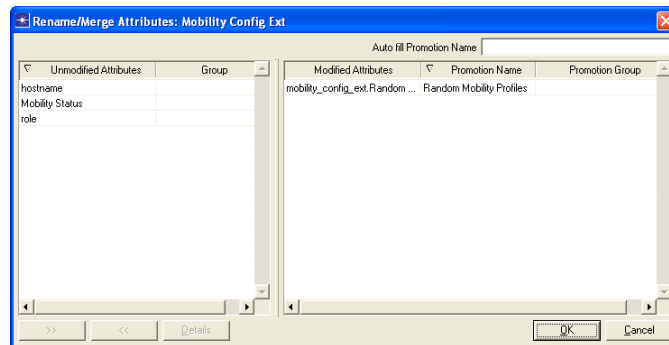


FIGURE B.31 – Boîte de dialogue Rename/Merge Attributes : Mobility Config Ext

5. constater l’affichage de la boîte de dialogue « Node Model Description : Mobility Config Ext » ;
6. cliquer le bouton « Edit... » ;
7. constater que l’affichage de la boîte de dialogue « Edit Derived Model : Mobility Config Ext », voir Fig. B.20 ;
8. cliquer sur l’attribut « Random Mobility Profiles » situé dans le tableau « Attributes » au bas de la boîte de dialogue ;
9. cliquer le bouton « Rename/Merge Attributes... » ;
10. constater l’affichage de la boîte de dialogue « Rename/Merge Attributes : Mobility Config Ext », voir Fig. B.31 ;
11. cliquer dans le tableau de gauche, dans la colonne intitulée « Promotion Name », le nom « Random Mobility Profiles » et le renommer en « Mobility Profiles » ;
12. cliquer le bouton « OK » pour fermer cette boîte de dialogue ;
13. cliquer dans la colonne « Status » de la ligne de l’attribut « Mobility Profiles » afin de le remettre sur « set » plutôt que « promoted » qui signifie que la valeur peut être entrée par l’utilisateur dans les paramètres de la simulation ;
14. cliquer le bouton « Save... » ;
15. constater l’affichage de la boîte de dialogue standard « Save As » ;
16. se déplacer jusqu’au dossier suivant :

D:\Program Files\OPNET\16.0.A\models\std\wireless

- où D représente la lettre du lecteur sur lequel est installé *Opnet Modeler* ;
17. nommer le fichier « Mobility Config Ext » et cliquer « OK » : pas de demande de confirmation de remplacement dudit fichier (...);
 18. cliquer le bouton « Close » afin de fermer la boîte de dialogue « Edit Derived Model : Mobility Config Ext » ;
 19. cliquer le bouton « Close » afin de fermer la palette d'objets ;

NOTA BENE

Cette dernière modification nous l'avons passée alors que nous avions déjà plusieurs centaines de scénarios qui comportaient des instances ce nouvel objet. Et, en vertu de la NOTE IMPORTANTE en fin de section B.10.1, nous avons perdu le paramétrage de de cet objet (central dans nos modèles) dans tous nos scénarios ! D'où l'intérêt d'avoir des *sauvegardes quotidiennes* à jour de notre travail!! Nous avons rapidement descendu la sauvegarde de la veille et avons abandonné cette modification trop tardive.

B.10.3 Création, suppression

La création d'un attribut suit les étapes suivantes :

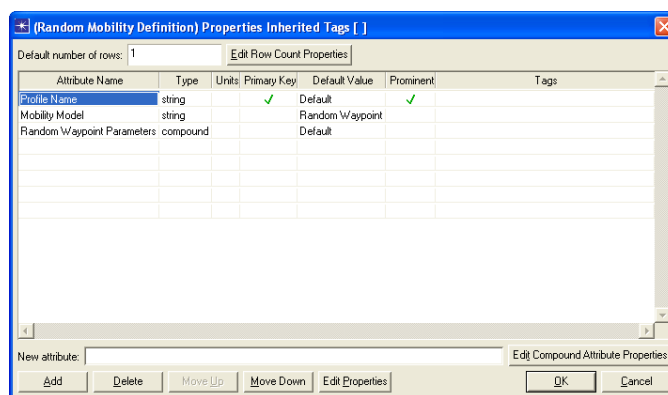


FIGURE B.32 – Boîte de dialogue Random Mobility Definition Properties

1. cliquer « Topology | Open Objet Palette » ;

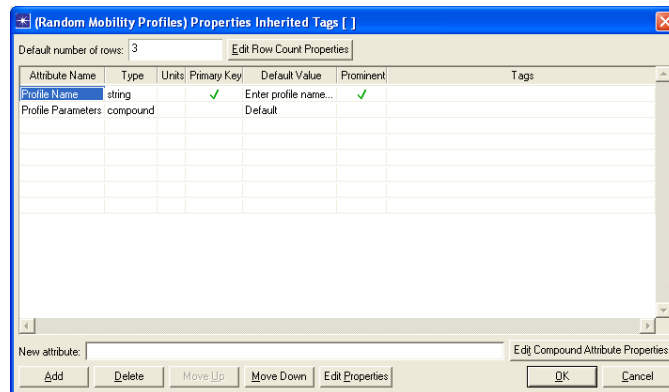


FIGURE B.33 – Boîte de dialogue Random Mobility Definition Properties

2. constater l’affichage de la palette d’objets ;
3. se déplacer jusqu’à la branche « MANET_EXT » et l’ouvrir puis ouvrir « Node Models » ;
4. cliquer droit sur notre nouvel objet « Mobility Config Ext » et choisir « View Model Details » ;
5. constater l’affichage de la boîte de dialogue « Node Model Description : Mobility Config Ext » ;
6. cliquer le bouton « View Parent » en haut à droite ;
7. cliquer le bouton « Edit... » ;
8. constater l’affichage de la fenêtre « Node Model : mobility_config_ext », voir Fig. B.25 ;
9. double-cliquer sur l’unique nœud de la zone cliente de la fenêtre ;
10. constater l’affichage de la fenêtre « Process Model : mobility_config_ext », voir Fig. B.26 ;
11. cliquer dans le menu « Interfaces | Model Attributes »
12. constater l’affichage de la boîte de dialogue « Model Attributes : mobility_config_ext », voir Fig. B.28.
13. cliquer l’attribut « Random Mobility Profiles » ;
14. cliquer le bouton « Edit properties » ;

15. constater l'affichage de la boîte de dialogue « (Random Mobility Profiles) Properties Inherited Tag[] », voir la Fig. B.32.

Observons la boîte de dialogue « (Random Mobility Profiles) Properties Inherited Tag[] », Fig. B.32 et formulons les remarques suivantes :

- l'attribut « Profile Name » est à sa place : c'est bien à ce niveau qu'il faut paramétrer le nom que l'on donne au modèle de mobilité en cours de paramétrage ;
- l'attribut « Random Waypoint Parameters » doit être renommé en « Profile Parameters » : en effet, nous avons plus d'un modèle de mobilité à proposer maintenant ; à noter que nous savons déjà réaliser une modification d'attribut, voir la section B.10.2 ;
- en conséquence, l'attribut « Mobility Model » n'a plus sa place à ce niveau mais au niveau immédiatement inférieur dans l'attribut « Profile Parameters » ;
- sélectionner cette attribut ;
- cliquer le bouton « Delete » ;

Afin d'introduire l'attribut « Mobility Model » à sa bonne place à l'intérieur des « Random Waypoint Parameter », nous créons ce nouvel attribut.

Les développements concernant le code proto-C sont hors du cadre de cette thèse.

B.11 Attribut étendu et XML ou limites de l'outil

Le but de cette section est de donner un exemple concret des limites d'un outil de simulation et de la confiance qu'on peut lui attribuer, notamment pour tout ce qui est relatif aux fonctionnalités peu usitées.

Dans les réseaux denses, 40 nœuds, en OLSR, scénarios 055_XXX à 108_XXX, et en AODV, 163_XXX à 216_XXX, soit donc 108 scénarios au total, nous découvrons à une phase très avancée de notre thèse, les 2 problèmes suivants.

Premier problème, les points d'attraction ont disparu. *Second problème*, l'attribut "Mobility Profile Name" des objets "station_manet_social" n'a plus sa liste déroulante empêchant de changer de modèle de mobilité pour les stations de ces scénarios.

Par exemple, afin de poursuivre nos investigations, nous devons changer environ 15 % des stations des scénarios (en les dupliquant bien sûr) pour les remettre en "Manhattan" de

base et relever l'influence que ces stations ont sur l'ensemble des performances du réseaux selon le modèle de mobilité social choisi : *SMN*, *WMN* ou *AMN*.

Après analyse, nous remontons à la source.

Afin de passer du réseau que nous qualifions de « normal », c'est à dire qui comporte 20 stations, au réseau que nous qualifions de « dense », c'est à dire qui comporte 40 stations, l'idée d'exporter les scénarios *réseau normal* en XML, les dupliquer et les transformer en *réseau dense*, dans ce langage de marquage, puis les importer, nous avait séduits. Après moult essais et nous être bien documentés, sans outil XML professionnel tel que *Altova Mapforce*, la charge de travail est aussi importante que de dupliquer directement les scénarios dans *Opnet Modeler*. Nous abandonnons donc cette voie et décidons de faire évoluer après duplication le premier des 54 scénarios *réseau normal*, puis de dupliquer celui-ci 53 fois afin d'en décliner l'ensemble des scénarios *réseau dense*.

Ce faisant, nous omettons, c'est l'origine du *premier problème*, mais là c'est bien notre faute, de déposer de nouveau, les points d'attractions à leur place. Nous ne nous en apercevons pas de suite. Mais, dès lors que nous en prenons conscience, nous entreprenons de suite la dépose des points d'attraction omis et cela nous coûte une bonne journée de travail d'intervenir dans les 90 scénarios concernés (108 - 18 Manhattan de base qui n'ont pas de point d'attraction).

Mais, c'est l'origine du *second problème*, et là nous ne sommes pas en cause, à notre insu, la fonctionnalité d'export/import XML, nous fait perdre 2 éléments :

- le cadre Mobility Domain ;
- l'éditeur d'attribut associé à "Mobility Profile Name" qui est `mobility_domain_name.ad.m` qui fait apparaître la liste déroulante en question.

Le Mobility Domain étant notre façon de représenter le cadre de la zone de simulation, nous remarquons de suite la chose et y remédions en déposant de nouveau, cet objet, avant duplication des 53 autres scénarios.

