

LA REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

Université MOULOUD MAMMERI de Tizi Ouzou

Faculté de génie électrique et d'informatique

Département d'informatique

Spécialité : Systèmes Informatiques

Pour l'obtention d'un diplôme de master en informatique

Thème intitulé :

Implémentation des modèles de mobilité basé sur l'activité et Downtown sous J-Sim

Encadré par : M^r Mehammed Daoui

Présenté par : M^{elle} Boukhoufane Akila

M^{elle} Hadj-said Lynda

épouse Belkadi

2010-2011

Remerciements

Nous tenons tout d'abord à exprimer nos sincères remerciements à notre encadreur Mr.Mehamed DAOUI. Pour son aide sans limite et ses précieux conseils qu'il nous a donnés et sans lesquels ce travail n'aurait pas vu le jour. Il nous a ainsi offert l'opportunité de faire nos premiers pas de scientifiques sur un travail de recherche passionnant et prometteur.

Nous remercions les membres de jury qui ont accepté de juger ce travail.

Nous remercions également tous ceux qui de près ou de loin, nous ont accompagnés et soutenus pour mener à bien ce travail

Dédicaces

À mes parents,

À mes frères et sœurs,

À ma belle sœur,

À mon beau frère,

À mes neveux,

À tous ceux qui me sont chers,...

Je dédie affectueusement ce modeste travail

Akila

Dédicaces

À mes parents,

À mon mari,

À mon ange Mailis,

À mes frères,

À mes beaux parents,

À tous ceux qui me sont chers,...

Je dédie affectueusement ce modeste travail

Lynda

Résumé :

Les réseaux ad hoc véhiculaires (VANETs) sont considérés comme une solution adéquate à la conduite coopérative entre les voitures communicantes sur la route. Le déploiement et le test de ces réseaux dans la réalité implique un coût élevé, ainsi la simulation est une alternative utile dans la recherche. Une des issues les plus critiques dans une étude de simulation de VANETs est l'utilisation d'un modèle de mobilité qui ressemble, aussi étroitement que possible au vrai comportement du trafic véhiculaire. Les modèles de mobilité sont cruciaux pour obtenir des résultats précis et significatifs dans la simulation.

Dans notre travail nous présentons les deux modèles de mobilité : le modèle Downtown de CityMob, qui est un générateur de modèle de mobilité pour VANETs, et le modèle basé sur l'activité qui est à l'origine conçu pour les réseaux cellulaires. Afin de rendre ces modèles plus représentatifs de la réalité des déplacements des réseaux véhiculaires nous leur avons apporté quelques améliorations.

Nous avons réalisé la simulation de ces deux modèles avec le simulateur J-SIM qui est dédié aux réseaux de capteurs.

Mots clés :

Modèle de mobilité, VANET, MANET, modèle de mobilité Downtown, modèle de mobilité basé sur l'activité, J-Sim.

Abstract:

Vehicular Ad hoc Networks (VANETs) are regarded as an adequate solution to cooperative driving between communicating cars on the road. Deploying and testing these networks involves a high cost in the real world, and so simulation is an useful alternative in research. One of the most critical issues in a simulation study of VANETs is the use of a mobility model which resembles, as closely as possible, the real behavior of vehicular traffic. The mobility models are crucial to obtain precise and significant results in simulation.

In our work we present two mobility models: the Downtown mobility model of CityMob which is the generator of mobility model for VANETs and the activity based model which is conceived for cellular networks. In order to make these models nearer to reality we have add some improvements to them. We have realized the simulation of these two models with the simulator J-SIM which is dedicated to sensor's networks.

Key words:

Mobility model, VANET, MANET, activity base mobility model, Downtown model, J-Sim

Table des matières

Résumé	i
Table des matières	ii
Liste des figures	vii
Liste des tables	x
Liste des acronymes	xi
Introduction générale	1
CHAPITRE I : GENERALITES SUR LES RESEAUX SANS FIL	
I.1 Introduction.....	4
I.2 Définition d'un réseau sans fil	4
I.2.1 Réseaux personnels sans fil (WPAN)	5
I.2.2 Réseaux locaux sans fil (WLAN)	5
I.2.3 Réseaux métropolitains sans fil (WMAN)	5
I.2.4 Réseaux étendus sans fil (WWAN)	5
I.3 Mode opératoires des réseaux sans fil	6
I.3.1 Mode infrastructure	6
I.3.1.1 La communication avec le point d'accès	7
I.3.2 Mode ad hoc	8
I.4 Les réseaux Ad hoc sans fil	9
I.4.1 Définition d'un réseau ad hoc :.....	9
I.4.2 Caractéristiques des réseaux Ad Hoc :.....	10
I.4.3 Domaines d'utilisation des réseaux Ad hoc :.....	11
I.5 Les réseaux de capteurs sans fils :.....	12
I.5.1 Définition:	12
I.5.2 Domaines d'applications des RCSF :.....	13
I.5.3 Caractéristiques des RCSF:.....	15
I.6 Les réseaux véhiculaires(VANET) :.....	16
I.6.1 Les modes de communication dans les réseaux VANET :.....	17

I.6.1.1 Mode de communication Véhicule-à-Véhicule (V2V) :.....	17
I.6.1.2 Mode de communication de Véhicule à Infrastructure (V2I) :.....	18
I.6.2 Les caractéristiques des VANETs :.....	19
I.6.3 Applications des réseaux VANETs :.....	19
I.6.4 Travaux dans le domaine des VANETs :	23
I.6.5 Mobilité des nœuds dans un VANET :	24
I.7 Conclusion :.....	25
 CHAPITRE II : TAXONOMIE DE MODELES DE MOBILITE	
II.1. Introduction :.....	27
II.2 Définition d'un modèle de mobilité :.....	27
II.3. Les modèles de mobilité MANET :.....	27
II.2.1 Modèles basés sur le déplacement individuel :.....	28
II.3.1.1 Random Mobility Model :.....	28
II.3.1.2 Random WayPoint :.....	29
II.3.1.3 Random Direction :.....	31
II.3.1.4 Boundless Simulation Area :.....	32
II.3.1.5 Smooth Random Mobility Model :.....	32
II.2.2 Modèles basés sur le déplacement en groupe :.....	34
2.2.1 Reference Point Group Model (RPGM):.....	34
2.2.2 Nomadic Mobility Model :.....	35
2.2.3 Pursue Mobility Model :.....	35
2.2.4 Column Mobility Model:.....	36
I.3 Les modèles de mobilité VANETs :.....	37
II.3.1 Freeway :.....	37
II.3.2 Manhattan :.....	38

II.3.3 City Section Mobility Model :.....	39
II.3.4 STRAW :.....	40
II.3.5 Stop Sign Model :.....	41
II.3.6 Traffic Sign Model :.....	42
II.4 Activity- based Model:.....	43
II.4.1 Time and money allocation model:.....	43
II.4.2 Constraint-based Models :.....	44
II.4.3 Utility-based models (discrete choice model):.....	44
II.4.4 Micro-Simulation Model	45
II.5 Conclusion :.....	45
 CHAPITRE III PRESENTATION DES MODELES A IMPLEMENTER	
III.1 Introduction	47
III.2 Conception d'un modèle de mobilité	47
III.2.1 Générateur de déplacements (ou de trafic)	47
III.2.2. Distribution des déplacements	48
III.2.3. Répartition sur les modes de transport (ou Modal split)	48
III.2.4. Affectation des déplacements	48
III.3 Description des modèles à implémenter.....	49
III.3.1 Le modèle basé sur l'activité :.....	49
III.3.1.1 Les données d'entrées :.....	51
III.3.1.2 Matrice de transition d'activité :.....	52
III.3.1.3 La matrice des durées :.....	53
III.3.1.4 La matrice de la plus courte distance :.....	54
III.3.1.5 Le fichier de plus court chemin :	55

III.3.1.6 Déplacement géographique.....	55
III.3.1.7. Informations sur les utilisateurs.....	56
III.3.2 Le modèle Downtown :.....	57
III.4 Critiques des modèles de mobilité étudiés :.....	58
III.4.1 Critiques de modèle de mobilité basé sur l'activité :.....	58
III.4.2 Critiques de modèle de mobilité Downtown :.....	59
III.5 Amélioration des deux modèles :.....	59
III.5.1 Amélioration du modèle d'activité :.....	59
III.5.2 Amélioration du modèle Downtown :.....	59
III.6 Conclusion	62

CHAPITRE IV : IMPLEMENTATION DES MODELES SOUS J-SIM

IV.1 Introduction :.....	64
IV.2 Environnement de la simulation :.....	64
IV.2.1 Le simulateur J-SIM :.....	64
IV.2.1.1 Architecture de J-Sim:	65
IV.2.1.2 Langage utilisé pour définir un scénario de simulation :.....	66
IV.2.1.3 Modèle de transmission dans J-Sim :.....	67
IV.2.1.4 Installation de J-Sim :.....	68
IV.2.1.5 Intégration du code source du protocole dans J-SIM :.....	68
IV.2.1.6 Connexion entre les composants :.....	68
IV.2.1.7 Implémentation d'un modèle dans J-Sim:.....	69
IV.3 Implémentation des modèles étudiés :.....	69
IV.3.1 Le modèle basé sur l'activité.....	69
IV.3.1.1 L'événement de génération de déplacement :.....	70
IV.3.1.2 La description des classes :.....	72
IV.3.2 Le modèle Downtown.....	76

IV.3.2.1 L'événement de génération de déplacement :.....	76
IV.3.2.2 La description des classes :.....	78
IV.4 Exécution et simulation :.....	81
IV.5 Tests des modèles de mobilité	82
IV.5.1. Le modèle basé sur l'activité :.....	82
IV.5.1.1 Les paramètres de la simulation :.....	82
IV.5.1.2 Les résultats de la simulation	82
IV.5.1.3 Description de l'événement	83
IV.5.2. Le modèle Downtown :	84
IV.5.1.1 Les paramètres de la simulation :	84
IV.5.1.2 Les résultats de la simulation	85
IV.5.1.3 Description de l'événement	88
IV.6 Conclusion :.....	88
CONCLUSION GENERALE.....	90
BIBLIOGRAPHIE :.....	92
ANNEXE :.....	97

Liste des figures

Chapitre I

Figure I.1	Le classement des réseaux sans fil selon l'étendue.....	6
Figure I. 2	Mode infrastructure.....	7
Figure I. 3.	Exemple d'un ensemble de services étendu.....	7
Figure I. 4.	Mode Ad hoc.....	8
Figure I.5	La modélisation d'un réseau Ad Hoc.....	9
Figure I.6	Le changement de la topologie des réseaux Ad Hoc.....	10
Figure I. 7	Exemple de réseau de capteurs sans fil.....	12
Figure I.8	Réseau véhiculaire sans fil.....	17
Figure I.9	Exemple de réseau C.....	18
Figure I.10	Exemple de réseau V2I.....	18
Figure I.11	Véhicule en panne.....	20
Figure I.12	Travaux sur la route.....	21
Figure I.13	Risque de collision.....	21
Figure I.14	Accès à internet.....	22
Figure I.15	Parking intelligent.....	23

Chapitre II

Figure II. 1	Le modèle Random Mobility Model.....	28
Figure II. 2	Le modèle Random WayPoint.....	30
Figure II.3	Le modèle Random Direction.....	31
Figure II.4	Boundless Simulation Area Model.....	32
Figure II. 5	Mouvements de trois MNs utilisant le modèle RPGM.....	34
Figure II.6	Mouvements des nœuds utilisant le modèle Nomadic Mobility Model.....	35
Figure II.7	Mouvements des nœuds utilisant le modèle Pursue Mobility Model.....	36
Figure II.8	Mouvements des nœuds utilisant le modèle Column Mobility Model.....	37

Figure II.9	Le modèle Freeway.....	38
Figure II.10	Déplacements dans le modèle Manhattan.....	39
Figure II.11	City Section Mobility Model.....	40
Figure II.12	Classification de modèles bases sur l'activité.....	43

Chapitre III

Figure III.1	La décomposition de la zone simulée en cellules.....	56
Figure III.2	exemple de scenario dans le centre ville.....	58
Figure III.3	Le fichier d'activités.....	61

Chapitre IV

Figure IV .1	Connexions entre composants dans J-Sim.....	66
Figure IV.2	Champs de voisinage dans J-Sim.....	67
Figure IV.3	Fonctionnement du modèle d'activité.....	71
Figure IV.4	Schéma fonctionnel de la classe EventTraversZone.....	72
Figure IV.5	Schéma fonctionnel de la classe ZoneGeography.....	73
Figure IV.6	Schéma fonctionnel de la classe Mouvement.....	74
Figure IV.7	Le schéma fonctionnel de la classe Mobilité.....	75
Figure IV.8	Schéma fonctionnel de modèle Downtown.....	77
Figure IV.9	Schéma fonctionnel de la classe Control.....	78
Figure IV.10	Schéma fonctionnel de la classe Simulation.....	78
Figure IV.11	Schéma fonctionnel de la classe Vstreet/Hstreet.....	79
Figure IV.12	Schéma fonctionnel de la classe Node.....	80
Figure IV.13	Schéma global de la classe Mobility.....	81

Figure IV.14 : Représentation de l'itinéraire de l'utilisateur numéro (2).....	83
Figure IV.15: Représentation de l'itinéraire de l'utilisateur numéro (9).....	87

Liste des acronymes

ACA	Autonomous Component Architecture
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
CSM	City Section Model
DRCL	Distributed Realtime Computing Laboratory
DS	Distribution System
ESS	Extended Service Set
ESSID	Extended Service Set Identifier
GPS	Global Positioning System
IBSS	Independent Basic Service Set
ITS	Intelligent transportation System
JVM	Java Virtual Machine
MANET	Mobile Ad hoc Network
QoS	Quality of Service
RCSF	Réseau de Capteurs Sans Fils
RMM	Random Mobility Model
RPGM	Reference Point Group Model
RSU	Road Side Units
RUV	RUntime Virtual
RWP	Random WayPoint
SSM	Stop Sign Model
STRAW	Street Random Waypoint
TCL	Tool Command Line
TSM	Traffic Sign Model

VANET	Vehicular Ad hoc Network
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
WPAN	Wireless Personal Area Network
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WWAN	Wireless Wide Area Network

Le domaine des réseaux et de la communication est l'un des domaines clés de l'informatique. Ce domaine n'a cessé d'évoluer ces dernières années pour répondre aux besoins de l'homme. Les réseaux sont utilisés dans la vie de tous les jours comme outil de travail ou de loisir (partage de données, internet, jeux ...). Avec l'adoption des technologies sans fil, de nouvelles utilisations et perspectives ont vu le jour. L'interface filaire a cédé sa place aux technologies sans fil, permettant ainsi à des calculateurs une communication plus souple et moins contraignante. Les calculateurs utilisés dans ce type de réseau sont, soit des PC munis d'une interface réseaux sans fil, ou bien des calculateurs légers dont la capacité de stockage et l'autonomie sont limitées (PDA, téléphones portables, ordinateurs portables...). Ces derniers ont fait naître un nouvel environnement appelé environnement mobile, qui permet aux unités du réseau une libre mobilité.