En revanche, l'éditeur d'attribut nous échappe, loin de penser que l'import/export XML pourrait commettre une telle bévue!!

Afin de restituer cette liste déroulante, dans l'après coup, sur toutes les stations d'un

scénario, nous déroulons la procédure qui suit.

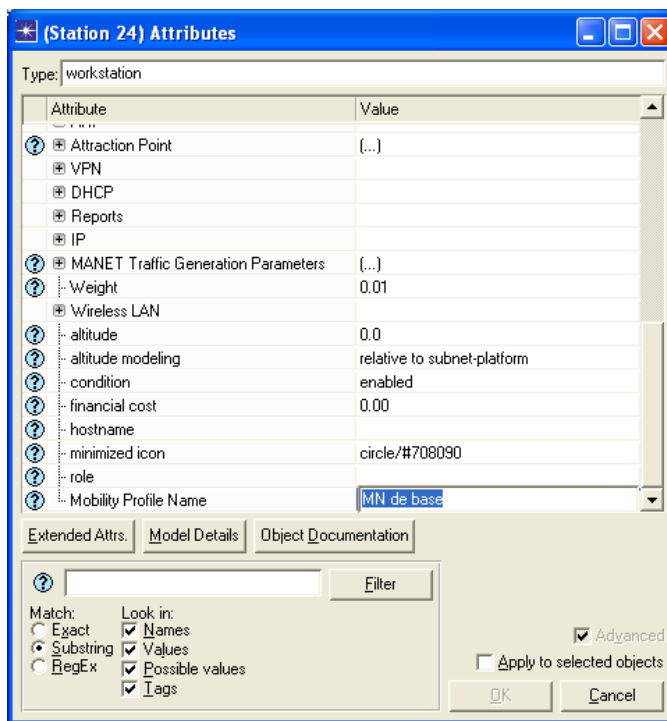
Nous cliquons une station : mettons la Station 24.

Nous cliquons droit puis "Select Similar Node".

Nous constatons que toutes les stations similaires sont sélectionnées.

Nous cliquons droit puis "Edit Attributes (advanced)".

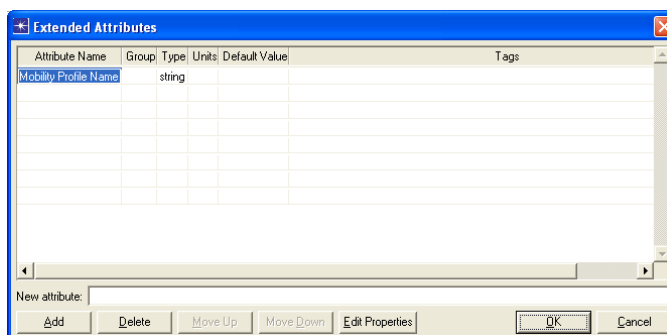
Nous constatons l'affichage de la boîte de dialogue : "(Station 24) Attributes" :



Nous cochons la case : Apply to selected objects.

Nous cliquons le bouton : "Extended Attrs.".

Nous constatons l'affichage de la boîte de dialogue : "Extended Attributes" :

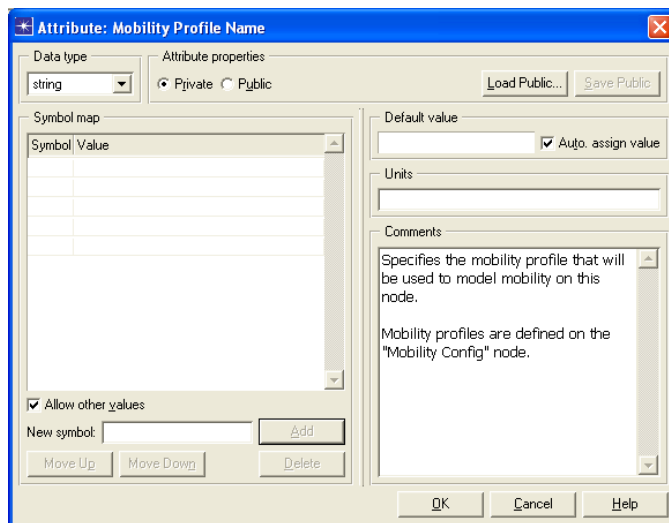


ANNEXE B. CHOIX RELATIFS À LA SIMULATION ET IMPLÉMENTATION DE NOS MODÈLES

Nous constatons que "Mobility Profile Name" est sélectionné ou le sélectionnons dans le cas contraire.

Nous cliquons le bouton : Edit Properties.

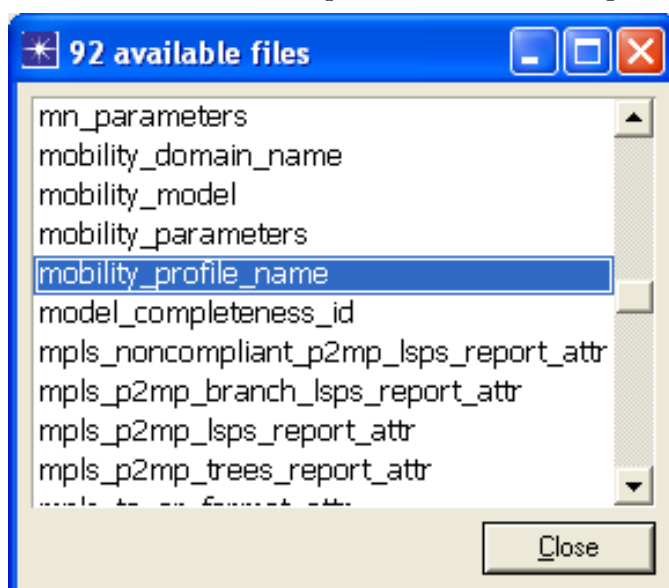
Nous constatons l'affichage de la boîte de dialogue : "Attribute : Mobility Profile Name" :



Nous cliquons la case : "Public" qui devient "Public unnamed".

Nous cliquons le bouton : "Load Public...".

Nous constatons l'affichage de la boîte de dialogue : 92 available files :



Nous cliquons la ligne : `mobility_profile_name`.

Nous constatons la fermeture de la boîte de dialogue : 92 available files.

Nous constatons le retour sur la boîte de dialogue : "Attribute : Mobility Profile Name".

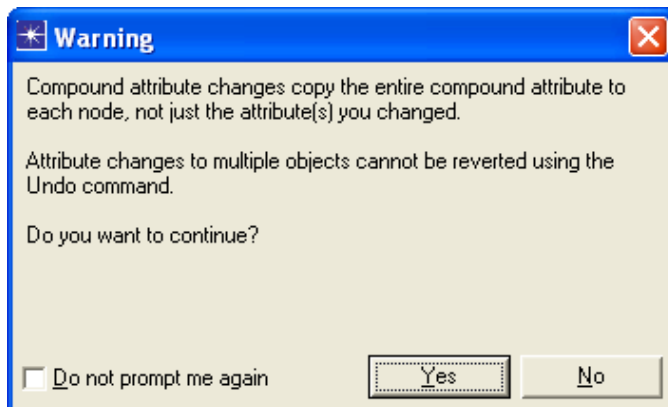
Nous constatons que la case à cocher "Public unnamed" est devenue "Public mobility_profile_name".

Nous cliquons OK pour fermer la boîte de dialogue : "Attribute : Mobility Profile Name".

Nous cliquons OK pour fermer la boîte de dialogue : "Extended Attributes".

Nous cliquons OK pour fermer la boîte de dialogue : "(Station 24) Attributes".

Nous constatons l'affichage de la boîte de dialogue Warning :



Nous cliquons Yes.

Nous rejouons donc cette procédure 108 fois !

C'est dire combien un petit dysfonctionnement d'*Opnet Modeler* non relevé et corrigé au départ peut prendre de l'ampleur avec l'avancement de la thèse : cela nous coûte 2 journées de travail, un travail fort peu passionnant, autant le dire !

Il s'agit d'un exemple assez intéressant pour avoir sa place dans notre thèse, car il est très représentatif des comportements surprenants, pourquoi ne pas dire des bogues, des outils utilisés qui nous causent des pertes de temps qui suivent un accroissement exponentiel avec l'avancement de la thèse.

Annexe C

Routage social

C.1 Introduction

Dans les chapitres précédents, nous avons traité des aspects sociaux de la mobilité en vue d'en améliorer la modélisation ; celle-ci ayant comme objectif unique de rendre plus réalistes les résultats des mesures de performances en milieu simulé des protocoles de routage des réseaux mobiles. Dans ce chapitre, nous abordons les protocoles de routage eux-mêmes en tant que nouvel espace prospectif alternatif à l'utilisation des comportements communautaires dans les réseaux mobiles. Après un succinct état de l'art en section ??, nous nous proposons de revisiter les tenants et les aboutissants du protocole *Optimized Link State Routing (OLSR)* en section C.2. Nous signalons des résultats intéressants concernant : la gestion de l'énergie, *e-OLSR*, en section C.3, ces résultats nous ont inspirés dans notre démarche d'optimisation du protocole *OLSR*. Ensuite, en section C.4, nous passons au cœur de ce chapitre qui est une approche sociale du protocole *OLSR*, *OLSR-S*. En section C.5, nous décrivons notre algorithme *OLSR-S* et nous testons cette approche par la simulation d'un réseau mobile mettant en œuvre ce protocole en section C.6. Nous relatons les résultats des mesures réalisées sur les différents organes de notre réseau mobile simulé en section C.7 et concluons notre chapitre en section C.8 sur les avantages d'une telle démarche.

« L'acheminement d'un message dans un réseau mobile épars est rendu difficile par le fait que le graphe du réseau est rarement (voire jamais) connexe. Dans ces conditions où les nœuds sont libres dans leur déplacement, trouver un chemin qui offre de bonnes performances d'acheminement de bout en bout en un temps réduit est un véritable défi », écrit

Daly and Haahr [2007b] dans un article qui présente une solution multidisciplinaire fondée sur la prise en compte de l'*effet du petit monde* (*small world dynamics*). Celui-ci a été proposé initialement comme levier facilitant les études économiques ou sociales. Récemment, il a été appliqué avec succès à la propagation des informations caractéristiques des réseaux sans fil. À cette fin, certains nœuds ponts sont identifiés en fonction de leur aptitude à être central (au sens central téléphonique), c'est-à-dire, selon leur faculté à jouer le rôle de courtier d'échange d'informations entre des nœuds qui seraient sans cela déconnectés. Compte tenu de la difficulté de mesurer la centralité dans les réseaux peuplés, la notion de *réseaux ego* est exploitée où les nœuds ne sont pas tenus d'échanger des informations globales sur la topologie du réseau, mais uniquement locales. Le routage *SimBet* [Daly and Haahr 2007a] est ensuite proposé, lequel exploite la centralité prévisionnelle (*betweenness*) ainsi que la similitude sociale au nœud destination (*similarity*), déterminée localement. L'article présente des simulations utilisant des données de traces réelles qui démontrent que le routage *SimBet* donne des résultats en termes d'acheminement des messages qui sont proches du routage *épidémique* [Yoneki et al. 2008], mais pour un coût considérablement réduit. De plus, l'article montre que le routage *SimBet* surpasse le protocole de routage *PRoPHET* de Lindgren et al. [2003], en particulier lorsque les nœuds émetteur et récepteur ont une connectivité faible.

Costa et al. [2008] énonce que les applications qui nécessitent la diffusion d'informations destinées aux humains (publicité, nouvelles, alertes environnement) s'appuient souvent sur le modèle publication-abonnement (*publish-and-subscribe*), dans lequel le réseau délivre (un message relatif à) une publication uniquement à ses (nœuds) abonnés. Par essence, le modèle publication-abonnement est particulièrement adapté aux environnements mobiles, en ce sens où il minimise le couplage entre les partis communicants. Toutefois, à notre connaissance, aucun des (rares) travaux qui abordent le modèle publication-abonnement dans les environnements mobiles, n'a encore traité des réseaux humains connectés de façon intermittente. Les personnes socialement liées ont souvent des collocations proches. Cette caractéristique peut être prise en compte dans les décisions d'acheminement au sein d'une couche transport centrée sur l'intérêt, laquelle sous-tend le réseau de publication-abonnement. Cela produit de meilleures performances et accroît la tolérance aux taux élevés de mobilité ainsi

que la capacité à endiguer de longs épisodes de déconnexion. Costa et al. [2008] propose *SocialCast*, un cadre de travail pour le routage de type publication-abonnement qui exploite des prédictions basées sur la mesure de l'interaction sociale (comme par exemple les modes de déplacement à l'intérieur d'une communauté) pour identifier les meilleurs porteurs d'informations. Ce travail met en évidence les principes qui sous-tendent ce protocole, illustre son fonctionnement et évalue ses performances en utilisant un modèle de mobilité basé sur un réseau social validé à l'aide de traces réelles. L'analyse des résultats montre que la prédiction des collocation et mobilité des nœuds permet de publier à leur intention, des événements à un rythme régulier et soutenu, en maintenant au plus bas les trafic de contrôle et latence, indépendamment des fluctuations au niveau de la densité, du nombre de répliquas par message et de la vitesse.

Un protocole de routage pour réseaux mobiles, sensible aux relations sociales, conçu pour le partage de contenu des applications sociales mobiles, est proposé par An et al. [2009]. Depuis qu'un contenu peut être partagé par un groupe d'utilisateurs qui ont des intérêts similaires, cette similitude est un bon outil de prédiction de « *qui va consommer quel contenu* ». Ces intérêts communs peuvent en effet être exploités dans le routage et la réplication des demandes/réponses de contenu afin d'accroître l'efficacité de leur partage. Remarquons que le routage seul détermine quel contenu sera transmis par qui. De plus, un itinéraire est basé sur la similarité des intérêts. Il vient que la réplication du contenu peut avoir lieu dans chacun des nœuds intermédiaires, ces nœuds étant intéressés par ce contenu. L'auteur décrit la conception et la mise en œuvre d'un tel protocole de routage qui, sensible aux relations sociales, donne à celles-ci une place centrale dans le système d'évaluation (*metrics*) du routage. Les résultats de simulation montrent que le protocole de routage proposé a un taux d'acheminement plus élevé et un flux de contrôle réseau plus réduit que ceux du protocole de routage *dynamic Destination-Sequenced Distance-Vector*, *DSDV*, proposé par Perkins and Bhagwat [1994].

C.2 Le protocole *OLSR*

Le protocole *Optimized Link State Routing*, *OLSR* (Routage à État de lien Optimisé) peut être décrit comme une adaptation du *Open Shortest Path First (OSPF)*, *RFC 2328*,

pour les réseaux mobiles. L'optimisation consiste en l'introduction de la notion de *MultiPoint Relay*, *MPR* (Relais Multipoint). L'idée est de réduire le nombre de paquets de contrôle en évitant le *flooding* (inondation), c'est-à-dire le fait que tous les nœuds diffusent systématiquement tous les paquets de contrôle. Dans *OLSR*, un nœud n désigne parmi ses voisins, à un saut, V_{n_1} , ses *MPR*, V_{n_m} , c'est-à-dire ceux qui se chargeront de diffuser ses paquets de contrôle. Ses autres voisins, $V_{n_1} - V_{n_m}$, reçoivent et traitent normalement ses paquets sans en rediffuser aucun. Par ailleurs, n maintient un autre sous-ensemble de ses voisins, V_{n_s} qui comprend ceux qui l'ont désigné leur *MPR*. Les éléments de V_{n_s} sont dits les *MPR Selectors* de n . Tout message en diffusion en provenance de $s \in V_{n_s}$, reçu par n est impérativement diffusé par celui-ci. V_{n_m} et V_{n_s} sont définis et évoluent au rythme des messages *Hello* émis toutes les deux secondes, *Hello Interval*.

Un message *Hello* contient, outre le paramètre *Willingness* qui indique la propension du nœud à assumer le rôle de *MPR*, a_n , l'adresse du nœud émetteur n , et $\forall v \in V_{n_1}$, a_v , l'adresse de v et l_v , le type du lien (n, v) qui prend ses valeurs dans $T_L = \{\text{symétrique, asymétrique, perdu, inconnu, voisin symétrique, MPR, ...}\}$. Tout nouveau lien de (n, v) est déclaré par n comme *asymétrique* a priori. Il devient *symétrique* dès lors que n reçoit à son tour ce même lien en provenance de v . Dès lors v est éligible en tant que *MPR* de n par n et réciproquement. V_{n_1} est ainsi constitué des v tels que (n, v) est *symétrique*. V_{n_2} , les voisins à 2 sauts de n , est constitué des nœuds v_2 tels que v_2 est voisin de $v \in V_{n_1}$ ET $v_2 \notin V_{n_1}$.

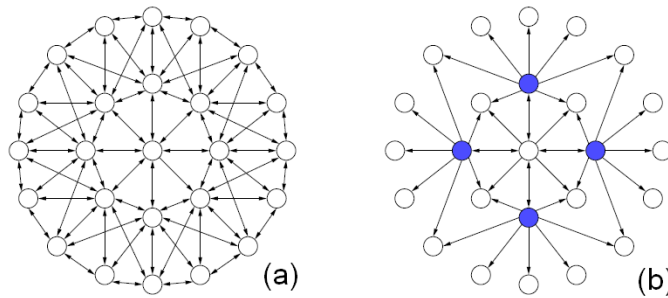


FIGURE C.1 – Gain de retransmission 50 % : 12 au lieu de 24 : (a) sans MPR, (b) avec MPR

La Fig. C.1 montre le gain obtenu par l'utilisation des *MPR* : 12 retransmissions sont

suffisantes pour diffuser un message aux nœuds éloignés de 3 sauts du nœud émetteur i.e. le nœud central de la Fig. C.1a. Sans cette notion, il faut 24 retransmission, Fig. C.1b. Il s'agit de la *première optimisation*. La *deuxième optimisation* vient du fait que dans un message *Topology Control*, *TC*, qui fait connaître la topologie du réseau périodiquement, seul le voisinage des nœuds est transmis et non l'intégralité du réseau comme dans le message *Link State Advertisement*, *LSA* du protocole *OSPF*. Enfin, la *troisième optimisation* vient du fait que ces messages *TC* (ainsi que les autres messages de contrôles *MID* et *HNA* qui sont en dehors de notre propos) ne sont jamais relayés par les nœuds non *MPR*.

C.3 *OLSR* et énergie : e-OLSR

Le gain relaté à la section précédente, Fig. C.1, a bien entendu des conséquences très positives sur la consommation d'énergie du réseau mobile. Mais il est possible d'aller plus loin.

L'efficacité énergétique du routage dans un réseau mobile dépend de l'exactitude du niveau d'énergie instantané, fait remarquer Alhalimi and Kunz [2007]. Les contraintes de bande passante, de coûts de communication, de taux de perte élevés et de dynamique de la topologie des réseaux mobiles, complexifie considérablement ses collecte et mise à jour. L'erreur relative correspond à la différence moyenne entre les niveaux d'énergie perçu (par le nœud qui prend la décision du routage) et réel (connue suite au routage). Elle augmente avec le débit. Quoique le paramétrage de base du protocole *OLSR* ne permette pas de la corriger, différentes techniques sont proposées et la réduisent significativement. Notamment, la *prédiction intelligente* rend un niveau d'énergie précis quelque soit la densité du trafic [Alhalimi and Kunz 2007].

Parallèlement, Ghanem et al. [2005] propose 2 nouveaux mécanismes d'optimisation énergétique dans les réseaux mobiles utilisant *OLSR*. Lorsque par exemple 2 nœuds arrivent ex-æquo dans l'algorithme de choix des *MPR*, le fait de choisir celui qui a le plus d'énergie augmente l'autonomie globale du réseau mobile. Les résultats sont présentés Fig. C.2.

Cette approche, simple et originale, nous a orientés vers une optimisation du protocole *OLSR* par une « optimisation » du choix des *MPR*.