Les réseaux sans fil peuvent être classés en deux catégories, les réseaux avec infrastructure, qui obéissent à une architecture client/serveur, et les réseaux sans infrastructure, appelés aussi réseaux Ad hoc, qui reposent sur la collaboration des unités sans fil du réseau pour l'établissement et la maintenance du réseau. Le premier type est fiable, mais il nécessite le déploiement d'une importante infrastructure logistique et matérielle fixe. Le deuxième type, bien qu'il soit moins fiable que le premier, permet une plus grande liberté dans la communication.

Les réseaux Ad hoc peuvent être utilisés à des endroits où l'installation d'infrastructures est trop coûteuse ou impossible. L'une de leurs caractéristiques clés est leur topologie très dynamique, du fait de la mobilité des unités du réseau. Ils ont une large gamme d'applications, dont les réseaux ad hoc véhiculaire (VANETs, pour Vehicular Ad hoc NETwork). Ce type de réseaux fait communiquer des véhicules entre eux via des interfaces sans fil.

La conduite nécessite un besoin constant d'informations sur l'environnement extérieur du conducteur. Ces informations peuvent renforcer la prévention routière (informations sur le trafic, état de la chaussée, météo...), simplifier les voyages (emplacement des stations de ravitaillement en carburant, emplacement des bouchons, prix du carburant...) et améliorer le confort des passager (accès à internet, jeux interactifs entre les passagers de voitures proches, service de chat...). Ces informations peuvent être partagées grâce aux VANETs, afin de créer un système collaboratif de prévention et de communication.

Afin d'étudier les VANETs, le déploiement sur terrain réel n'est, malheureusement pas aisé, d'où le recours à la simulation. Plusieurs simulateurs ont été mis à la disposition des chercheurs dans ce but (GloMoSim, NS-2, J-SIM...). Lors d'une simulation, la mobilité est un paramètre à ne pas négliger, car les unités dans un VANET peuvent se déplacer suivant un schéma de mobilité particulier. Pour modéliser la mobilité des VANETs, plusieurs modèles ont été conçus. Ces modèles de mobilité doivent prendre en considération les contraintes de la mobilité véhiculaire, pour que la simulation soit proche de la réalité. L'objectif principal de notre travail, est l'implémentation de deux modèles de mobilité qui seront utilisés pour la simulation des réseaux véhiculaires. Ces modèles peuvent prendre en compte l'impact de la mobilité réelle des usagers lors de la conception de protocoles spécifiques aux VANETs.

Pour mener à bien notre travail, nous l'avons organisé en quatre chapitres selon le plan suivant :

Le premier chapitre porte sur les généralités des réseaux sans fil. Nous y parlerons des réseaux sans fil en général, ensuite des réseaux mobiles ad hoc, pour enfin aboutir aux réseaux ad hoc véhiculaires. Le deuxième chapitre porte sur la modélisation de la mobilité. Nous y parlerons des modèles de mobilité généraux, utilisés dans le contexte des réseaux ad hoc, puis des modèles de mobilité dédiés aux réseaux véhiculaires. Le troisième chapitre comporte la description des deux modèles à implémenter et le quatrième chapitre comporte la phase d'intégration des modèles dans le simulateur J-SIM dédié aux réseaux de capteurs.

I.1 Introduction :

L'outil informatique n'a cessé d'évoluer ces dernières années dans divers domaines, notamment dans le domaine de la communication. Depuis leur apparition, les réseaux sont devenus de plus en plus rapides et performants.

Les premiers chaînons de cette évolution sont les réseaux filaires, qui sont encombrants et rigides. Les réseaux ont ensuite gagné en liberté avec l'adoption des technologies sans fil, offrant plus de souplesse, plus de rapidité et moins de frais. Ensuite, les réseaux informatiques sont entrés dans l'ère de la mobilité. De nouvelles technologies ont vu le jour, permettant aux unités du réseau de se déplacer librement, tout en étant connectées entre elles. Avec la prolifération des appareils légers (PDA, ordinateurs portables, téléphones mobiles...) la communication par réseaux mobiles est devenue plus efficace et plus accessible. Ces réseaux ont trouvé une large gamme d'applications, citons par exemple, la téléphonie, le partage rapide de données et la sécurité routière.

Dans ce chapitre nous allons donner des notions générales sur les réseaux sans fil. Nous parlerons des deux grandes classes de réseaux sans fil : les réseaux avec infrastructure et ceux sans infrastructure (ad hoc). Ensuite, nous aborderons les réseaux mobiles ad hoc (MANETs) pour enfin parler des réseaux ad hoc véhiculaires (VANETs).

I.2 Définition d'un réseau sans fil :

Un réseau sans fil est un réseau dans lequel deux terminaux, au minimum, peuvent communiquer sans utiliser de liaison filaire. Un réseau sans fil offre la possibilité de se déplacer, tout en restant connecté, dans un périmètre géographique plus ou moins étendu. C'est pourquoi l'on parle de «mobilité».

Les réseaux sans fil utilisent donc les ondes radioélectriques comme mode de transmission, ils permettent donc de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres.

Plusieurs technologies existent qui se distinguent par la fréquence d'émission utilisée, le débit et la portée des transmissions.

C'est d'ailleurs cette dernière caractéristique qui permet de classer les réseaux sans fil, à l'instar des réseaux filaires, selon 4 grandes classes. Celles-ci sont donc définies en fonctions

du périmètre géographique offrant une connectivité, plus communément appelées zone de couverture.

2.1 Réseaux personnels sans fil (WPAN) :

Le réseau personnel sans fil, noté WPAN pour Wireless Personal Area Network, concerne les réseaux sans fil d'une faible portée : de l'ordre de quelques dizaines de mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fil entre deux machines très peu distantes.

2.2 Réseaux locaux sans fil (WLAN) :

Le réseau local sans fil, noté WLAN pour Wireless Local Area Network est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier les terminaux présents dans la zone de couverture.

2.3 Réseaux métropolitains sans fil (WMAN) :

Un WMAN est un réseau sans fil qui se tisse sur une ville, permettant à tous les habitants d'être connectés entre eux. La portée peut dépasser les 50 km. Cette technologie est destinée principalement aux opérateurs de télécommunication.

2.4 Réseaux étendus sans fil (WWAN) :

Le réseau étendu sans fil, WWAN pour Wireless Wide Area Network, est également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil.

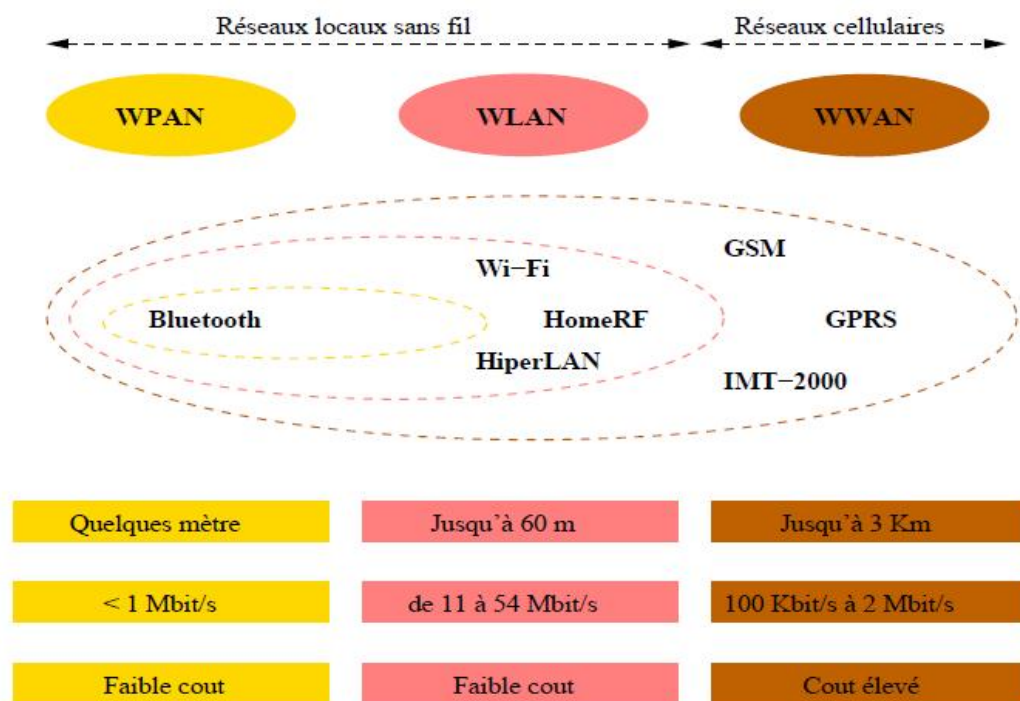


Figure I.1 : le classement des réseaux sans fil selon l'étendue

I.3 Mode opératoires des réseaux sans fil :

Nous distinguons deux modes opératoires dans les réseaux sans fil : le mode avec infrastructure et le mode sans infrastructure.

I.3.1. Mode infrastructure

En mode infrastructure chaque station se connecte à un point d'accès via une liaison sans fil. L'ensemble formé par le point d'accès et les stations situés dans sa zone de couverture est appelé ensemble de services de base (en anglais basic service set, noté BSS) et constitue une cellule. Chaque BSS est identifié par un BSSID. Dans le mode infrastructure, le BSSID correspond à l'adresse MAC du point d'accès.

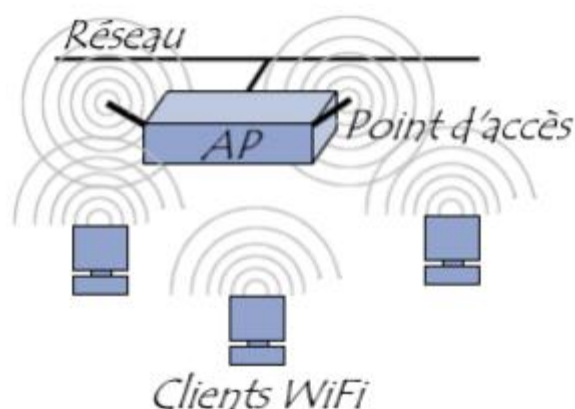


Figure I.2: Mode infrastructure.

Il est possible de relier plusieurs points d'accès entre eux (ou plus exactement plusieurs BSS) par une liaison appelée système de distribution (notée DS pour Distribution System) afin de constituer un ensemble de services étendu (extended service set ou ESS). Le système de distribution (DS) peut être aussi bien un réseau filaire, qu'un câble entre deux points d'accès ou bien même un réseau sans fil.

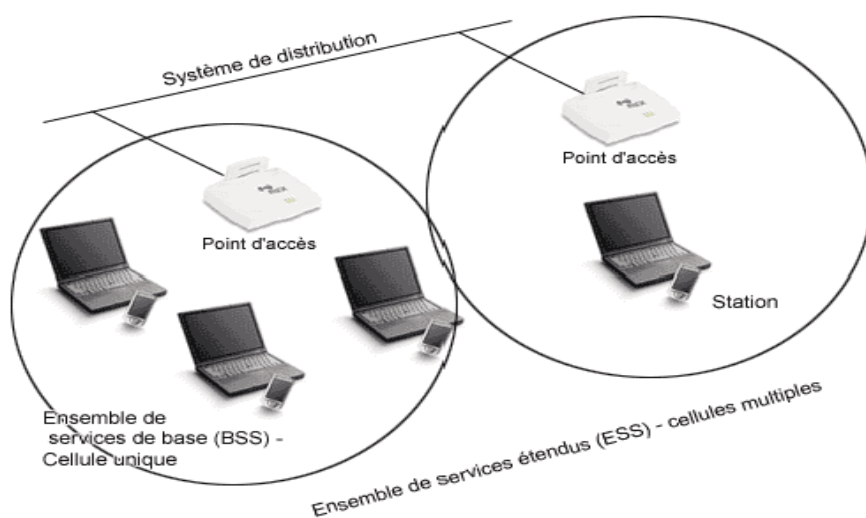


Figure I.3 : Exemple d'un ensemble de services étendu.

3.1.1 La communication avec le point d'accès :

Lors de l'entrée d'une station dans une cellule, celle-ci diffuse sur chaque canal une requête de sondage (probe request) contenant l'ESSID (identifiant de réseau) pour lequel elle est configurée ainsi que les débits que son adaptateur sans fil supporte. Si aucun ESSID n'est configuré, la station écoute le réseau à la recherche d'un ESSID.

En effet chaque point d'accès diffuse régulièrement (à raison d'un envoi toutes les 0.1 secondes environ) une trame balise donnant des informations sur son BSSID, ses caractéristiques et éventuellement son ESSID. L'ESSID est automatiquement diffusé par défaut.

A chaque requête de sondage reçue, le point d'accès vérifie l'ESSID et la demande de débit présent dans la trame balise. Si l'ESSID correspond à celui du point d'accès, ce dernier envoie une réponse contenant des informations sur sa charge. La station recevant la réponse peut ainsi constater la qualité du signal émis par le point d'accès afin de définir la distance qui les sépare. En effet d'une manière générale, plus un point d'accès est proche, meilleur est le débit.

I.3.2. Mode ad hoc :

En mode ad hoc les machines sans fils clientes se connectent les unes aux autres afin de constituer un réseau point à point (peer to peer en anglais), c'est-à-dire un réseau dans lequel chaque machine joue en même temps le rôle de client et le rôle de point d'accès. L'ensemble formé par les différentes stations est appelé ensemble de services de base indépendants (en anglais indépendant basic service set, abrégé en IBSS).

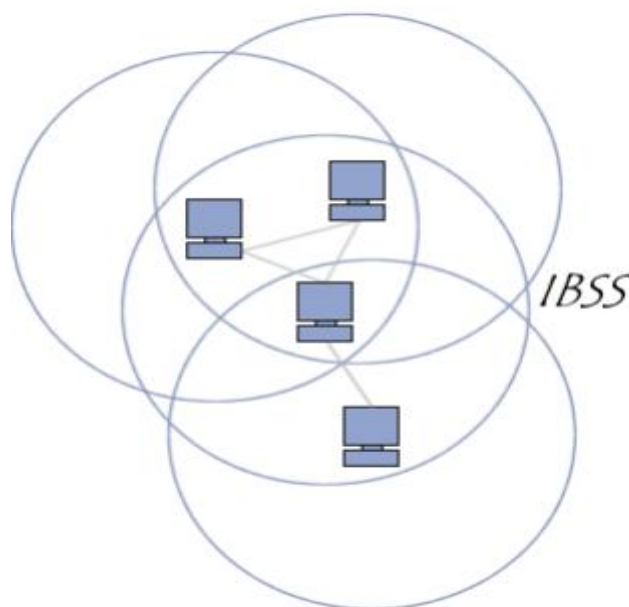


Figure I.4: Mode Ad hoc.