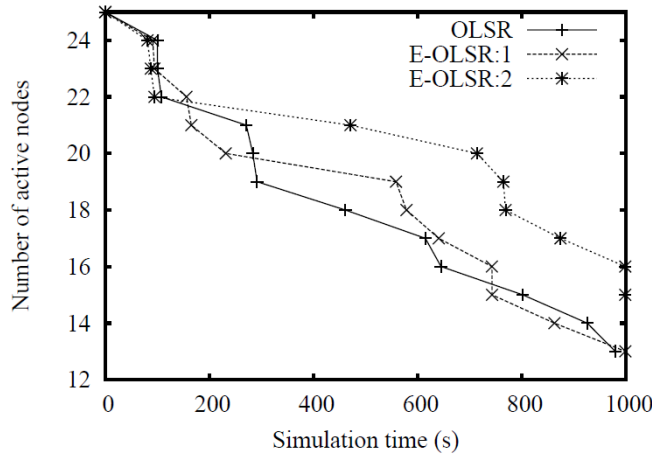


FIGURE C.2 – 2 mécanismes d’optimisation énergétique utilisant OLSR

C.4 OLSR social : OLSR-S

La pertinence de la démarche de Ghanem et al. [2005] vient de son originalité et sa simplicité de mise en œuvre pour des résultats fort intéressants. Du coup, nous nous en inspirons et considérons le cœur du protocole *OLSR*, c’est-à-dire son algorithme de choix *MPR*, comme étant la place idéale pour y introduire la prise en compte des regroupements sociaux des utilisateurs d’un réseau mobile, et créer notre *OLSR Social*.

La notion de critère de regroupement peut être mise en œuvre de différente façon. Un mécanisme de *publication-abonnement* (*publish-and-subscribe*) peut permettre à un nœud d’émettre une offre et aux autres de l’accepter pour une durée déterminée par une date, et une heure de début et une date et heure de fin. Pendant ce laps de temps, notre protocole *OLSR Social* saura s’adapter afin d’être local à ce groupe. Un autre mécanisme plus proche de nos travaux est de considérer que le premier Point d’Attraction visité par un nœud, fait prendre à ce dernier la « couleur locale » du Point d’Attraction en ce sens où celui-ci lui attribue un identificateur de groupe qui lui est propre, cela, bien entendu pour une durée déterminée. On remarque en effet que souvent un groupe se forme à un événement et ne se défait pas de suite. Enfin, rien n’empêche d’attribuer aux nœuds un identificateur de groupe choisi arbitrairement avant de lancer les tests de simulation. L’aspect regroupement étant une spécialité en soi qui va bien au-delà de nos travaux, c’est cette dernière option

que nous choisissons pour l'implémentation de notre algorithme et nos test.

La question qui se pose à nous à présent est la suivante : comment ajouter à moindre frais cette « perception » des groupes communautaires dans *OLSR*? La réponse que nous donnons est la suivante : donner aux nœuds d'un même groupe communautaire la même propension à être *MPR*. En effet, comme mentionné à la section C.2, le paramètre *Willingness* du message *Hello* indique la propension d'un nœud à assumer le rôle d'*MPR*. Ce paramètre peut prendre une valeur comprise entre 0 (*Will_Never*) et 7 (*Will_Always*) avec comme valeur par défaut 3. Rien ne nous empêche a priori de définir une valeur particulière pour chaque groupe communautaire et l'affecter au paramètre *Willingness* de toutes ses stations.

C.5 Application

Nous ajoutons donc de nouvelles valeurs possibles pour le paramètre *Willingness* : *Will_Social_Group_a* où $a = A, B, C, D$ qui définit un identifiant de groupe communautaire. Nous n'avons besoin pour nos tests que de 4 groupes mais rien ne s'oppose à en créer d'avantage si besoin.

En outre, nous ajoutons à l'algorithme de choix *MPR* déroulé au nœud n , la condition suivante écrite en pseudo-code :

```
// Placer le code suivant dans la branche où l'on a le choix
// entre plusieurs candidats MPR et où le critère de choix
// consiste à prendre celui de plus grande Willingness

// Initialiser m, le MPR choisi, à rien
m = null

// n est-il un nœud social ?
SI n.Willingness > Will_Always ALORS

    // n est un nœud social

    // Parcourir les candidats MPR à la recherche de
    // celui qui aurait même Willingness que n

    POUR chaque candidat MPR c FAIRE
        SI c.Willingness = n.Willingness ALORS
            m = c
        SORTIE DU POUR
    FINSI
FINPOUR
FINSI

// Dérouler l'algorithme OLSR de base
SI m = null ALORS
    m = c[0]
    POUR chaque candidat MPR c FAIRE
        SI c.Willingness > m.Willingness ALORS
            m = c
    FINSI
FINPOUR
FINSI
```

Notons que la ligne « SORTIE DU POUR » est optionnelle. Dans la version *avec* « SORTIE DU POUR », nous choisissons le premier candidat MPR qui convient. C'est le choix que nous faisons. Dans la version *sans* « SORTIE DU POUR », nous choisissons le dernier candidat MPR qui convient en abandonnant les éventuels premiers candidats qui aurait pu être retenus.

C.6 Implémentation

Nous comparons notre approche avec le protocole *OLSR* de base en utilisant dans les 2 cas notre modèle de mobilité *Ant Manhattan Mobility Model (AMN)*. Les détails d'implémentation sont donnés en annexe D.

Pour nos études de performances, nous utiliserons la statistiques suivante :

- *Node statistics, OLSR, MPR Status* qui relate l'attribut « *est MPR* » du nœud considéré.

Afin de créer notre réseau et configurer ses différents paramètres, nous y déposons une instance de notre composant *Mobility Config Ext*.

Ce réseau sera composé de :

- 4 points d'attractions : AP_*a* où $a = A, B, C, D$;
- 4 machines nommées "Station An" où $n = 1, 2, 3, 4$ pour le groupe A ;
- 4 machines nommées "Station Bn" où $n = 1, 2, 3, 4$ pour le groupe B.

Le réseau résultat sera :

À chaque changement de voisinage¹ ou de topologie², le calcul³ de l'ensemble des *MPR* est lancé.

Nous étudions l'algorithme *OLSR* et après la détermination du lieu où *Willingness* est testé nous ajoutons notre code définissant nos nouvelles conditions. Notons la concision de l'intervention dans le code *Opnet* dont voici les lignes extraites sans aucune intervention supplémentaire :

```

//*****
/* OLSR SOCIAL --- DEBUT
/*
//**** Remplacer ceci :
/*
/* if (nbr_set_entry_ptr->willingness > willingness)
/*
//**** par cela :
/*
if (node_willingness > OLSRC_WILL_ALWAYS)
{
    if (nbr_set_entry_ptr->willingness == node_willingness)
    {
        mpr_candidate_ptr = nbr_set_entry_ptr;
        willingness = nbr_set_entry_ptr->willingness;
        reachability = nbr_set_entry_ptr->reachability;
        degree = nbr_set_entry_ptr->degree;
    }
}

```

1. apparition/disparition d'un nœud à 1 ou 2 sauts, ou bien nouveau lien asymétrique. Apparition $\iff$ lien devient symétrique. Disparition $\iff$ plus de lien du tout. Lien asymétrique $\iff$ nœud potentiellement nœud à 1 saut.

2. création/suppression/expiration d'interfaces principales ou secondaires de nœud, consécutives à l'arrivée d'un *Topologie Control Message* ou message *TC* ou à l'échéance d'un temporisateur d'expiration de validité d'interface.

3. pour traduire au plus près du nom de fonction mais il s'agit plutôt d'une mise à jour

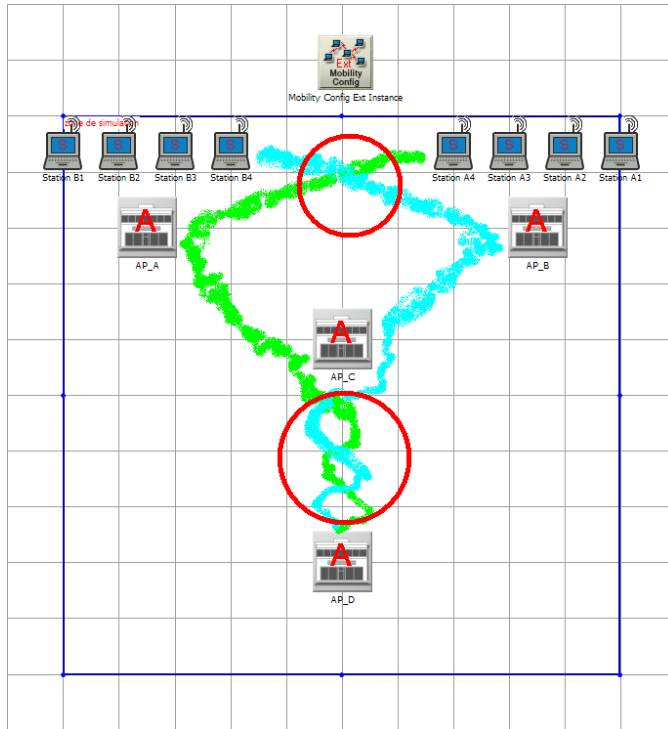
```

else if (nbr_set_entry_ptr->willingness > willingness)
/*
/* OLSR SOCIAL --- FIN
/******

```

On posant que les nœuds *Station Ai* et les nœuds *Station Bi* appartiennent respectivement aux mêmes groupes, les nœuds d'un groupe vont, comme nous l'avons souligné dans les chapitres précédents, avoir les mêmes comportements et se diriger vers les mêmes endroits (i.e. les mêmes PA)

Les trajectoires des nœuds des groupes *A* et *B* sont décrites ci après :



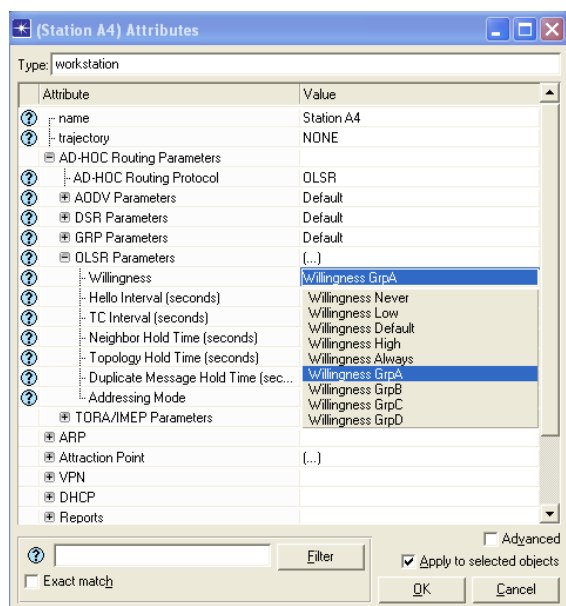
Cette configuration va nous permettre de voir l'impact de notre algorithme *OLSR-S* sur le choix des *MPR*.

Les lieux où la probabilité est grande pour un nœud de changer ses *MPR* sont représentés par 2 cercles.