Un IBSS est ainsi un réseau sans fil constitué au minimum de deux stations et n'utilisant pas de point d'accès. L'IBSS constitue donc un réseau éphémère permettant à des personnes

situées dans une même salle d'échanger des données. Il est identifié par un SSID, comme l'est un ESS en mode infrastructure.

I.4 Les réseaux Ad hoc sans fil :

4.1 Définition d'un réseau ad hoc :

Un réseau mobile ad hoc [1], appelé généralement MANET (Mobile Ad hoc Network), consiste en une grande population, relativement dense, d'unités mobiles qui se déplacent dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée. Un réseau ad hoc peut être modélisé par un graphe $G_t = (V_t, E_t)$ où V_t représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau et E_t modélise l'ensemble les connexions qui existent entre ces nœuds (voir la figure I.4). Si $e = (u,v) \in E_t$, cela veut dire que les nœuds u et v sont en mesure de communiquer directement à l'instant t .

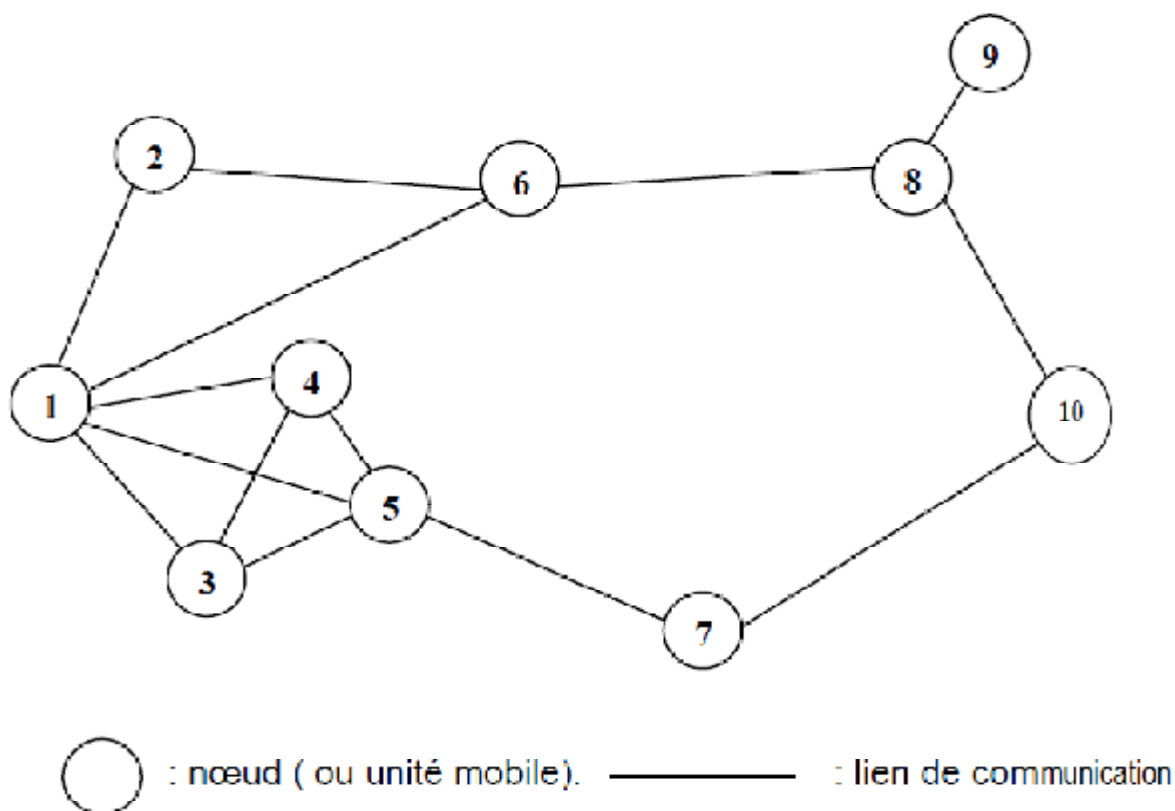


Figure I.5 : La modélisation d'un réseau Ad Hoc

La topologie du réseau peut changer à tout moment (voir la figure I.6), elle est donc dynamique et imprévisible ce qui fait que la déconnexion des unités soit très fréquente.

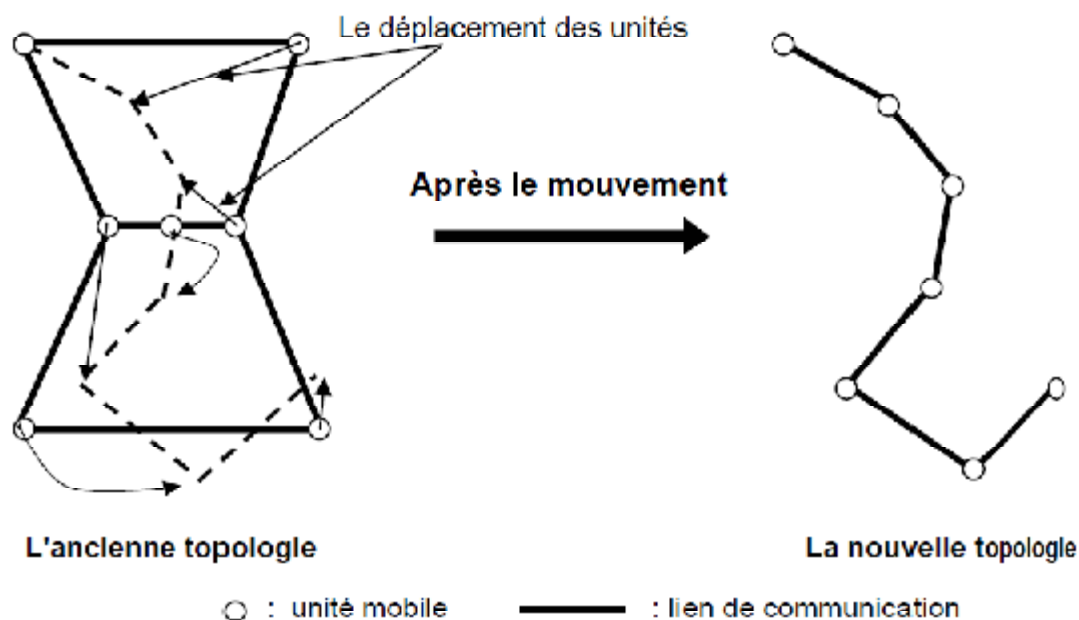


Figure I.6 : Le changement de la topologie des réseaux Ad Hoc

4.2 Caractéristiques des réseaux Ad Hoc :

Les réseaux Ad hoc sont principalement caractérisés par [1] [2] :

- Ø **Des contraintes d'énergie** : Les hôtes mobiles sont alimentés par des sources d'énergie autonomes comme les batteries ou les autres sources consommables. Le paramètre d'énergie doit être pris en considération dans tout contrôle fait par le système.
- Ø **Une bande passante limitée** : Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un média de communication partagé. Ce partage fait que la bande passante réservée à un hôte soit modeste.
- Ø **Une topologie dynamique** : Les unités mobiles du réseau, se déplacent d'une façon libre et arbitraire. Par conséquent la topologie du réseau peut changer, à des instants imprévisibles, d'une manière rapide et aléatoire. Les liens de la topologie peuvent être unis ou bidirectionnels.

- Ø **Une sécurité physique limitée** : Etant basés sur les communications sans fil, les réseaux Ad hoc sont plus sensibles aux attaques qui menacent les données transmises. De plus, les techniques conventionnelles utilisées pour faire face à ces attaques ne sont plus applicables dans les réseaux ad hoc à cause des limitations de ressources connues dans ce type de réseau (puissance de calcul et mémoire).
- Ø **Absence d'infrastructure** : Les réseaux mobiles Ad hoc se caractérisent par l'absence d'une infrastructure préexistante et de tout genre d'administration centralisée, ce qui induit que les unités mobiles sont responsables de l'établissement et du maintien de la connectivité du réseau de façon continue.
- Ø **Contraintes de performances** : dans un réseau mobile ad hoc, les unités mobiles ont, en général, des performances et des capacités de stockage limitées.
- Ø **Interférences** : Les liens radios ne sont pas isolés, deux transmissions simultanées sur une même fréquence ou utilisant des fréquences proches peuvent interférer.

4.3 Domaines d'utilisation des réseaux Ad hoc :

Les réseaux mobiles ad hoc ont une très large palette d'utilisations. En effet, ils sont robustes, peu coûteux et s'adaptent aussi bien aux milieux urbains, qu'aux milieux ruraux. Parmi leurs utilisations nous citerons :

- Ø **Les applications militaires** : Un réseau mobile Ad hoc est la solution idéale pour maintenir la liaison entre des chars d'assauts, des avions de chasse ou même entre les soldats et leurs supérieurs au cours des exercices militaires ou dans un champ de bataille.
- Ø **Opérations de secours** : Dans une région touchée par une catastrophe naturelle (cyclone ou séisme), il est impossible d'installer des infrastructures fixes, d'où la nécessité de déployer un réseau mobile Ad hoc. La rapidité du déploiement de ce type de réseaux les rend idéaux pour ces situations.
- Ø **Utilisation privée** : Ce type de réseau peut facilement être utilisé à l'intérieur d'une maison (réseau appelé *home network*), où un ensemble de robots ou bien d'équipements (audio/vidéo, alarme...) peuvent s'échanger des informations.
- Ø **Utilisation à des fins éducatives** : Il se peut, lors d'une conférence ou d'une séance de cours, que les chercheurs ou les étudiants aient besoin de partager des données. Ce type de réseau serait une bonne solution pour communiquer dans cette situation.

Ø Mise en œuvre des réseaux véhiculaires : (Vehicular Ad hoc NETwork):

Il se peut que des véhicules aient besoin de communiquer, soit entre eux, soit avec leur environnement, pour partager des informations ou par simple loisir. Etant donné que les véhicules circulent à grandes vitesses, il ne serait pas aisé d'établir entre eux un réseau avec infrastructure. La solution ad hoc est donc la plus judicieuse.

I.5 Les réseaux de capteurs sans fils :

5.1 Définition:

Les réseaux de capteurs sont souvent composés d'un nombre très important de nœuds, appelés capteurs. Ces nœuds sont des entités capables d'opérer en toute autonomie afin de collecter, traiter et envoyer les données relatives à leur environnement. Déployés de façon à couvrir un territoire donné, les capteurs communiquent par radio afin de concentrer l'information sur une station collectrice située au cœur ou en bordure du territoire, appelée nœud « Sink » [4,5,6]. Ces entités doivent être équipées d'une batterie qui est une source d'énergie non durable, ce qui présente la contrainte la plus gênante pour la survie d'un tel réseau. Les réseaux de capteur ne définissent pas une architecture préexistante puisque les capteurs peuvent se trouver fixés à un endroit précis ou mobiles, formant ainsi un réseau sans infrastructure prédéfinie. Un tel réseau ne se limite pas à un domaine particulier mais il peut s'adresser à une diversité de secteurs [4,5,6,7] comme la biologie, la chimie, l'environnement, ainsi que la surveillance sismique et même la télésurveillance personnelle...

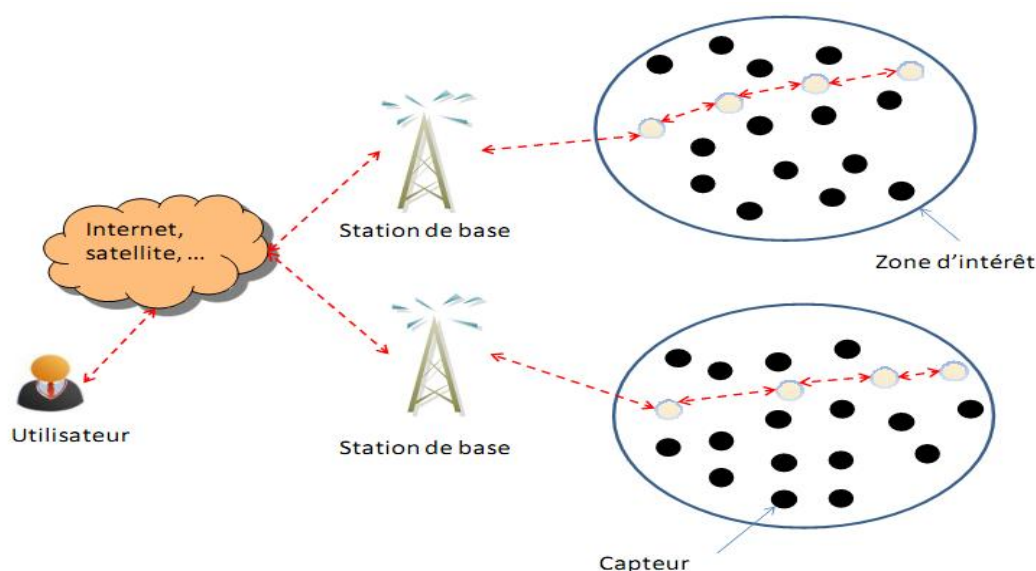


Figure I.7 : Exemple de réseau de capteurs sans fil.

5.2 Domaines d'applications des RCSFs :

La taille de plus en plus réduite des micro-capteurs, leur coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, de vibrations, etc.) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir rapidement plusieurs domaines d'applications. En effet, ces réseaux se révèlent très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi ces domaines, nous citons les domaines : militaire, environnemental, écologique, domestique, de santé, de sécurité, etc. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous.

Ø Applications militaires:

Comme pour de nombreuses autres technologies, le domaine militaire a été le moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Actuellement, les RCSFs peuvent être une partie intégrante dans le commandement, le contrôle, la communication, la surveillance, la reconnaissance, etc.

Ø Applications médicales:

Le champ de contrôle de santé (Health monitoring) représente un grand marché pour les réseaux de capteurs sans fil qui a tendance à croître très rapidement. Ces réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain (surveillance de la glycémie, détection de cancers, ..) grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau. Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, le rythme cardiaque, ... à l'aide des capteurs ayant chacun une tâche bien particulière.

Ø Applications architecturales :

Transformation des bâtiments en environnements intelligents capables de reconnaître des personnes, interpréter leurs actions et y réagir.

Ø Applications environnementales :

Les capteurs peuvent être dispersés à partir d'un avion dans une zone difficile d'accès pour détecter des incendies, surveiller des catastrophes naturelles (inondations, séismes, éruptions volcaniques), surveiller des phénomènes météorologiques, de détecter de la pollution (qualité des eaux, taux d'ensoleillement, taux de radioactivité, fuite du pétrole, taux de CO₂...).

Ø Applications domestiques :

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. En plaçant, sur le plafond ou dans le mur, des capteurs, on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes.

Ø Applications écologiques :

L'intégration de plusieurs micro-capteurs dans le système de climatisation et de chauffage des immeubles. Ainsi, la climatisation ou le chauffage ne sont déclenchés qu'aux endroits où il y a des personnes présentes. Le système distribué peut aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait probablement de réduire la demande mondiale en énergie.

Ø Applications d'agriculture de précision :

Les réseaux de capteurs sont capables d'apporter des bénéfices considérables au domaine d'agriculture, grâce à leur habilité de surveiller les taux de pesticides dans l'eau, le degré d'érosion du sol, détection de parasites, et le niveau de pollution de l'air en temps réel.

Ø Applications industrielles :

L'intégration des micro-capteurs dans un processus de stockage et de livraison de marchandises peut être utilisée pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré.

5.3 Caractéristiques des RCSF :

Parmi les caractéristiques les plus importantes d'un réseau de capteurs, nous citons [3] :

- a) **Topologie dynamique** : la topologie des réseaux de capteurs instable est le résultat des trois facteurs essentiels suivants:
 - Ø **La mobilité des nœuds** : les nœuds capteurs peuvent être attachés à des objets mobiles qui se déplacent librement et arbitrairement, introduisant ainsi une topologie instable du réseau.
 - Ø **La défaillance des nœuds** : du fait de l'autonomie énergétique limitée des nœuds, la topologie du réseau n'est pas fixée (les nœuds « morts » sont, d'un point de vue logique, simplement supprimés).
 - Ø **L'ajout de nouveaux nœuds** : de nouveaux nœuds peuvent facilement être rajoutés. Il suffit de placer un nouveau capteur qui soit dans la portée de communication du réseau déjà existant.
- b) **Auto-organisation** : l'auto organisation s'avère très nécessaire pour ce type de réseau afin de garantir sa maintenance. Vu les différentes raisons résultant une topologie instable du réseau de capteur, ce dernier devra être capable de s'auto organiser pour continuer ses applications.
- c) **La durée de vie limitée** : les nœuds capteurs sont très limités par la contrainte d'énergie, ils fonctionnent habituellement sans surveillance dans des régions géographiques éloignées. Par conséquent recharger ou remplacer leurs batteries devient quasiment impossible.
- d) **Ressources limitées** : Les nœuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans ces nœuds. En conséquence, la capacité de traitement et de mémoire est très limitée.
- e) **Agrégation des données** : Dans les réseaux de capteurs, les données produites par les nœuds capteurs sont très reliées, ce qui implique l'existence de redondances de données. Une approche répandue consiste à agréger les données au niveau des nœuds intermédiaires afin de réduire la consommation d'énergie lors de la transmission de ces données.

- f) **La tolérance de fautes** : le réseau doit être capable de maintenir ses fonctionnalités sans interruptions en cas de défaillance d'un ou plusieurs de ses capteurs. Cette défaillance peut être causée par une perte d'énergie, ou par dommage physique ou interférence de l'environnement. Le degré de tolérance dépend du degré de criticité de l'application et des données échangées.

- g) **La scalabilité** : les réseaux de capteurs engendrent un très grand nombre de capteurs, ils peuvent atteindre des milliers voire des millions de capteurs. Le défi à relever par les RCFSs est d'être capable de maintenir leurs performances avec ce grand nombre de capteurs.

- h) **Bande passante limitée** : En raison de la puissance limitée, les nœuds capteurs ne peuvent pas supporter des débits élevés.

- i) **Sécurité physique limitée**: cela se justifie par les contraintes et limitations physiques qui minimisent le contrôle des données transmises.

I.6 Les réseaux véhiculaires(VANET) :

Les réseaux véhiculaires sont une extension des réseaux ad-hoc souvent équipés de capteurs avec des contraintes très spécifiques. Ils ont une topologie très dynamique impliquant des problèmes de densité et de congestion momentanée, ils n'ont pas de contrainte d'énergie car les équipements sont alimentés sur la grande énergie des véhicules, ils disposent souvent d'informations fiables de localisation voire de cartes routières. Les applications potentielles de ces réseaux sont nombreuses, allant de la communication entre véhicules à des services d'urgence ou de gestion de trafic routier. Il peut s'agir de communications point à point ou de diffusion de données comme dans le cas de la diffusion d'information d'état de trafic.

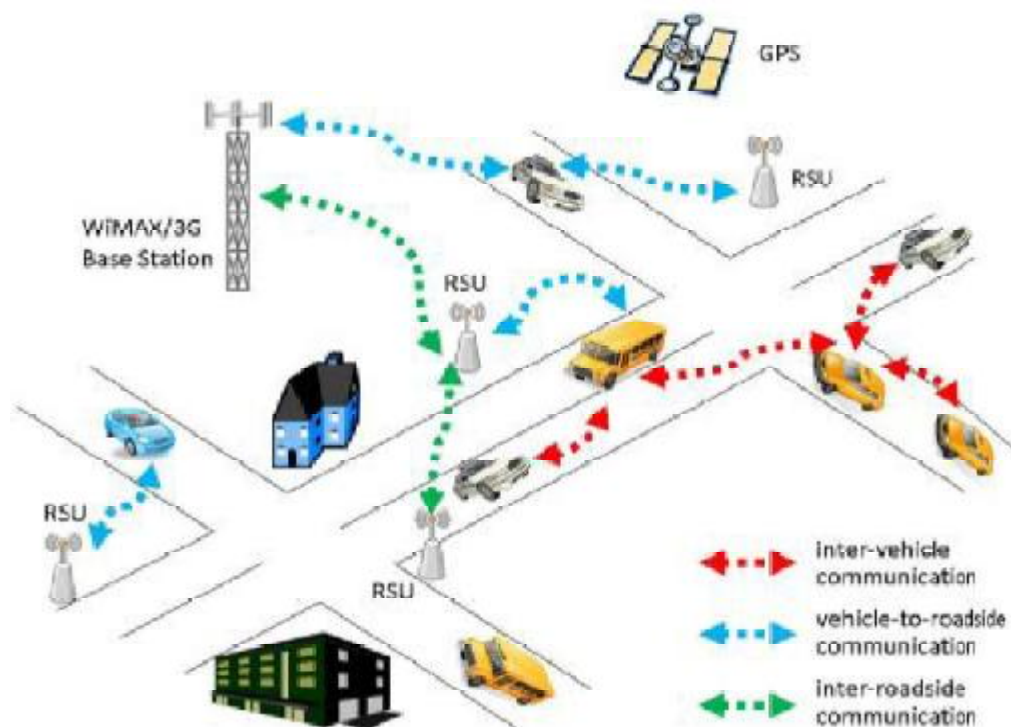


Figure I.8 : Réseau véhiculaire sans fil.

6.1 Les modes de communication dans les réseaux VANET :

Dans les réseaux de véhicules, on peut distinguer deux modes de communication, les communications Véhicule-à-Véhicule (V2V) et les communications Véhicule-à-Infrastructure (V2I). Les véhicules peuvent utiliser un de ces deux modes ou bien les combiner s'ils ne peuvent pas communiquer directement avec les infrastructures. Dans cette section, nous présentons le principe et l'utilité de chaque mode :

6.1.1 Mode de communication Véhicule-à-Véhicule (V2V) :

Ce mode de communication fonctionne suivant une architecture décentralisée, et représente un cas particulier des réseaux ad hoc mobiles. Il est basé sur la simple communication inter-véhicules ne nécessitant pas une infrastructure. En effet, un véhicule peut communiquer directement avec un autre véhicule s'il se situe dans sa zone radio, ou bien par le biais d'un protocole multi-sauts qui se charge de transmettre les messages de bout en bout en utilisant les nœuds voisins qui les séparent comme des relais. Dans ce mode, les supports de communication utilisés sont caractérisés par une petite latence et un grand débit de transmission [8] [9]. Les communications V2V sont très efficaces pour le transfert des informations concernant les services liés à la sécurité routière, mais elles ne garantissent pas une connectivité permanente entre les véhicules.

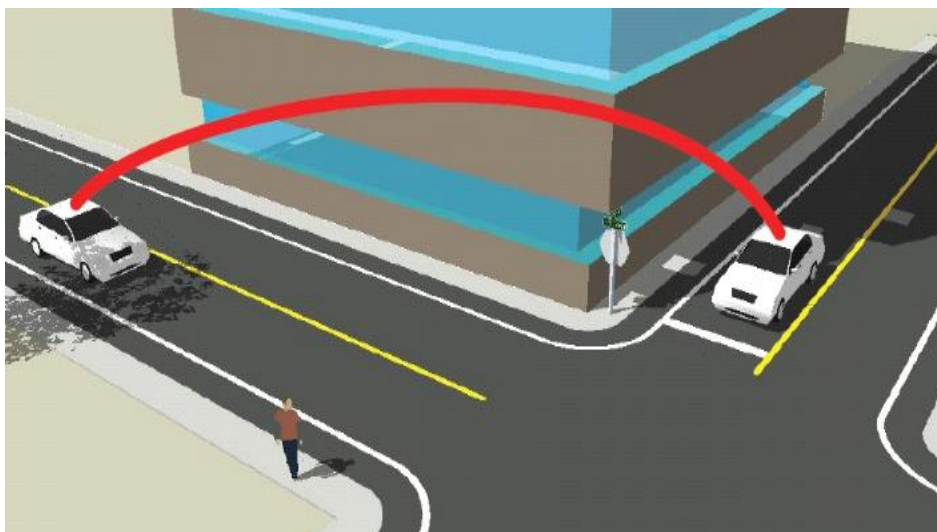


Figure I.9 : Exemple de réseau C

6.1.2 Mode de communication de Véhicule à Infrastructure (V2I) :

Ce mode de communication permet une meilleure utilisation des ressources partagées et démultiplie les services fournis (par exemple : accès à Internet, échange de données de voiture-à domicile, communications de voiture-à-garage de réparation pour le diagnostic distant, ...etc.) grâce à des points d'accès RSU (*Road Side Units*) déployés aux bords des routes; ce mode est inadéquat pour les applications liées à la sécurité routière car les réseaux à infrastructure ne sont pas performants quant aux délais d'acheminement [10] .

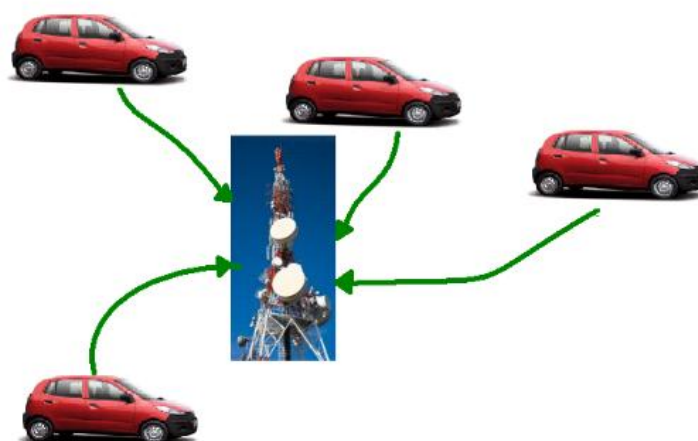


Figure I.10: exemple de réseau V2I

6.2 Les caractéristiques des VANETs :

En plus des caractéristiques des réseaux ad hoc mobiles classiques, les VANETs ont la particularité d'avoir une très grande mobilité (les nœuds mobiles circulent à très grande vitesse), [21] ce qui donne une topologie encore plus dynamique que dans le cas des MANETs, et par conséquent, moins de performances. Les nœuds mobiles sont également restreints à des déplacements suivant une carte routière, dans le sens où, en général, un nœud ne peut quitter la route lors de ces déplacements.

6.3 Applications des VANET :

Une des applications de ce concept consiste à munir les voitures et les routes de capacités de communication permettant de rendre la route plus sûre et de rendre le temps passé sur les routes plus convivial. Cette application est appelée le système de transport intelligents (ITS, Intelligent transportation System) [12].

On peut donc distinguer deux types d'applications avec les réseaux de véhicules, les applications de confort et les applications de sécurité routière. Les contraintes de ces applications sont différentes comme par exemple la vitesse de propagation de l'information. Dans le cas d'un accident, il faut prévenir les usagers dans un temps borné alors que la diffusion de publicités n'a pas cette contrainte de temps mais elle sera par contre plus consommatrice de bande passante.

Nous allons donc décrire dans les paragraphes suivants quelques applications.

6. 3.1 Applications de sécurité routière :

a) Alerte :

Ce service permet, dans le cas d'un véhicule en panne, d'avertir les véhicules se dirigeant vers le lieu de la panne que les conditions de circulation se trouvent modifiées et qu'il est nécessaire de redoubler de vigilance (figure I.11). Il est nécessaire, également, en cas de densité réduite de véhicule de pouvoir conserver l'information pour pouvoir la retransmettre si un véhicule entre dans la zone de retransmission. Les messages de sécurité devront être émis à des périodes régulières. Ainsi le ou les nœuds désignés pour la retransmission des messages émettront des alertes à des instants réguliers. Les messages devront être de taille réduite pour

être transmis le plus rapidement possible. Les messages devront comporter les coordonnées du lieu de la panne et les paramètres de la zone de retransmission.

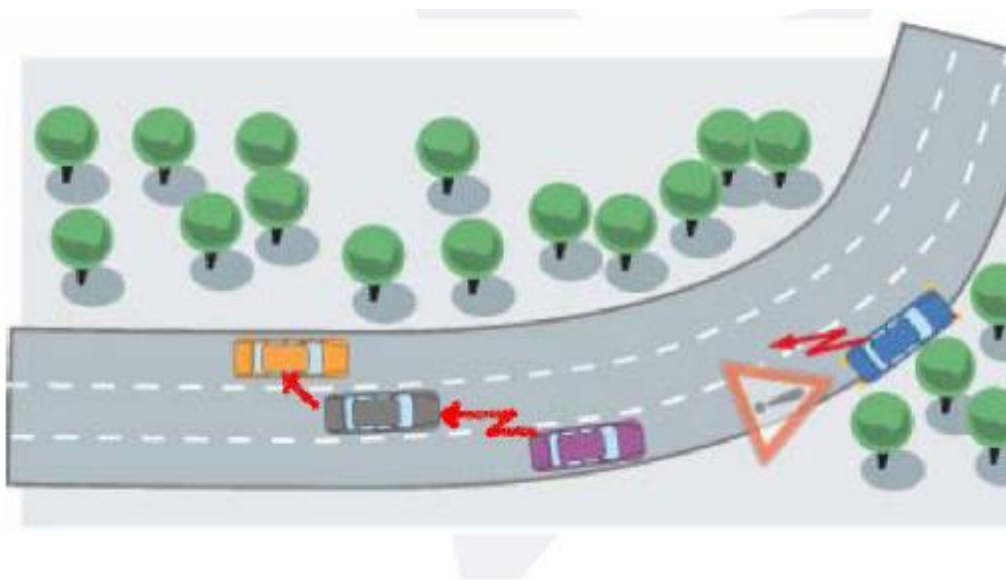


Figure I.11 : véhicule en panne

b) alerte en cas de ralentissement anormal (bouchon, travaux, intempéries) :

Ce service permet d'avertir les automobilistes de situations de circulation particulières (Figure I.12). Quelque soit la nature des difficultés de circulation l'information renseigne l'automobiliste qu'il est nécessaire de ralentir. Le message d'alerte est émis par un véhicule détectant les difficultés de circulation (freinage important par exemple, déclenchement des feux de détresse, pluie). Un véhicule banalisé effectuant des travaux peut également être l'origine du message d'alerte. Comme pour le message d'alerte informant d'un accident ou d'une panne, le message d'alerte informant d'un ralentissement doit être transmis aux autres véhicules de façon efficace et rapide.