Afin de forcer les groupes à suivre ces trajectoires, il nous suffit de sélectionner les Points d'Attraction à visiter, cf. annexe D.

Afin de comparer notre nouvelle approche avec l'algorithme *OLSR* de base, nous réalisons 2 scénarios, l'un avec *OLSR* de base et l'autre avec notre *OLSR-S* :

- « Without Social OLSR », laisse à *Will_default* le *Willingness* ;
- « With Social OLSR », initialise le *Willingness* à :
 - *Willingness_grpA* pour le groupe *A* :
 - *Willingness_grpB* pour le groupe *B*.



Notre choix est d'avoir des stations uniformes d'un point de vue fonctionnement (iso-fonctionnelles). Nous paramétrons donc le générateur de trafic uniformément sur toutes les stations.

Enfin, la statistique *OLSR Status* de *Opnet Modeler* est choisie.

OLSR Status est expliquée dans l'aide en ligne comme suit(traduction libre par nos soins) :

« Cette donnée statistique représente l'état "Est MPR" d'un nœud. Elle est à 1 lorsque le nœud est MPR et 0 dans le cas contraire. La fréquence de mise à jour de cette donnée suit celle des messages TC. C'est au moment où un nœud essaie d'émettre un message TC⁴ que cette donnée est mise à jour. D'une certaine manière, cette donnée indique également les points dans le temps où ce nœud a émis un message TC. En effet, seul les nœuds MPR

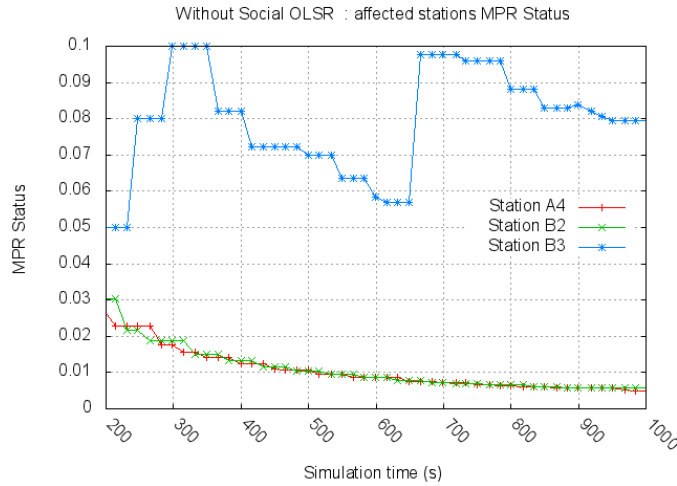
4. cela dépend donc de la taille de la liste des MPR Selectors

peuvent émettre des messages TC ».

Dans nos résultats, nous affichons la valeur moyenne de cette données statistique. Le lecteur ne s'étonnera donc pas de voir des valeurs différentes de 0 ou 1.

C.7 Résultats

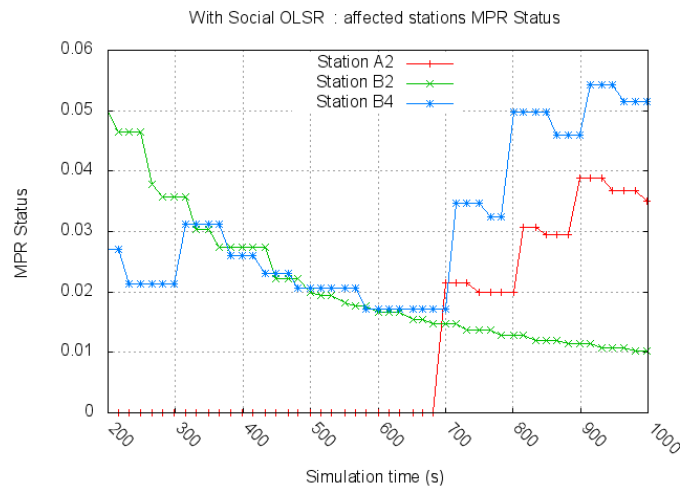
Le premier scénario n'utilise pas *OLSR Social* et donne les courbes suivantes :



ceci représente les 3 stations qui ont joué un rôle de *MPR*. Il s'agit des stations *A4*, *B2* et *B3*. *B3* est depuis le départ bien placée pour devenir l'unique *MPR*. Au bout d'environ 4 minutes, nous constatons effectivement que *A4* et *B2* s'inclinent et cèdent leurs positions de *MPR* au profit de *B3*. Le groupe *A* a donc au finish adopté comme *MPR* une station du groupe *B*.

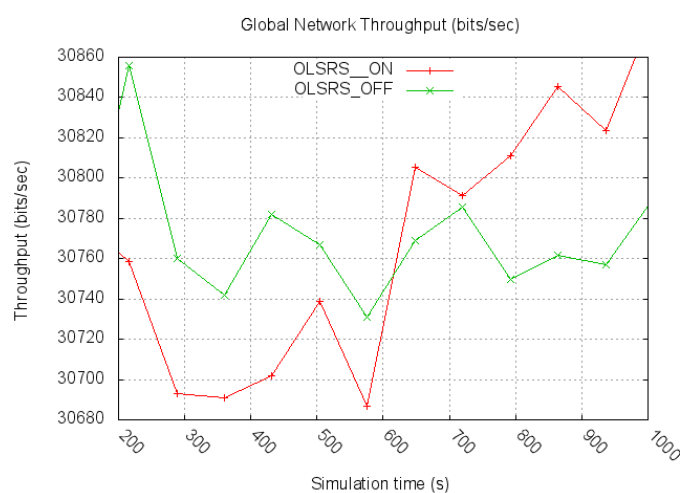
Le second scénario utilise *OLSR Social* et donne les courbes suivantes :

C.7. RÉSULTATS



ceci représente les 3 stations qui ont joué un rôle de *MPR*. Il s'agit des stations *A2*, *B2* et *B4*. *B2* est depuis le départ bien placée pour devenir l'unique *MPR*. Mais, au bout d'une dizaine de minutes, nous constatons au contraire que *B2* s'incline et cède sa position de *MPR* au profit de *A4* et non *B2*. Les stations du groupe *A* ont donc au finish opté pour un *MPR* de leur groupe. Il en va de même au sein du groupe *B*.

Nous pouvons nous poser la question de savoir qu'est-ce que cela change pour le réseau dans sa globalité d'avoir moins de changements d'*MPR*. Nous comparons alors les quantité de données reçues *Throughput* des deux scénarios, et la réponse nous est donnée par les courbes ci-dessous :



Celles-ci montrent que la quantité de données reçue globalement, en moyenne, sur l'ensemble des stations du réseau mobile est de plus en plus élevée lorsqu'on utilise *OLSR*

Social que lorsqu'on ne le met pas en œuvre où elle stagne.

C.8 Conclusion

Dans ce chapitre, nous avons présenté, une autre facette de l'utilisation des comportements communautaires dans les réseaux mobile : l'intégration d'une faculté de « perception » des aspects sociaux aux cœurs des protocoles de routage pour les réseaux mobiles.

Nous avons d'abord donné un bref aperçu de ce qui se fait en la matière à l'heure actuelle.

Ensuite, nous avons proposé un protocole de routage original basé sur le protocole *Optimized Link State Routing* auquel nous adjoignons la composante sociale de perception des regroupements des individus.

Nous avons assorti notre proposition de tests en milieu simulé qui montrent qu'effectivement la stabilité du réseau mobile est plus pérenne lorsque cette perception existe qu'en son absence.

Toutefois, un inconvénient doit être mentionné et il s'agit de la gestion de l'énergie des *MPR* sociaux qui, étant plus stables, risquent fort de consommer plus d'énergie et de devenir sur ce plan, le maillon faible de l'ensemble. D'où l'intérêt d'une solution hybride *e-OSLR + OSLR-S*. Mais, cela est en dehors du cadre de cette thèse.

Annexe D

Implémentation d'OLSR-S

Nous comparons notre approche avec celle par défaut en utilisant dans les 2 cas notre modèle de mobilité *Ant Manhattan Mobility Model (AMN)*. Nous utilisons l'unique statistiques suivante :

- *Node statistics, OLSR, MPR Status* qui relate l'attribut « *est MPR* » du nœud considéré.

Nous déposons une instance de notre composant *Mobility Config Ext* que nous paramétrons comme suit :

Attribute	Value
name	Mobility Config Ext Instance
model	Mobility Config Ext
Random Mobility Profiles	(...)
Number of Rows	1
Ant Social Manhattan de base	
Profile Name	Ant Social Manhattan de base
Profile Parameters	(...)
Mobility Model	Ant Social Manhattan
Ant MN Parameters	(...)
Mobility Domain Name	Not Used
x_min (meters)	0.0
y_min (meters)	0.0
x_max (meters)	1,000
y_max (meters)	1,000
Speed (meters/seconds)	uniform_int (0, 3)
Pause Time (seconds)	constant (1)
Start Time (seconds)	constant (10)
Stop Time (seconds)	End of Simulation
Animation Update Frequency (...)	1.0
Record Trajectory	Disabled
Safety Distance (meters)	30
Move Step (meters)	25
Block Length (meters)	50
Maximum Acceptable Factor	(...)
Function Type	Constant Function
Constant Function Paramet...	(...)
Parabola Function Paramet...	(...)
Parabola/Constant Parame...	(...)
Pheromone	(...)
Decay Factor	0.0
Distance Exponent	1.0

Dans View | Background | Set properties, nous paramétrons ceci :