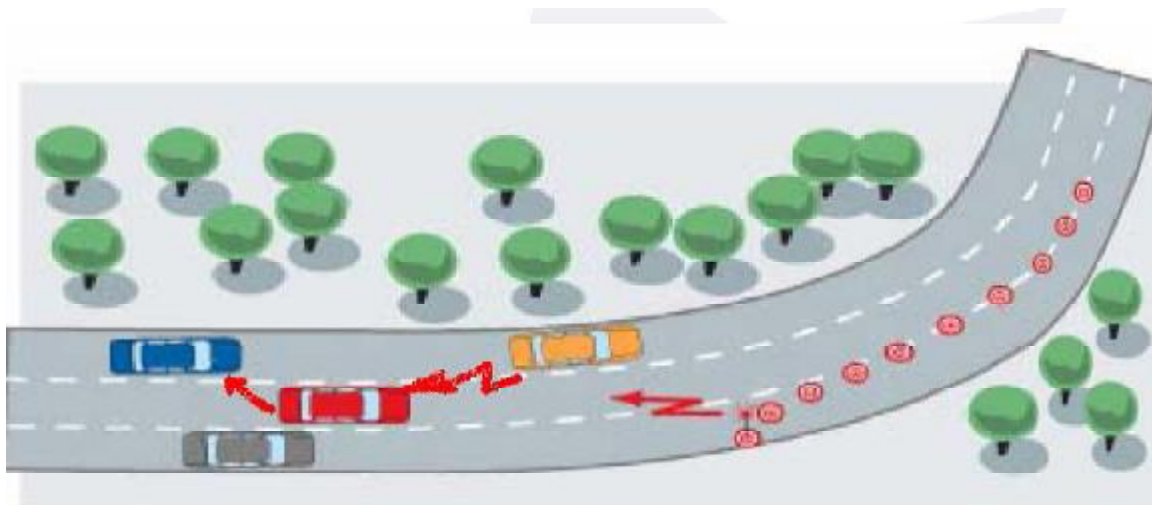


Figure I.12 : Travaux sur la route

c) la conduite collaborative

La conduite collaborative est un concept qui améliore considérablement la sécurité du transport routier (figure I.13). Cette innovation est basée sur un échange de renseignements entre des véhicules munis de dispositifs (ex : capteurs) leur permettant de percevoir ce qui les entoure et de collaborer en groupes. Ces groupes de véhicules ou réseaux ponctuels, peuvent élaborer une stratégie de conduite collective qui exigerait peu ou pas d'interventions de la part des conducteurs.

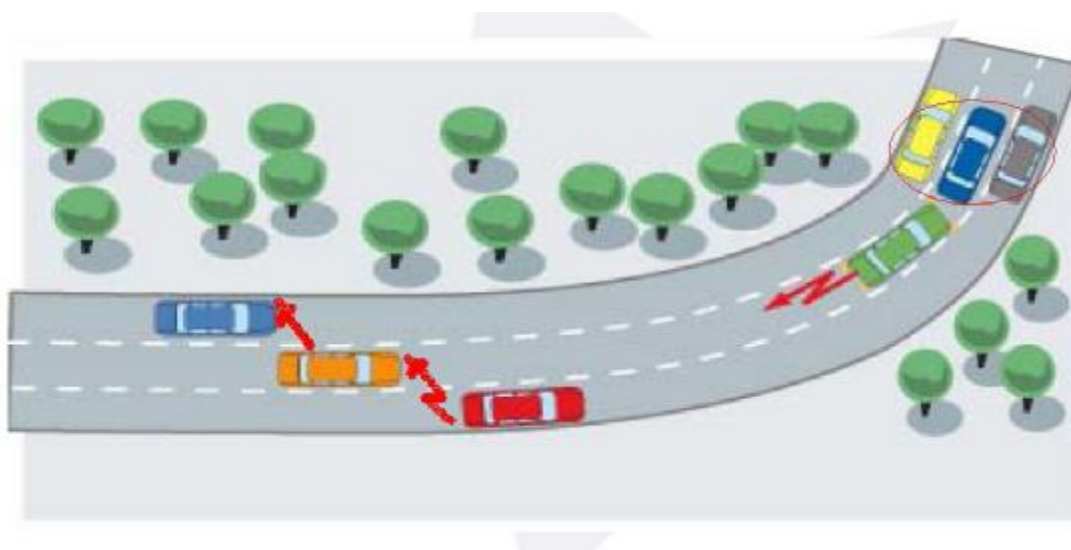


Figure I.13 : Risque de collision

6.3.2 Applications de confort :

a) réseaux collaboratifs :

Les réseaux collaboratifs sont en train de se développer en particulier avec les réseaux pairs à pairs. On peut imaginer une chaîne de radio ou de « télévision distribuée » où chaque véhicule va partager les musiques et vidéos qu'il a en sa possession pour construire un programme de diffusion continu.

b) internet dans les transports :

Aujourd'hui, les hot spots (zone wifi à accès Internet) sont de plus en plus développés dans les villes, en particulier avec les initiatives des communautés et des opérateurs de télécommunication. En voiture, on peut imaginer acheter de la musique et de la vidéo, au niveau d'une station essence, d'une gare ou même en pleine autoroute (en passant d'une voiture à une autre jusqu'au point d'accès le plus proche). Les passagers dans la voiture pourront ainsi jouer en réseaux, ou encore même naviguer sur Internet (figure I.14).

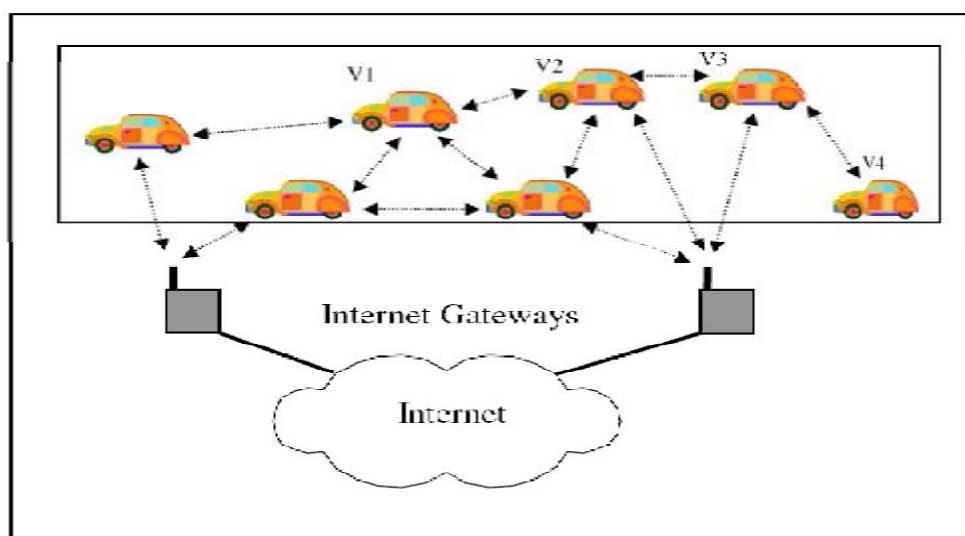


Figure I.14 : Accès à internet

c) gestion des espaces libres dans les parkings

Ce service permet de rassembler des informations sur la disponibilité de l'espace de stationnement dans les parkings et de coordonner entre automobilistes afin de les guider aux espaces libres (figure I.15).

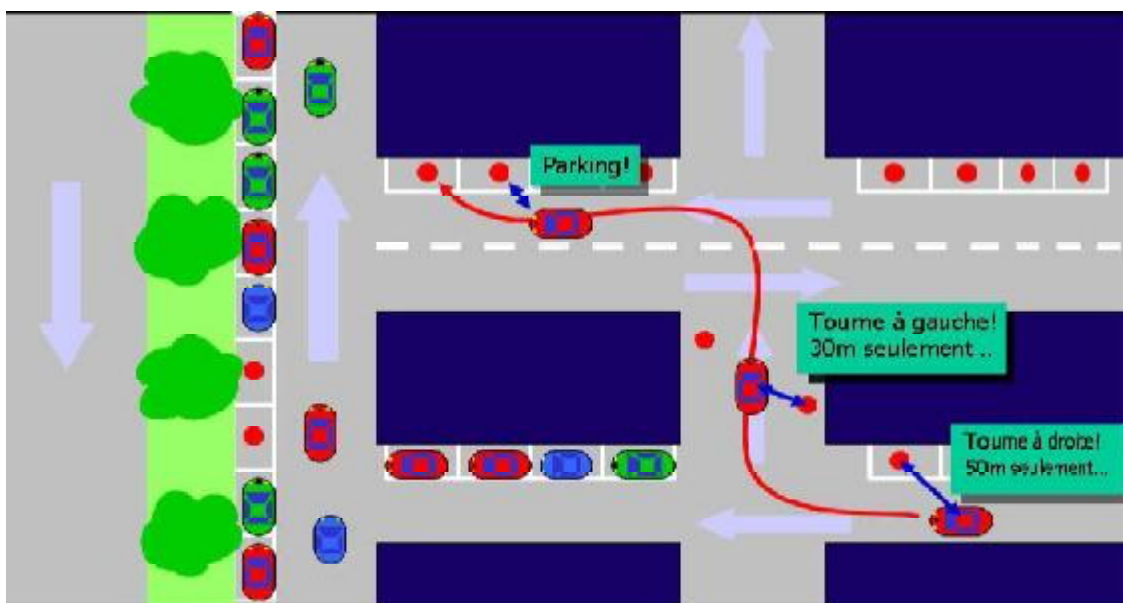


Figure I.15 : Parking intelligent

6.4 Travaux dans le domaine des VANET :

Les VANETs s'ouvrent à plusieurs domaines de recherche dont nous citons les plus importants [21]:

- Ø **Sécurité** : vu la sensibilité des domaines d'utilisation des VANETs, une intrusion d'un véhicule malicieux aurait des conséquences graves sur l'ensemble des véhicules interconnectés, c'est justement pour cette raison que la majorité des recherches lui sont consacrés.
- Ø **Localisation des véhicules** : si l'un des véhicules du réseau doit être localisé (dans le cas d'un accident par exemple), les autres doivent être informés de sa position. Le problème est que tous les véhicules ne sont pas équipés d'un système de repérage par satellite (GPS), c'est pour cette raison qu'un mécanisme de localisation sans utiliser le GPS est nécessaire.

- Ø **Problèmes de congestion** : l'un des problèmes des VANETs est que chaque véhicule communique avec tous ceux qui sont dans sa zone de couverture, ce qui entraîne une dégradation de la qualité de service (QoS) avec l'augmentation du nombre de véhicules. Ce problème a été l'objet de plusieurs études.
- Ø **Routage** : la principale particularité des VANETs est qu'ils changent rapidement de topologie à cause de leur mobilité élevée, ce qui fait du routage un problème très difficile à gérer et un axe de recherche pour beaucoup de chercheurs.

6.5 Mobilité des nœuds dans un VANET :

Pour analyser les performances des solutions proposées dans les VANETs, les chercheurs recourent généralement à des outils de simulation. Dans ces derniers le facteur de mobilité a longtemps été négligé et réduit à des modèles de mobilité aléatoires ou déterministes, or le déplacement des usagers n'est ni aléatoire ni complètement déterministe. Il obéit toutefois à des règles qui peuvent être reproduites par des modèles appropriés.

Le rendement et la performance des solutions peuvent dépendre fortement du déplacement des usagers. C'est pour cela que le choix d'un modèle réaliste est important pour la simulation de ces solutions.

Notre travail se situe justement dans le cadre d'intégration d'un modèle de mobilité réaliste à un simulateur de réseaux de capteur sans fil qui est J-SIM.

I.7 Conclusion :

Dans ce chapitre, nous avons présenté les réseaux sans fil et le rôle primordial qu'ils jouent dans de nombreux domaines. Nous les avons classifié suivant la zone de couverture, ou encore le mode opératoire (avec ou sans infrastructure).

L'objectif principal de ce chapitre a été de présenter les réseaux ad hoc mobiles (MANET) en général, et les réseaux ad hoc véhiculaire (VANET) en particulier.

Les VANETs représentent un axe de recherche récent comportant plusieurs domaines. Vu l'importance de la simulation dans le test de ces réseaux, de plus en plus d'équipes de recherche s'intéressent à un aspect de la simulation qui est la modélisation de la mobilité. Dans le chapitre suivant, nous parlerons des différents modèles de mobilité les plus importants pour les MANETs et les VANETs.

II.1 Introduction :

Afin de simuler un nouveau protocole pour un réseau ad-hoc, il est impératif d'employer un modèle de mobilité qui peut être déterminant pour la performance du protocole.

Avant qu'un protocole soit exploitable, il faut qu'il soit testé en utilisant un modèle de mobilité pour évaluer ses performances. Il y a deux types de modèles de mobilité utilisés dans la simulation des réseaux: traces et modèles synthétiques. Les traces sont des modèles de mobilité qui sont observés dans les systèmes réels. Les traces fournissent des informations précises, particulièrement quand elles impliquent un grand nombre de participants et une période d'observation convenablement longue.

Cependant, de nouveaux environnements de réseau (par exemple les réseaux ad hoc) ne sont pas facilement modélisés si des traces n'ont pas été encore créées. Dans ce type de situation, il est nécessaire d'employer les modèles synthétiques. Les modèles synthétiques essaient de représenter normalement les comportements de nœuds mobiles sans utilisation des traces.

Dans ce chapitre, nous présentons premièrement plusieurs modèles de mobilité synthétiques qui ont été proposés pour l'évaluation des performances des protocoles des réseaux ad hoc mobiles (MANET) en les classifiant selon le mode de déplacement : individuel et en groupe, puis les modèles de mobilité dans les VANET Vehicular Ad hoc NETWORK en deuxième lieu, en dernier lieu les modèles basés sur l'activité des nœuds.

Un modèle de mobilité définit les mouvements des nœuds mobiles lors d'une simulation d'un réseau mobile sans fil dans une surface donnée (carte réelle, salle de conférence,...). Ces modèles peuvent définir des mouvements libres ou restreints par des contraintes de mobilité (vitesse limitée, présence d'obstacles, contraintes de réalisme, ...).

II.3 Modèles de mobilité des MANETs:

Les modèles qui vont suivre sont des modèles qui décrivent des déplacements libres et sans obstacles, avec plus ou moins de réalisme. Ils sont adaptés aux mouvements d'unités mobiles sur des espaces ouverts tels qu'une salle de conférence, un campus ou encore un jardin public. Ces modèles ont été utilisés pour simuler les MANETs.

II.3.1 Modèle de mobilité basé sur les déplacements individuels :

Les modèles de ce type définissent des déplacements dits individuels, c'est-à-dire, chaque nœud mobile se déplace indépendamment des autres nœuds. Autrement dit, quand un nœud calcule sa prochaine destination, il utilise des paramètres qui lui sont propres (vitesse, direction, ...). Le nombre et la nature de ces paramètres dépendent du modèle de mobilité.

3.1.1 Random Mobility Model :

RMM est un modèle simple, sa caractéristique est l'utilisation des valeurs aléatoires pour les paramètres vitesse et direction.

Ces deux valeurs restent constantes pendant une période de temps ou une distance parcourue, à chaque fois qu'une période ou une distance est achevée, le nœud choisit des nouvelles valeurs aléatoirement, et recommence le processus.

Si la période de temps et la distance sont petites, la simulation de mouvement du nœud sera restreinte. Ce modèle est simple à utiliser ne demande pas beaucoup de paramètres mais ne représente pas la réalité ; comme ce modèle ne dispose pas d'un historique, les valeurs courantes choisies ne dépendent pas des anciennes valeurs.

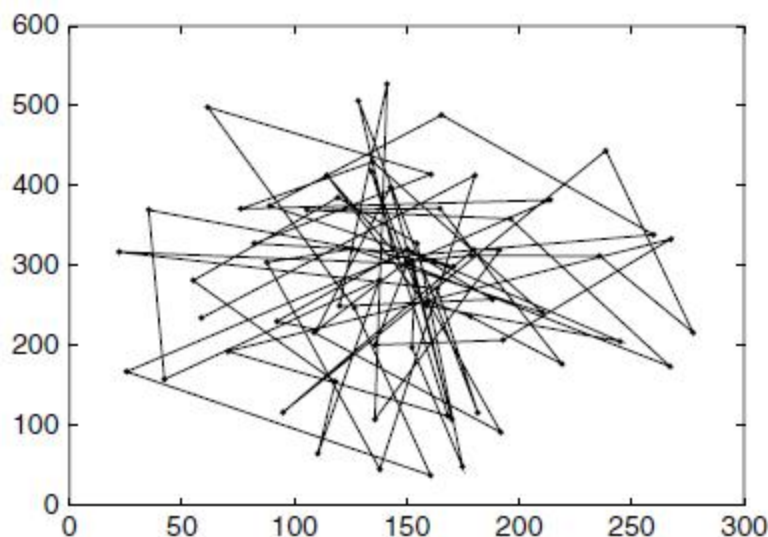


Figure II.1: le modèle Random Mobility Model.

3.1.2 Random Waypoint :

Le modèle Random Waypoint Mobility Model proposé par [11], il est fréquemment utilisé dans la simulation des MANET car il est présent dans les simulateurs les plus connus. Sa simplicité est peut-être la raison pour laquelle l'utilisation de ce modèle est très répandue.

Ce modèle inclut des temps de pause entre les changements de direction et de vitesse. Au début de la simulation chaque nœud sélectionne aléatoirement une destination dans la zone de simulation, il se déplace vers cette destination avec une vitesse choisie aléatoirement dans l'intervalle $[0, V_{max}]$. Le paramètre V_{max} est la vitesse maximale permise pour chaque nœud, la vitesse et la direction sont choisies indépendamment des autres nœuds. En arrivant à la destination le nœud marque une pause pour une période de temps spécifiée T_{pause} . V_{max} et T_{pause} sont les deux paramètres qui déterminent le comportement de la mobilité des nœuds. À noter que le mouvement de nœud qui utilise ce modèle est similaire au modèle Random MobilityModel si le temps de pause $T_{pause}=0$ et l'intervalle de vitesse est le même.

Les nœuds sont initialement distribués dans la zone de simulation. Cette distribution initiale des nœuds ne représente pas la manière dont les nœuds se déplacent. Selon des études récentes la densité de la distribution des nœuds dans le centre de la zone de simulation est forte. La variation dans la densité des nœuds dépend de la vitesse et du temps de pause.

La figure II.2 illustre un exemple de mouvement du nœud dans le Random Waypoint Model qui commence dans un point choisi aléatoirement (130,180). La vitesse de nœud est choisie entre 0 et 10 m s^{-1} .

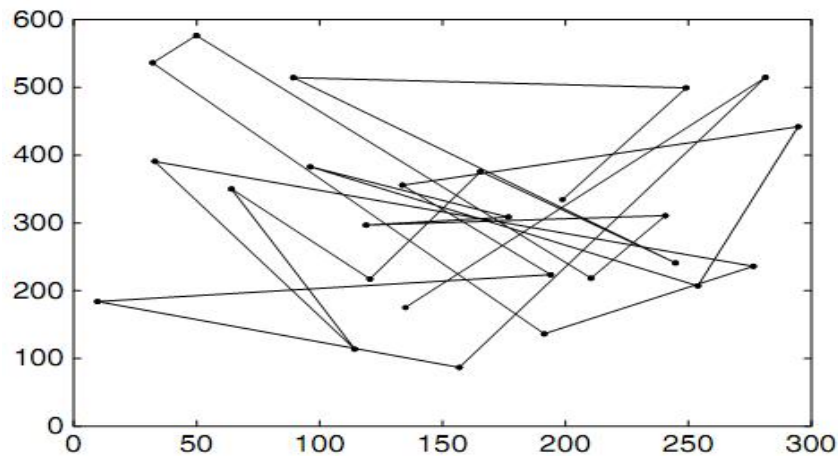


Figure II.2 : Modèle Random WayPoint.

Il existe également une relation complexe entre la vitesse des nœuds et le temps de pause dans le modèle de mobilité Random Waypoint. Par exemple, un scénario avec des mouvements rapides et de longues pauses produit au final un réseau plus stable que le scénario où les nœuds sont lents avec de courtes pauses [15].

Ce modèle présente quelques inconvénients qui sont les suivants :

- Des mouvements peu réalistes et imprévisibles, un nœud mobile change brusquement de vitesse et de direction.
- L'absence totale de mémoire, la vitesse et la direction courantes ne dépendent pas de leurs valeurs précédentes.
- L'absence de dépendance entre les mouvements des nœuds, dans le monde réel, il y a souvent une dépendance entre les mouvements des nœuds.

3.1.3 Random Direction :

Ce modèle présenté dans [13] est conçu pour la gestion de mouvement au centre de la zone de la simulation qui est produit par le modèle RWP. La congestion de mouvement est présentée sur une partie de la zone de la simulation. La probabilité qu'un nœud choisisse une destination qui se situe au centre de la zone est très élevée, pour alléger cette situation le modèle Random Direction est développé.

Ce modèle choisit une direction aléatoire, la caractéristique de ce modèle est qu'au lieu de se diriger vers le centre les nœuds se dirigent vers les bords de la zone de la simulation pour éviter la congestion au centre, une fois que le nœud est arrivé au bord il marque une pause puis choisit un autre angle de direction et recommence le processus.

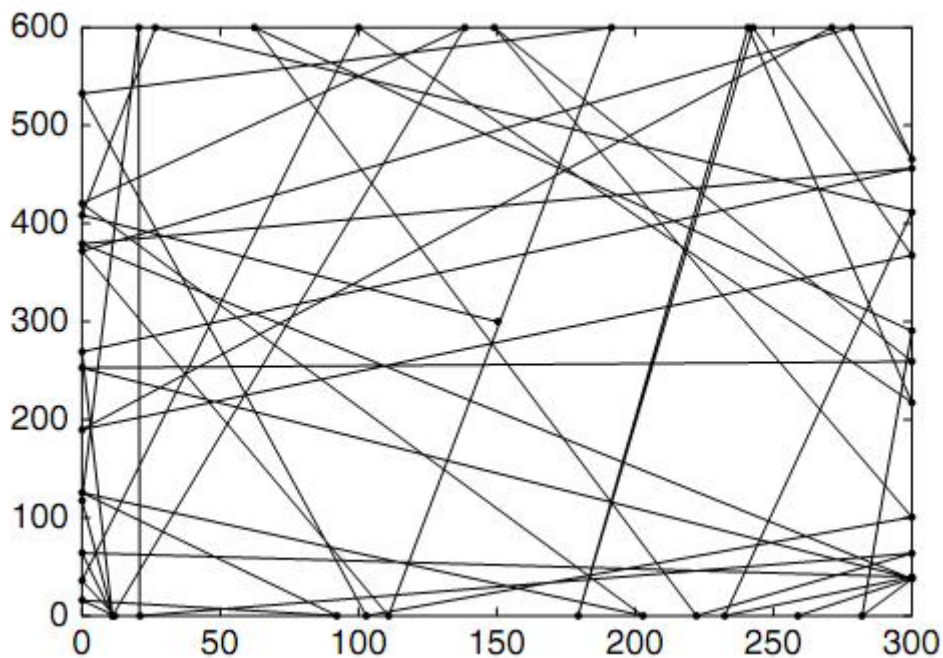


Figure II.3: Le modèle Random Direction

3.1.4 Boundless Simulation Area :

Dans ce modèle une relation entre la direction précédente et la vitesse d'un nœud avec sa direction courante est établie

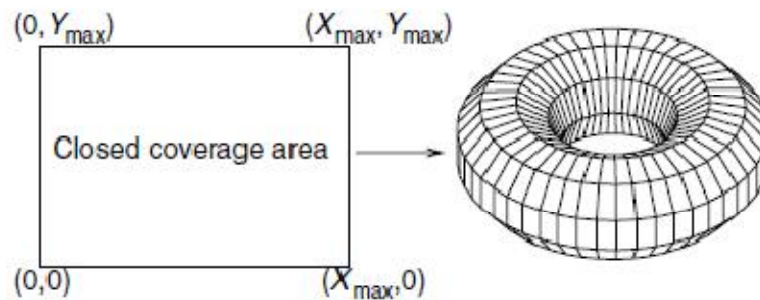


Figure II.4: Boundless Simulation Area Model.

Ce modèle est différent dans la manière dont les bords de la zone de simulation sont manipulés. Dans les modèles précédents quand un nœud atteint la frontière il marque une pause mais dans ce modèle il continue son déplacement et réapparaît dans le côté opposé et continue son déplacement avec la même vitesse et la même direction. Quand les étapes de déplacement se terminent le nœud choisit une nouvelle direction et une nouvelle vitesse. Cette technique crée une zone de simulation sous forme d'un tore qui permet à un nœud de se déplacer sans obstacles.

3.1.5 Smooth Random Mobility Model :

Ce modèle est une amélioration de modèle RMM, il est caractérisé par les changements progressifs de vitesse et de direction, en donnant des courbes douces ce qui permet d'avoir des mouvements réalistes. Les changements sont effectués grâce à deux processus aléatoires, le premier contrôle la vitesse, le deuxième contrôle la direction.

La modélisation de la vitesse se fait grâce à deux notions : la vitesse cible (la vitesse que le nœud veut atteindre) et l'accélération. Au début de la simulation, tous les nœuds ont une

vitesse nulle ($v(0)=0$). Un nœud garde sa vitesse courante jusqu'à ce qu'une nouvelle vitesse lui soit choisie par le processus aléatoire de vitesse. Il accélère ou ralentit (selon sa vitesse actuelle) jusqu'à atteindre la vitesse cible.

Le contrôle de la direction ne diffère pas de celui de la vitesse. Au début de la simulation, chaque nœud a une direction initiale $\theta(0)$ choisie uniformément dans $[0, 2\pi]$. Un nœud se déplace en ligne droite jusqu'à ce qu'une nouvelle direction soit choisie par le processus qui contrôle la vitesse. Le nœud change de direction progressivement jusqu'à atteindre la nouvelle. A ce moment, il continue son mouvement en ligne droite jusqu'à ce qu'un nouveau changement de direction ait lieu.

Ce modèle peut simuler un comportement de pause/marche en choisissant la vitesse 0 comme une des vitesses préférées et en lui affectant une forte probabilité. Il prend également en considération le fait qu'un nœud doit ralentir lorsqu'il change de direction, et cela pour donner une simulation plus réaliste.

II.3.2. Modèles basés sur un déplacement en groupe :

Contrairement aux modèles vus précédemment, les modèles de ce type sont utilisés pour simuler une mobilité où les déplacements des nœuds sont dépendants et tendent à se ressembler, c'est-à-dire que les nœuds se déplacent en groupe et non individuellement. Les modèles qui vont être présentés illustrent ce type de mobilité.

3.2.1 RPGM (Reference Point Group Model):

Dans ce modèle, les nœuds sont arrangés en groupes autour d'un nœud appelé centre logique ou point de référence [14]. Le centre définit le déplacement du groupe suivant un vecteur de mouvements \vec{G} . Ce vecteur contient la vitesse, la direction et la position actuelle du point de référence. Les autres membres se déplacent à l'intérieur du groupe suivant un vecteur \vec{M} obtenu en combinant \vec{G} avec un vecteur aléatoire \vec{R} . Le vecteur résultant permet d'avoir des nœuds qui suivent les mouvements du centre et qui se déplacent autour de lui d'une certaine manière.

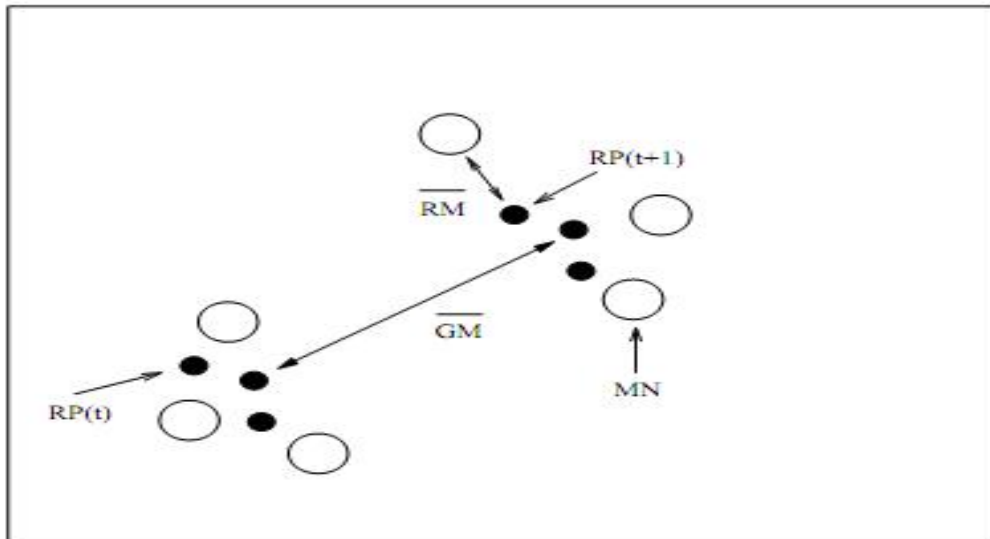


Figure II.5 : Mouvements de trois MNs utilisant le modèle RPGM

3.2.2 Nomadic Mobility Model :

Ce modèle, défini dans [15], simule le comportement d'une communauté nomade, qui se déplace de lieu en lieu. Les nœuds mobiles d'une même communauté se déplacent collectivement d'un point à un autre de la zone de simulation. Chaque nœud de la communauté se déplace à l'intérieur de celle-ci en utilisant un modèle de mobilité à déplacement individuel (par exemple, Random Mobility Model), mais il ne doit pas trop s'éloigner de son point de référence. Quand le point de référence change de position, les nœuds de sa communauté le suivent jusqu'à sa nouvelle destination. Ensuite, ils commencent à se déplacer autour de lui. Les paramètres du modèle de mobilité à déplacement individuel utilisé définissent la dispersion des nœuds autour de leur point de référence.