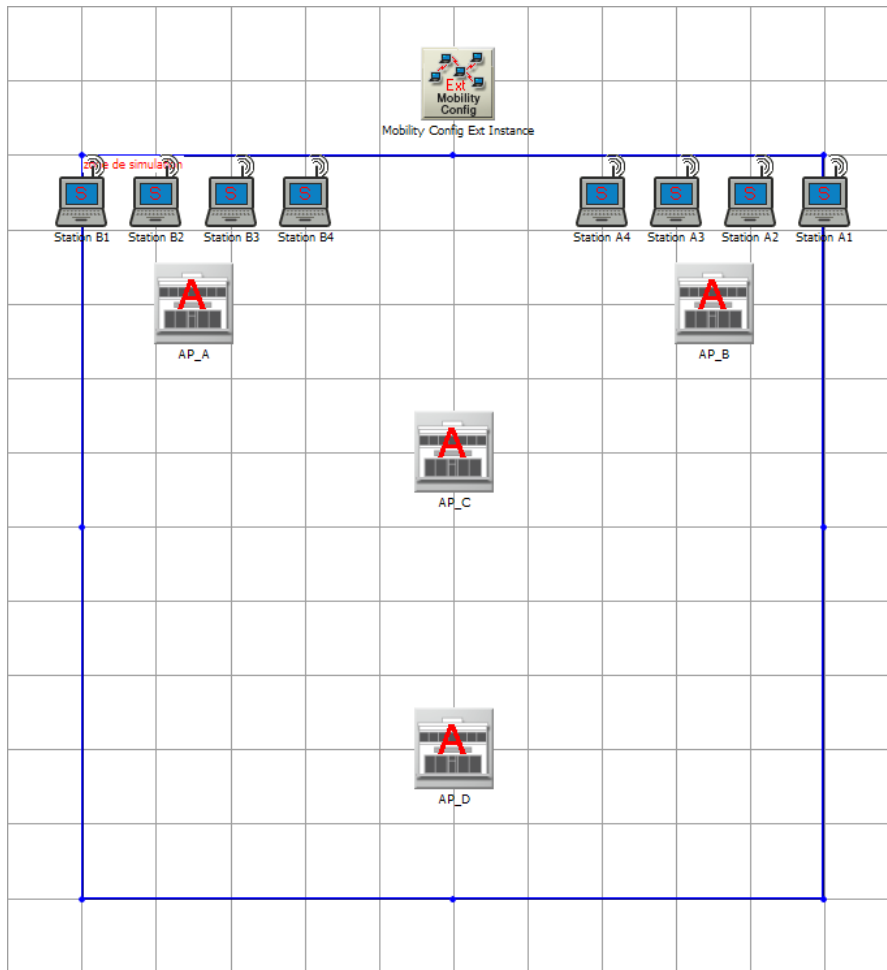
Children positions:	Grid
Units: Meters	Drawing: Solid Division: 50

Nous déposons :

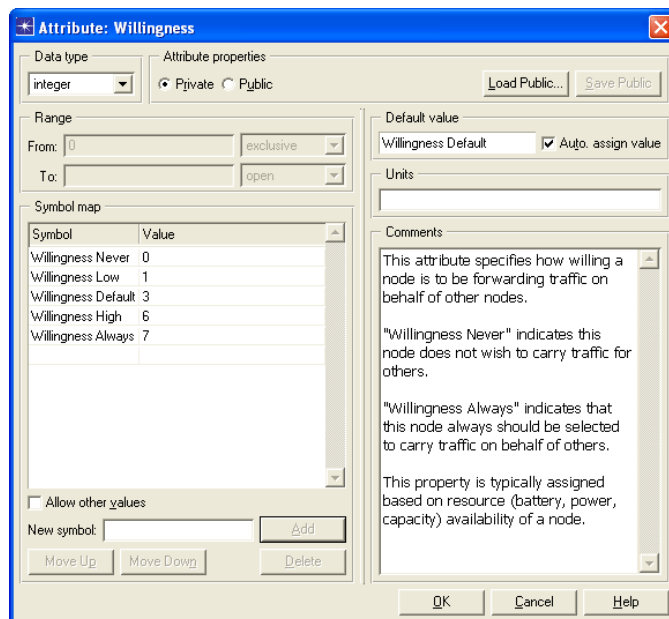
- 4 points d'attractions : AP_a où $a = A, B, C, D$;
- 4 machines nommées "Station An" où $n = 1, 2, 3, 4$ pour le groupe A ;
- 4 machines nommées "Station Bn" où $n = 1, 2, 3, 4$ pour le groupe B.

Cela nous donne le réseau suivant ¹ :

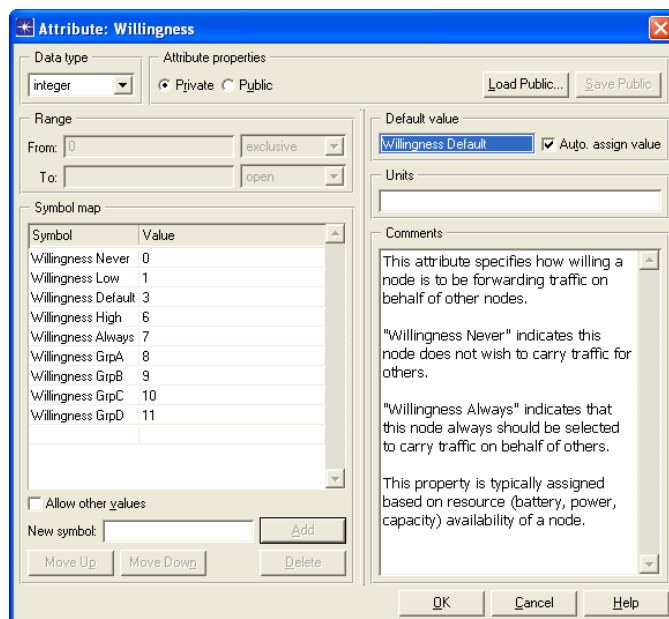
1. nous avons choisi la couleur blanche dans Edit | Preference | UI | Background color



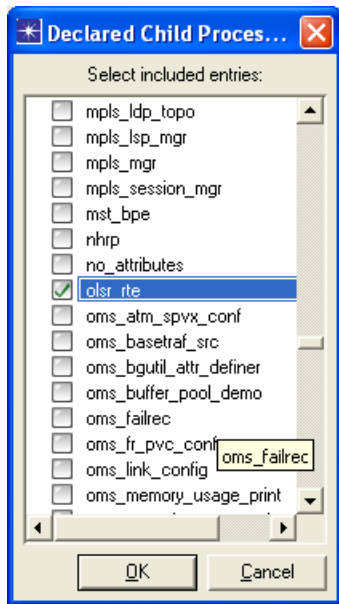
Afin d'ajouter nos nouvelles valeurs de *Willingness*, nous déroulons la procédure décrite à l'annexeB.10.3 en partant d'un de nos composants déjà déposés. Il peut s'agir aussi bien de notre *manet_station_social* que de notre composant *manet_station_attraction_point*. En effet, tous les deux ont même parent : *manet_station_adv_ext*. Or, c'est en éditant le modèle de nœuds de ce dernier que l'on trouve le nœud *manet_rte_mgr*. C'est en double-cliquant sur celui-ci qu'on ouvre son modèle de processus. Arrivé là, on clique Interfaces | Model Attributs puis en sélectionnant OLSR Parameters, on clique sur Edit Properties. Enfin, en sélectionnant Willingness, on clique de nouveau sur Edit Properties qui ouvre la boîte de dialogue suivante :



qui devient suite à nos ajouts, ceci :



Après une série (importante!) de OK, nous retrouvons le modèle de processus *manet_rte_mgr.pr.m*. Nous cliquons sur File | Declare Child Process Models... pour constater que *olsr_rte.pr.m* est bien processus fils de *manet_rte_mgr.pr.m*, comme il apparaît ci-dessous :



Nous ouvrons donc ce modèle de processus et découvrons son code.

Nous en établissons rapidement l'architecture :

- le processus s'initialise ;
- il passe en sommeil ;
- il est réveillé par l'un des événements suivants :
 - arrivée d'un paquet ;
 - échéance d'un temporisateur.

À l'arrivée d'un paquet, la fonction *olsr_rte_pkt_arrival_handle* est appelée. Elle appelle à son tour, suivant le type de message arrivé, *olsr_rte_neighborhood_topology_check_hello* ou *olsr_rte_neighborhood_topology_check_tc* lesquels appellent *olsr_rte_neighborhood_topology_check*.

De même à l'échéance d'un temporisateur, les fonctions *olsr_hello_processing_expiry_handle* ou *olsr_rte_symmetric_link_expiry_handle* sont appelées lesquelles appellent à leur tour *olsr_rte_neighborhood_topology_check*.

Ensuite, quelque soit sa fonction appelante, *olsr_rte_neighborhood_topology_check* appelle *olsr_rte_calculate_mpr_set* qui est l'algorithme de choix des *MPR*, le cœur d'*OLSR*, comme nous le dénommons au début de notre propos.

En résumé, à chaque changement de voisinage² ou de topologie³, le calcul⁴ de l'ensemble des *MPR* est lancé.

Le flot de contrôle de *olsr_rte_calculate_mpr_set* est le suivant :

```
// Construire l'ensemble des nœuds à 2 sauts
olsr_rte_strict_two_hop_set_create
// Déterminer le degré de centralité5 de chaque nœud à 1 saut
olsr_rte_calculate_degree
// Déclarer MPR tous les nœuds à 1 saut passage obligatoire vers un nœud à 2 sauts
olsr_rte_mpr_with_nbr_count_one_add
// TANT QUE tous les nœuds à 2 sauts n'ont pas leur MPR
while (num_strict_two_hops > 0)
    // Déterminer la portée6 des nœuds à 1 saut
    olsr_rte_calculate_reachability
    // POUR tous les nœuds à 1 saut
    for (i=0 ; i< num_keys ; i++)
        // SI nœud à 1 saut courant a un lien non symétrique OU
        // a son Willingness à Will_never OU
        // est déjà MPR (OLSRC_VALUE_MPR=-1) ALORS
        if ((nbr_set_entry_ptr->status == OLSRC_NOT_SYM_STATUS) ||
            (nbr_set_entry_ptr->willingness == OLSRC_WILL_NEVER) ||
            (nbr_set_entry_ptr->reachability == OLSRC_VALUE_MPR))
            // Passer au nœud à 1 saut suivant
            continue
```

2. apparition/disparition d'un nœud à 1 ou 2 sauts, ou bien nouveau lien asymétrique. Apparition $\iff$ lien devient symétrique. Disparition $\iff$ plus de lien du tout. Lien asymétrique $\iff$ nœud potentiellement nœud à 1 saut.

3. création/suppression/expiration d'interfaces principales ou secondaires de nœud, consécutives à l'arrivée d'un *Topologie Control Message* ou message *TC* ou à l'échéance d'un temporisateur d'expiration de validité d'interface.

4. pour traduire au plus près du nom de fonction mais il s'agit plutôt d'une mise à jour

5. nous nommons degré de centralité ce qu'*Opnet* dénomme degree car il s'agit en fait du nombre de lien symétrique dont dispose le nœud.