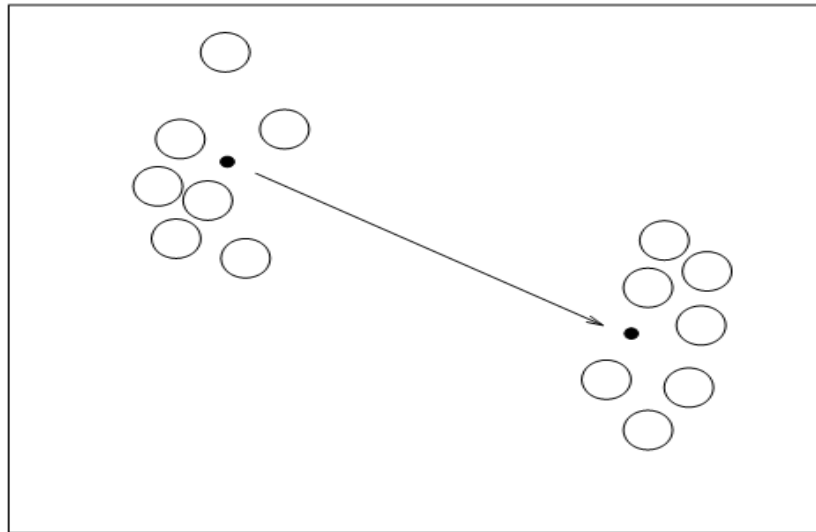


Figure II.6 : mouvements des nœuds utilisant le modèle Nomadic Mobility Model.

3.2.3 Pursue Mobility Model :

Le modèle décrit dans [15] permet de représenter la poursuite d'un nœud cible par un groupe de nœuds et cela est défini par l'équation suivante :

Nouvelle-position = ancienne_position* accélération (cible - ancienne_position) + vecteur aléatoire.

Où l'accélération (cible - ancienne_position) définit l'information de nœud poursuivi et le vecteur aléatoire représente le déplacement aléatoire de chaque nœud, ce déplacement aléatoire est obtenu par un modèle de mobilité tel que RWP. Les mouvements aléatoires sont limités afin de maintenir l'acheminement efficace vers la cible.

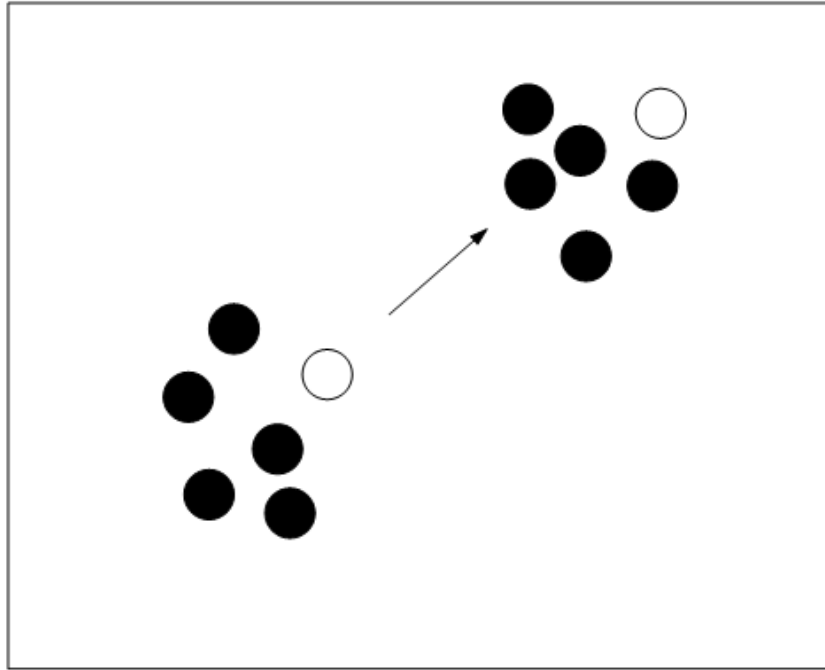


Figure II.7 : mouvements des nœuds utilisant le modèle Pursue Mobility Model.

3.2.4 Column Mobility Model:

Ce modèle crée des scénarios selon [37]. Dans ce modèle, les nœuds mobiles forment une ligne et se déplacent uniformément vers une direction particulière. L'utilisateur indiquera le nombre de nœuds et le nombre de groupes. Le nombre de nœuds doit être également divisible par le nombre de groupes, c.-à-d., tout le groupe de nœuds doit être complètement rempli. La colonne des points de référence sélectionne un angle aléatoire d'orientation et un vecteur aléatoire de mouvement. Les nœuds suivent leur point de référence individuel à travers la carte. Ils ont un paramètre qui détermine la distance qui les éloigne de leur point de référence.

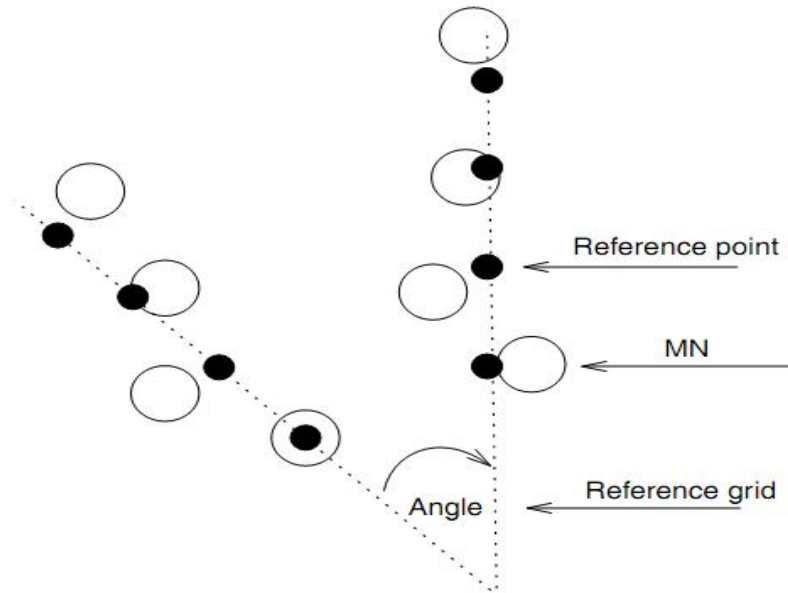


Figure II.8 : mouvements des nœuds utilisant le modèle Column Mobility Model.

II.4. Les modèles de mobilité de VANETS :

Les modèles de mobilité vus précédemment sont des modèles généraux pour les MANETs. Cependant, la mobilité véhiculaire est restreinte à des routes et obéit à des mécanismes de contrôle de trafic. Il est donc nécessaire de mettre en œuvre des modèles adéquats suivant certains scénarios. Ces modèles spécifiques aux VANET sont plus gourmands en temps et en mémoire que les modèles généraux mais sont plus adaptés aux déplacements véhiculaires.

4.1 Freeway :

Ce modèle est représenté par une carte contenant plusieurs autoroutes dont chaque route est composée de plusieurs voies [16]. Ce modèle ne prend pas en considération les intersections car il n'inclut pas les routes urbaines. Au début de la simulation les nœuds sont placés aléatoirement sur les voies, et ils se déplacent avec des vitesses qui dépendent d'un historique, où la vitesse de chaque véhicule change en suivant une accélération aléatoire. En plus de ces deux paramètres (la vitesse et l'accélération), ce modèle définit la distance de sécurité entre deux véhicules qui se suivent sur la même voie. Si la distance entre deux véhicules est inférieure à la distance exigée, le deuxième véhicule doit ralentir pour s'éloigner de premier nœud. Ce modèle ne permet pas le changement de voies. Le véhicule se déplace sur la voie jusqu'à atteindre la limite de la zone de la simulation.

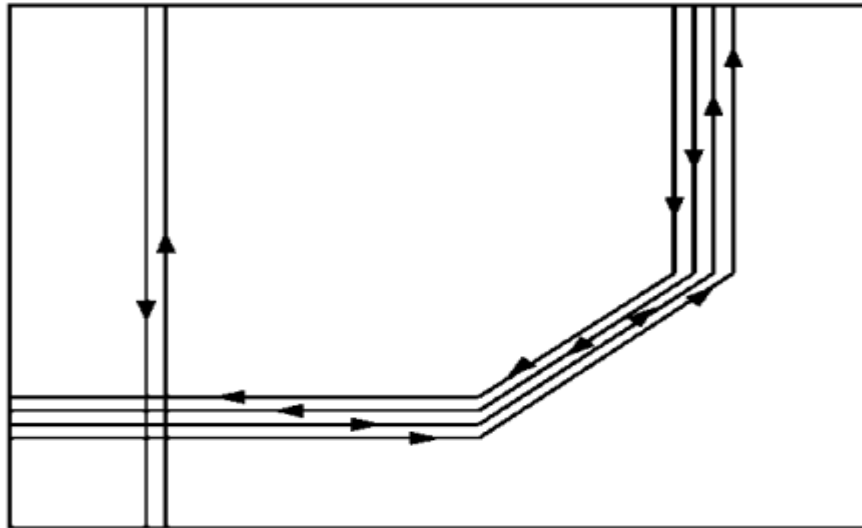


Figure II.9 : le modèle Freeway.

4.2 Manhattan :

Ce modèle [16] se base aussi sur une carte, il est introduit pour simuler l'environnement urbain.

Ce modèle utilise une carte composée de routes horizontales et verticales dont chaque route inclut deux voies, permettant le mouvement dans les deux directions (nord/sud pour les routes verticales et est/ouest pour les routes horizontales). En atteignant un carrefour, le véhicule choisit aléatoirement une direction à suivre : continuer tout droit, tourner à droite ou à gauche avec des probabilités de 0.5, 0.25 et 0.25 respectivement.

La distance de sécurité est également utilisée dans ce modèle et les nœuds suivent la même stratégie que dans le modèle Freeway pour maintenir cette distance. Mais contrairement au modèle précédent, un véhicule peut changer une voie dans un carrefour. Néanmoins, il n'y a aucun mécanisme de contrôle dans ces points (carrefour), où les nœuds continuent leurs mouvements sans s'arrêter.

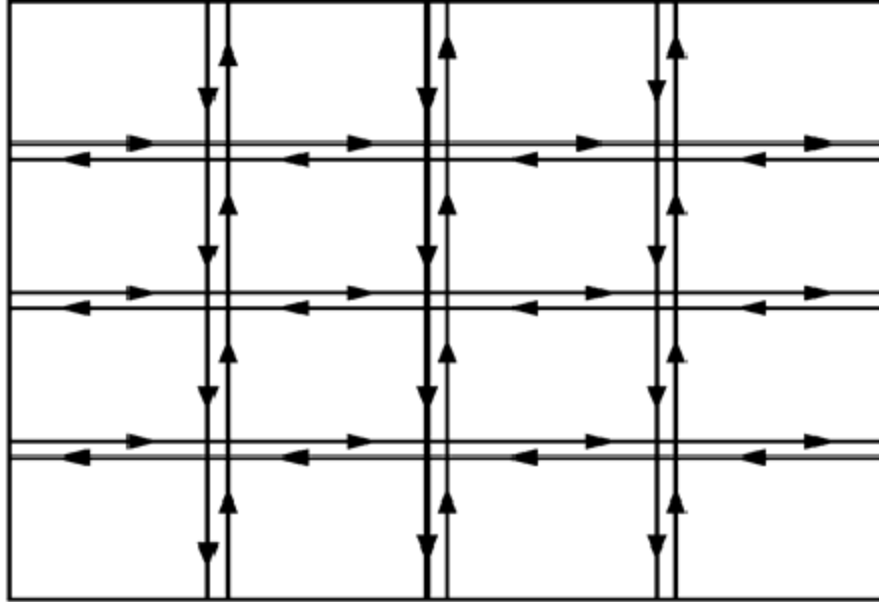


Figure II.10 : Déplacements dans le modèle Manhattan.

4.3 City Section Mobility Model :

CSM [17] peut être considéré comme un modèle hybride entre le modèle (RWP), dans lequel les nœuds mobiles se déplacent aléatoirement et librement sans restrictions, et le modèle Manhattan qui introduit le choix aléatoire de la destination, dans une zone urbaine. À chaque étape du mouvement du véhicule un point aléatoire est choisi de la carte utilisée, vers lequel il se déplace suivant le plus court chemin.

Après avoir atteint cette destination, il s'arrête pour faire une pause, et ensuite répète le processus. La distance de sécurité contraint la vitesse des nœuds.

Sur le schéma suivant, le mouvement de deux nœuds A et B selon le modèle de mobilité city section a été illustré.

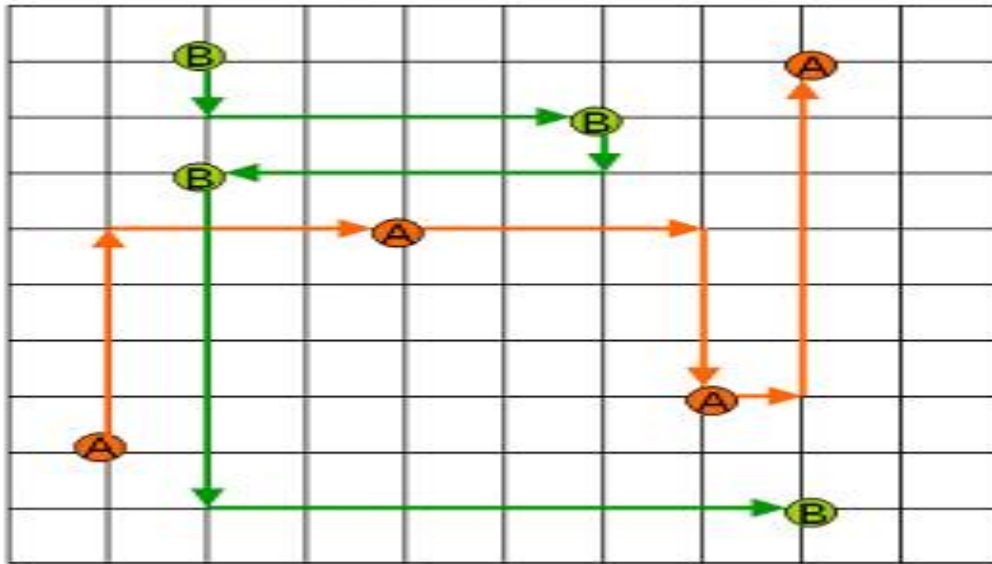


Figure II.11: City Section Mobility Model

4.4 STRAW:

Le modèle STRAW a été conçu par [18]. Comme les modèles précédents, excepté le modèle Freeway. Il impose aux nœuds des déplacements sur une carte routière et définit certaines règles de conduite.