6. nous traduisons *reachability* par portée car il s'agit en fait du nombre de nœuds à 2 sauts qu'on ne peut atteindre que par ce nœud à 1 saut. À noter, *Opnet* n'a pas hésité à utiliser un substantif exactement inverse à celui dont il s'agit...

```

// FIN SI
// SI nœud à 1 saut courant a meilleur7 Willingness ALORS
if (nbr_set_entry_ptr->willingness > willingness)
    // Le choisir comme candidat MPR
    mpr_candidate_ptr = nbr_set_entry_ptr...
// SINON SI nœud à 1 saut courant a même Willigness
// et meilleur Reachability ALORS
else if (... && nbr_set_entry_ptr->reachability > reachability))
    // Le choisir comme candidat MPR
    mpr_candidate_ptr = nbr_set_entry_ptr...
// SINON SI nœud à 1 saut courant a même Willigness
// et Reachability et meilleur Degree ALORS
else if (... && nbr_set_entry_ptr->degree) > degree))
    // Le choisir comme candidat MPR
    mpr_candidate_ptr = nbr_set_entry_ptr...
// FIN SI
// FIN POUR
// Déclarer ce nœud MPR
mpr_candidate_ptr->reachability = OLSRC_VALUE_MPR;
// Ré-évaluer les nœuds à 2 sauts restants
olsr_rte_update_strict_two_hop_set(&num_strict_two_hops, mpr_candidate_ptr)
// FIN TANT QUE

/* Optimization of MPR set should happen here. */
/* Not required if all the nodes in the network */
/* have same Willingness. MPR calculation */
/* itself gives optimized set. More optimization*/
/* will only be helpful when nodes in network */
/* have different level of WILLINGNESS */
/* Extra optimization currently not implemented.*/

```

7. meilleur que les nœuds à 1 saut qui l'ont précédé dans la boucle POUR.

```
// Construire l'ensemble des nœuds à 2 sauts
olsr_rte_build_new_mpr_set
/* Note: Here we are not doing an explicit check */
/* if the new MPR set is different from the */
/* previous one. RFC says : "if different then send */
/* out an Hello message to notify the changes". */
/* We will wait for next scheduled hello rather */
/* than explicit hello message. Lazy method. */
```

Nous mentionnons ci-dessus 2 commentaires d'*Opnet*. Quoique le second soit mentionné à titre humoristique (lazy method...), le premier est pertinent dans notre discussion. En effet, nous aurions bien profité d'une optimisation concernant Willingness. Mais, nous n'en disposons pas.

Quoiqu'il en soit, nous avons trouvé le lieu où Willingness est testé. Nous pouvons ici ajouter notre code. Nous remplaçons donc ceci :

```
// SI nœud à 1 saut courant a meilleur Willingness ALORS
if (nbr_set_entry_ptr->willingness > willingness)
```

par cela :

```
// Notons N : le nœud qui a lancé l'algorithme MPR
// SI N a son Willingness > Will_always ALORS
if (node_willingness > OLSRC_WILL_ALWAYS)
    // SI nœud à 1 saut courant a même Willingness que N ALORS
    if (nbr_set_entry_ptr->willingness == node_willingness)
        // Le choisir comme candidat MPR
        mpr_candidate_ptr = nbr_set_entry_ptr...
// SINON SI nœud à 1 saut courant a meilleur Willingness ALORS
if (nbr_set_entry_ptr->willingness > willingness)
etc.
```

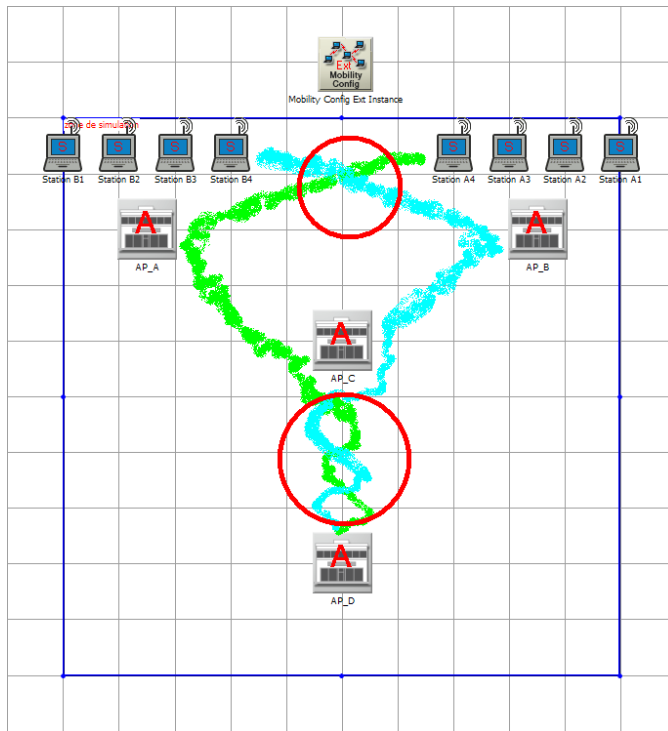
Notons la concision de l'intervention dans le code *Opnet* dont voici les lignes extraites sans aucune intervention supplémentaire :

```

/*****
/* OLSR SOCIAL --- DEBUT
/*
/* Remplacer ceci :
/*
/* if (nbr_set_entry_ptr->willingness > willingness)
/*
/* par cela :
/*
if (node_willingness > OLSRC_WILL_ALWAYS)
{
    if (nbr_set_entry_ptr->willingness == node_willingness)
    {
        mpr_candidate_ptr = nbr_set_entry_ptr;
        willingness = nbr_set_entry_ptr->willingness;
        reachability = nbr_set_entry_ptr->reachability;
        degree = nbr_set_entry_ptr->degree;
    }
}
else if (nbr_set_entry_ptr->willingness > willingness)
/*
/* OLSR SOCIAL --- FIN
*****/

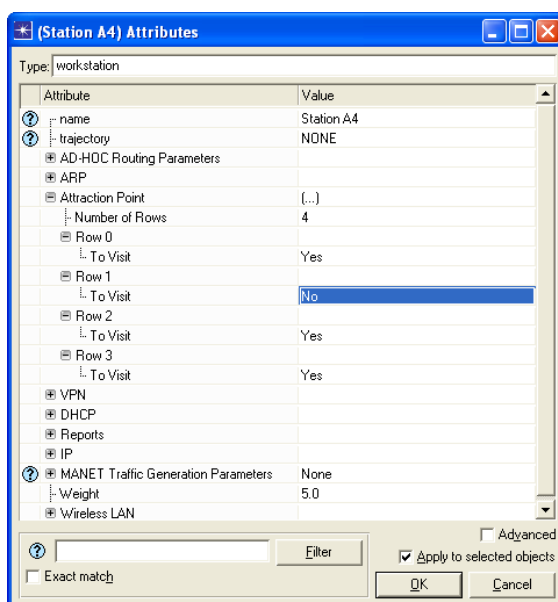
```

Ci-après les trajectoires des groupes A et B :

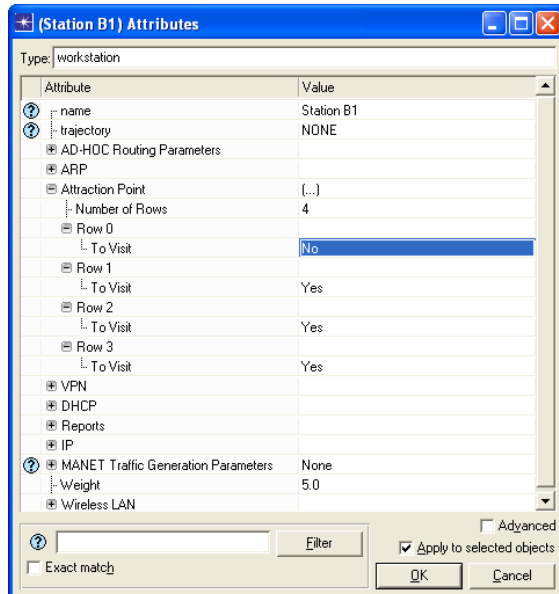


Les lieux où la probabilité est grande pour un nœud de changer ses MPR sont représentés par 2 cercles.

Afin de forcer les groupes à suivre ces trajectoires, il nous suffit de sélectionner les points d'attraction à visiter. Pour le groupe A voici la sélection réalisée dans les stations :

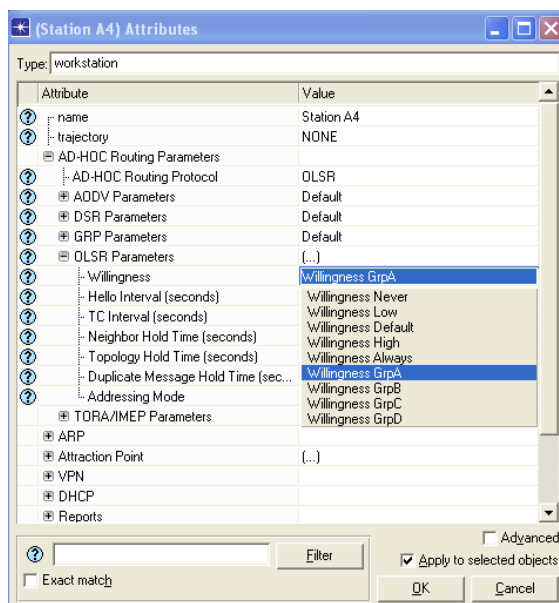


et voici celle du groupe *B* :

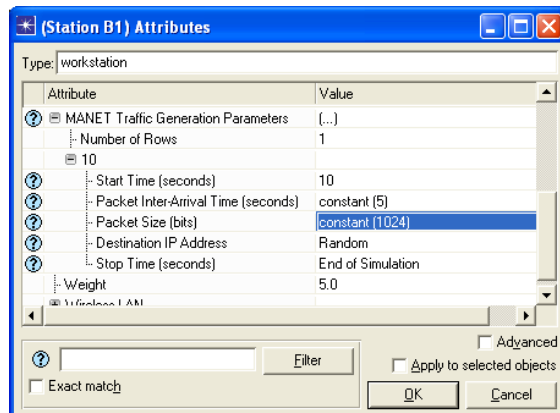


Nous réalisons 2 scénarios :

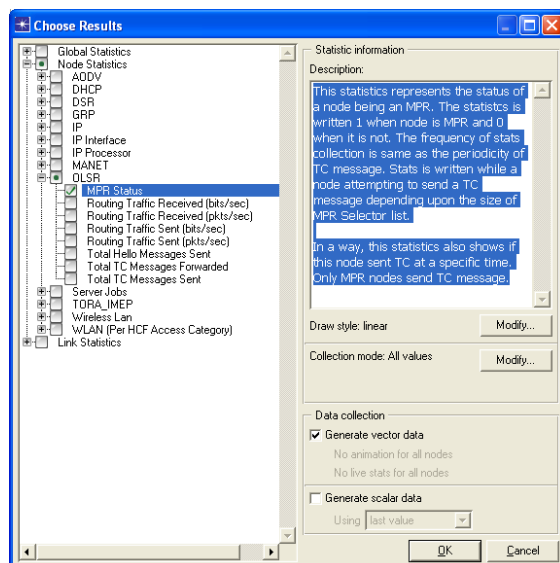
- « Sans OLSR Social » laisse à Will_default le Willingness ;
- « Avec OLSR Social » initialise le Willingness à :
 - Willingness_grpA pour le groupe *A* ;
 - Willingness_grpB pour le groupe *B*.



Il nous faut générer un trafic d'informations. Nous choisissons d'avoir des stations uniformes d'un point de vue fonctionnement (iso-fonctionnelles). Nous paramétrons donc le générateur de trafic uniformément sur toutes les stations comme suit :



Enfin, nous devons choisir nos statistiques *Opnet*. Nous optons pour *OLSR Status* dont voici la rubrique dans l'aide en ligne :



« This statistics represents the status of a node being an MPR. The statistics is written 1 when node is MPR and 0 when it is not. The frequency of stats collection is same as the periodicity of TC message. Stats is written while a node attempting to send a TC message depending upon the size of MPR Selector list. In a way, this statistics also shows if this node sent TC at a specific time. Only MPR nodes send TC message ».

Annexe E

Outil de conversion de trajectoires Opnet vers NS2

E.1 Conversion des Positions de départ

```
#####  
#                                                                 #  
# OUTIL DE CONVERSION DE TRAJECTOIRES OPNET --> NS2           #  
#                                                                 #  
#####  
  
#  
# Définir les constantes  
#  
  
# Nombre de groupes  
$MaxGroupe = 4  
# Description de la position des groupes  
$GroupePos = "bottom left", "bottom right", "top right", "top left"  
# Distribution des groupes indentés (1) ou pas (0)  
$Groupedepl = 0, 1, 1, 0  
# Indicateur de décalage positif (1) ou négatif (0) des secondes lignes de groupe  
$Groupe2ndL = 1, 1, -1, -1
```

E.1. CONVERSION DES POSITIONS DE DÉPART

```
# Nombre de station d'un groupe
$MaxStation = 12
# Incrément sur l'axe des abscisses
$Xincrement = 25
# Incrément sur l'axe des ordonnées
$Yincrement = 50

#
# Ecrire, par groupe, les coordonnées de départ de chaque station sur l'interface de sortie
#

# POUR chaque groupe FAIRE
for ($groupe = 0; $groupe -lt $MaxGroupe; $groupe++) {
# Ecrire le numéro de groupe
"$n# Group $groupe $($GroupePos[$groupe])'n"

# POUR chaque station FAIRE
for ($station = 0; $station -lt $Maxstation; $station++) {
# Déterminer le numéro de la station
$numsta = $station + $MaxStation*$groupe

# Déterminer son abscisse de départ
$x      =      $Xincrement*($station % 6) # Retour Chariot toutes les 6 stations, sinon incrément
$x += 7*$Xincrement*($Groupedepl[$groupe]) # Indentation de 7 incréments interstation pour le

# Déterminer son ordonnée de départ
$y =      $Groupe2ndL[$groupe]*$Yincrement*($station % 2 + 1) # Incrément alternatif : positif
$y += 2*$Yincrement*([math]::floor([int] $station / [int] 6)) # Décalage vers le haut de 2
$y += 10*$Yincrement*([math]::floor([int] $numsta / [int] 24)) # Décalage vers le haut de 10
```

```
# Ecrire les coordonnées de la station
"$node_($numsta) set X_ {0,5:N1}" -f $x # Formatter (-f) le premier champ (0 i.e. $x) sur 5 c
"$node_($numsta) set Y_ {0,5:N1}" -f $y
"$node_($numsta) set Z_ {0,5:N1}" -f 0

# FINPOUR
}

# FINPOUR
}

# Ecrire les coordonnées de la station 48
$x = 100; $y = 300
"$node_(48) set X_ {0,5:N1}" -f $x # Formatter (-f) le premier champ (0 i.e. $x) sur 5 c
"$node_(48) set Y_ {0,5:N1}" -f $y
"$node_(48) set Z_ {0,5:N1}" -f 0

# Ecrire les coordonnées de la station 49
$x = 200; $y = 300
"$node_(49) set X_ {0,5:N1}" -f $x # Formatter (-f) le premier champ (0 i.e. $x) sur 5 c
"$node_(49) set Y_ {0,5:N1}" -f $y
"$node_(49) set Z_ {0,5:N1}" -f 0
```

E.2 Conversion des Trajectoires

```
#####
#
# OUTIL DE CONVERSION DE TRAJECTOIRES OPNET --> NS2
#
#####

# Coordonnées min et max toléré par ns dans un réseau de 300x600
```

E.2. CONVERSION DES TRAJECTOIRES

```
[double]$xmin = 0.000000000001
[double]$xmax = 299.999999999999
[double]$ymin = 0.000000000001
[double]$ymax = 599.999999999999

# Déclarer x et y en tant que double
[double]$x = 0.0
[double]$y = 0.0

# Nombre de stations
$MaxStation = 50

# Déterminer le répertoire de lancement du script
$Source = $(Split-Path $MyInvocation.MyCommand.Path)

# Lancer le script d'initialisation des positions des stations
./OpnetNsInitPos.ps1

# Déterminer le répertoire des trajectoires OPNET
$Source = $Source + "..\Traj\Opnet"

# Éliminer le cas échéant le préfixe projet/scénario/réseau généré par Opnet
# pour ne garder que "Station n" où n est le numéro de station
Get-ChildItem $Source | Rename-Item -NewName '
{ $_.Name -Replace 'acs_Vi-064_40_5_11_2-Office Network.', '' }

for ($NumSta = 1; $NumSta -le $MaxStation; $NumSta++) {
# for ($NumSta = 1; $NumSta -le 1; $NumSta++) {

# Déterminer le nom du fichier trajectoire de la station courante
```

```
$TrajFile = $Source + "\Station $NumSta.trj"
" 'n# Station $($NumSta-1)'n"

# Initialiser la variable temps
$Temps = 0.0
$TempsIncrement = 10 + $NumSta / 10

# Obtenir un descripteur pour le fichier trajectoire
$TrajFileDesc = [System.IO.File]::OpenText($TrajFile)
# ESSAYER et aller dans finally en fin ou sur erreur
$i = 0 # HCO DBG
try {
# POUR chaque ligne du descripteur de fichier
for(;;) {
$TrajLine = $TrajFileDesc.ReadLine()

# SI il s'agit de la fin ALORS sortir de la boucle FINSI
if ($TrajLine -eq $null) { break }
# if ($i -eq 100) { break }

# SI la ligne n'est pas vide ET son premier caractère est un espace ALORS
if ($TrajLine -ne "" -and $TrajLine.SubString(1,1) -eq " ") {
# Extraire l'abscisse de la ligne
$TrajLineClean = $TrajLine.Replace(" ", "").Replace("'t", "")
$posvigule = $TrajLineClean.IndexOf(",")
$x = [double]$TrajLineClean.SubString(0,$posvigule)
if ($x -lt $xmin) { $x = $xmin }
if ($x -gt $xmax) { $x = $xmax }

# Extraire l'ordonnée de la ligne
```

E.2. CONVERSION DES TRAJECTOIRES

```
$TrajLineClean = $TrajLineClean.SubString($posvigule + 1)
$posvigule = $TrajLineClean.IndexOf(",")
$y = [double]$TrajLineClean.SubString(0,$posvigule)
if ($y -lt $ymin) { $y = $ymin }
if ($y -gt $ymax) { $y = $ymax }

# Écrire la ligne de mouvement sur l'interface de sortie
"$ns_ at {0:F12} ""$node_($($NumSta-1)) setdest {1:F12} {2:F12} 0.000000000000"" -f $temps

# Incréments la variable temps
$Temps += $TempsIncrement

$i++ # HCO DBG
# FINSI
}
# FINPOUR
}
# FIN ESSAYER
}
finally {
$TrajFileDesc.Close()
}
}
```


Résumé

L'avenir de la communication est perçu comme étant quasiment exclusivement constitué de *nœuds mobiles* évoluant dans un *réseau mobile*. Dans ce contexte, différentes approches contribuent continuellement à l'amélioration directe ou indirecte des délais d'acheminement des informations échangées entre les utilisateurs, dont : l'analyse des traces, l'évaluation des performances, les services de localisation, le routage. Nous décrivons chacun de ses thèmes et proposons des solutions faisant évoluer l'état de l'art. Celles-ci prennent appui sur des méthodes et outils tels que :

- les Réseaux de Petri, pour l'analyse des traces ;
- les modèles de mobilité, pour l'évaluation des performances ;
- l'introduction du social dans les services de localisation ;
- la mise en place d'une nouvelle métrique pour le routage.

Nous montrons comment ces solutions concourent de façon complémentaire les unes avec les autres, à améliorer l'expérience de l'utilisateur.

Mots clés

Réseaux sans fil, analyse de traces, modèles de mobilité, protocoles de routage, réseaux de Pétri stochastiques, chaînes de Markov, aggrégation de données, métrique, traitement du signal.

Abstract

Future of communication is perceived as being almost exclusively composed of *mobile nodes* operating in a *mobile network*. In this context, different approaches contribute to continually improve directly or indirectly, delivery times of information exchanged between users, including :

- Trace analysis,
- Performance evaluation,
- Location services,
- Routing.

We describe each of these topics and propose solutions by changing the state of the art. These are supported by tools and methods such as :

- Petri Nets for Trace analysis,
- Mobility Models for Performance evaluation,
- Social additions in Location services,
- The establishment of metrics for Routing.

We show how these solutions work together in a complementary manner with each other, to improve the user experience.

Keywords

Wireless networks, trace patterns, mobility models, performance evaluation, routing protocols, Petri nets, markov chains, metrics, signal processing.