STRAW utilise des informations sur des régions réelles afin de créer la zone de simulation. Ces informations sont tirées de la base de données TIGER/lines. Ainsi, la zone de simulation est une zone comportant un réseau de rues sur lesquelles se déplacent les nœuds mobiles.

La mise en œuvre de STRAW comporte trois modules qui sont : la mobilité intra segments, la mobilité inter segments et, la gestion de la route et l'exécution.

Le premier module est responsable de la mobilité à l'intérieur d'un segment. Lorsque la simulation commence, une position initiale est choisie pour chaque nœud : une rue est choisie aléatoirement et ce nœud y est placé juste avant l'une de ses deux intersections. Si un autre véhicule est déjà sur cette rue, le nouveau est placé derrière lui. Les nœuds se déplacent ensuite suivant le modèle car-following, de façon à ce qu'ils essaient d'accélérer à un rythme constant allant jusqu'à 5 m/s^2 pour atteindre la vitesse maximale.

Le deuxième module de STRAW comporte la mobilité au niveau des intersections. Le modèle intègre deux niveaux de contrôle d'admissions (dans un segment). La première forme de contrôle d'admission simule les mécanismes de contrôle du trafic usuels. STRAW utilise

les panneaux de stop et les feux de circulation. La seconde forme de contrôle d'admission assure que le prochain segment de route qu'une voiture veut emprunter aura assez d'espace pour l'accueillir, et ce, avant qu'elle n'enclenche sa traversée.

STRAW affecte les panneaux selon la classe des segments de routes des intersections. Par exemple, lorsque deux routes voisines se croisent, un feu de circulation est placé, par contre, lorsqu'une route secondaire rencontre une autoroute, un panneau de stop est installé.

Le troisième et dernier module de STRAW est la gestion de la route et l'exécution. Ce module détermine les chemins empruntés par chaque véhicule durant la simulation. Deux approches ont été utilisées : la mobilité inter segments simple (Simple STRAW) et la mobilité avec la paire origine-destination (STRAW OD). Dans la première approche, le prochain segment que prendra un véhicule est déterminé aléatoirement à chaque intersection. Dans ce cas une probabilité p est utilisée à chaque fois pour sélectionner le prochain segment que le véhicule devra emprunter : il continue tout droit avec la probabilité p et tourne avec la probabilité $1-p$ (dans ce cas, il choisit une direction parmi les directions disponibles de façon aléatoire). Dans la deuxième approche, la décision est basée sur le plus court chemin entre la position courante du véhicule et sa destination. Cette approche décrit avec plus d'exactitude les mouvements véhiculaires, cependant elle induit un temps de calcul et d'exécution plus important.

Le modèle STRAW est l'un des modèles les plus réalistes existants. En effet, il repose sur l'utilisation de cartes réelles, il intègre des mécanismes de contrôle du trafic au niveau des intersections et il définit des comportements réalistes (distance de sécurité entre deux véhicules qui se suivent, et calcul du chemin optimal pour STRAW OD).

4.5 Stop Sign Model:

Ce modèle [35] intègre un mécanisme de contrôle de trafic. Dans chaque carrefour, un panneau de stop est placé, qui oblige les véhicules à ralentir et faire une pause.

Ce modèle est basé sur des cartes réelles de la base de données de TIGER/Lines [36], SSM assigne une seule voie par route dans chaque direction. Un véhicule ne peut pas dépasser son successeur et doit contrôler sa vitesse pour garder la distance de sécurité. Si plusieurs véhicules arrivent à une intersection en même temps, ils forment une file d'attente, et

chacun attend son tour pour traverser le carrefour. Ceci a comme conséquence le rassemblement des nœuds, et affecte énormément la connectivité de réseau aussi bien que la mobilité de véhicule (vitesses moyennes).

La disposition des panneaux de stop sur les cartes n'est pas réaliste car ce n'est pas évident de trouver un panneau de stop de chaque route d'une intersection. Pour cela ce modèle est amélioré en supposant le modèle TSM (Traffic Sign Model).

4.6 Traffic Sign Model :

Ce modèle est une amélioration de SSM en remplaçant les panneaux de stop par des feux de signalisation aux intersections [19]. En général, les véhicules doivent s'arrêter seulement aux signaux qui sont rouges et traversent lorsque les signaux sont verts. Le comportement des véhicules avec ces feux est le suivant :

Quand un nœud s'approche d'une intersection et ce nœud lui-même est à la tête de la file d'attente à l'intersection, il s'arrête à un feu rouge avec une probabilité p ou avec $(1 - p)$ pour traverser. S'il s'arrête, la période de temps d'attente est aléatoirement choisie. Tout les nœuds qui sont derrière lui doivent attendre le temps restant plus une seconde (simuler le retard de démarrage).

À chaque fois que le signal devient vert, les véhicules démarrent un après l'autre à des intervalles d'une seconde, jusqu'à ce que la file d'attente soit vide. Le prochain véhicule qui arrive à la tête de la file d'attente prend encore une décision de s'arrêter ou de continuer selon le feu allumé.

II.5 Activity- based Model:

Ces modèles de mobilité ont été proposés par [20] qui l'ont classifié en quatre catégories: Constraint-based Models, Time and money allocation model, Utility-based models et Micro-Simulation Models.

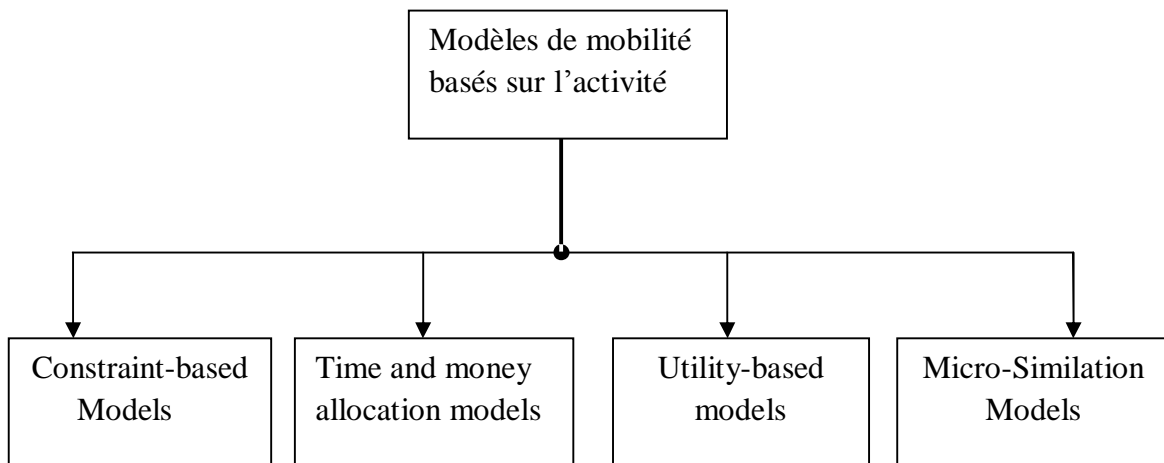


Figure II.11: Classification de modèles basés sur l'activité.

II.5.1 Time and money allocation model:

Ce modèle vise à définir le déplacement dans le contexte de participation dans les activités. Une personne participe à une activité selon son temps et le budget monétaire (les revenus).

La contrainte de temps définit le temps de déplacement et la durée de l'activité, ce temps est relié à la localisation de l'activité.

La disponibilité de budget monétaire permet d'acquérir des équipements nécessaires pour la participation à une activité ou pour dépenser l'argent directement dans une activité (regarder un film, manger dehors, etc). Pour cela la contrainte d'argent joue un rôle important dans la prise de décision concernant la participation à une activité. Si le revenu augmente, cela implique que le plus d'argent sera dépensé dans la participation à une activité.

II.5.2 Constraint-based Models :

Les modèles basés sur les contraintes représentent la première génération des modèles basés sur l'activité. La planification, l'organisation, et l'exécution des activités dépendent de diverses contraintes spatio-temporelles.

Trois types de contraintes ont été identifiés dans ce contexte: contraintes de capacité, des liens sociaux et d'autorisation.

Les contraintes de capacité représentent des limitations physiques posées sur une personne par son besoin biologique de sommeil, de manger, limite de vitesse de déplacement d'un certain mode de transport,... etc.

Les contraintes d'autorisation définissent les ressources accessibles pour effectuer une activité à un certain endroit à un certain temps. Par exemple, des heures d'ouverture des magasins, la disponibilité des sièges dans les restaurants, sont des contraintes typiques d'autorisation.

Les contraintes des liens sociaux se rapportent à la nécessité de l'individu de faire les contacts sociaux avec d'autres individus, par exemple, rencontrer des amis pour les récréations, assister à des réunions avec des clients, etc

II.5.3 Utility-based models (discrete choice model):

Ces modèles ont été utilisés dans la planification urbaine [22]. Ils sont fondés sur une théorie qui considère que les individus choisissent de consommer les produits et les services qui rapportent la plus grande quantité de satisfaction ou d'utilité. Ainsi, le choix représente un processus d'optimisation.

Les modèles basés sur l'utilité prolongent les modèles basés sur les contraintes en ajoutant le paramètre de choix pour prévoir des choix réels [24].

II.5.4 Micro-Simulation models :

Ce modèle est défini comme une approche pour modéliser un système dynamique [23]. L'argument d'utiliser le modèle micro-simulation est qu'une représentation analytique d'un tel système n'est généralement pas évidente. Ce type de modèles considère des unités de prise de décision de petit niveau telle que différentes personnes, familles et véhicules.

Plusieurs modèles basés sur l'activité se concentrent sur les déplacements pour l'activité quotidienne d'un individu et peuvent être regardés comme les modèles micro-simulation

II.6 Conclusion :

Nous avons abordé dans ce chapitre un des éléments les plus importants de la simulation des réseaux mobiles : les modèles de mobilité.

Tout d'abord nous avons vu des modèles qui définissent des déplacements individuels et qui sont valables pour les MANETs en général (Random Mobility Model, Random Waypoint, ...). Nous avons ensuite présenté quelques modèles à déplacement en groupe, qui permettent de modéliser des mouvements où les nœuds se déplacent en groupe. Nous avons enchaîné avec des modèles conçus pour les VANETs et nous avons terminé avec les modèles basés sur l'activité.

III.1 Introduction :

Dans le chapitre précédent nous avons présenté quelques modèles de mobilité existant dans les réseaux Ad hoc, au début nous avons parlé des réseaux MANET d'une manière générale puis des réseaux Ad hoc véhiculaires(VANET) qui prennent en considération les cartes routières, la vitesse et l'accélération...etc, et enfin les modèles qui se basent sur l'activité. Dans ce chapitre nous présentons deux modèles de mobilité : le modèle basé sur l'activité décrit dans [25] et le modèle Downtown de CityMob qui est un générateur de mouvements. Ces modèles permettent de représenter des scénarios de mobilité urbains pour simuler les déplacements en voitures.

Ce chapitre porte sur la description des deux modèles à implémenter avec les améliorations proposées.

III.2 Conception d'un modèle de mobilité:

L'élément de base des modèles de mobilité est le déplacement. Il est défini comme un mouvement d'une personne d'une origine à une destination en utilisant un moyen de transport. Le déplacement est souvent caractérisé par le but du déplacement, comme le travail, les achats ou le retour à la maison.

Un autre élément de base des modèles de mobilité est la zone de trafic. Elle représente une surface géographique allant de quelques mètres carrés à quelques kilomètres carrés. Elle correspond à une zone qui peut être l'origine ou la destination d'un déplacement comme une municipalité, un centre commercial ou une cité résidentielle.

Un modèle de mobilité est conçu en quatre étapes [26]:

III.2.1 Générateur de déplacements (ou de trafic) :

La première étape consiste à estimer le nombre total des déplacements T_i ayant pour origine la zone de trafic i et le nombre total de déplacements t_j ayant pour destination la zone de trafic j . En pratique, les déplacements sont classés en fonction de l'objectif du

déplacement, comme le travail, l'achat, ou le social/récréation, ...etc. La génération de déplacements se fait en fonction de la moyenne du trafic observé sur une période dans une zone de trafic ou en fonction de la moyenne des déplacements effectués par les différentes catégories de la population (travailleurs, étudiants, ...etc).

III.2.2 Distribution des déplacements :

Ayant le nombre total de déplacements originaires de la zone i , T_i , la seconde étape calcule la valeur de $T_{i,j}$ correspondant au nombre de déplacements de la zone de trafic i à toute autre zone j . Le nombre de déplacements ayant pour destination la zone j , (T_j) est également distribué sur toutes les autres zones.

III.2.3 Répartition sur les modes de transport (ou Modal split) :

Le Modal split consiste à déterminer le ratio (la probabilité) d'utiliser un mode de transport particulier (voiture, transport en commun, ou à pied) pour le déplacement $T_{i,j}$.

Cette troisième étape est très importante pour déterminer le chemin à prendre pour un déplacement car il peut exister des chemins différents selon que la personne se déplace à pied, en bus ou en métro.

III.2.4 Affectation des déplacements :

Etant donné le nombre de déplacements $T_{i,j,m}$ de la zone de trafic i à la zone j utilisant le mode de transport m , cette quatrième étape associe ce déplacement à une infrastructure de transport existante (ou proposée). Cette route associée est déterminée en fonction du coût général du déplacement qui est une combinaison du temps, de la distance, de la vitesse et de la préférence.

III.3 Description des modèles à implémenter :

3.1 Le modèle basé sur l'activité :

Le modèle de mobilité que nous avons implémenté est un modèle qui simule le déplacement d'individus dans une ville. C'est un modèle basé sur l'activité présenté dans [25]. Comme ce modèle est à l'origine conçu pour des réseaux cellulaires nous l'avons adapté aux simulateur J-Sim qui peut simuler les réseaux véhiculaires.

Ce modèle prend comme paramètres d'entrée des résultats des statistiques réalisées dans la région de Waterloo en 1987 [27], qui présentent les déplacements journaliers de plusieurs habitants de cette région, ces statistiques ont été obtenues par le remplissage d'un questionnaire de déplacement sur une période de 5 ans dans laquelle les détails des déplacements pris durant les sondages sont enregistrés.

Les informations issues des sondages ont été groupées dans une base de données contenant des détails appropriés.

Les données suivantes sont incluses dans chaque déplacement enregistré :

L'identifiant de la personne ;

Le temps de début de déplacement ;

Le temps de la fin de déplacement ;

La raison de déplacement (ou l'activité) de l'origine et destination ;

Le statut de l'individu qui se déplace (employé à mi-temps, employé à plein temps, employé dans ou hors la maison) ;

Lieu de travail fixe (vrai ou faux).

Chaque individu est caractérisé par plusieurs caractéristiques, la plus importante est le type de l'individu.

Les habitants sont classés en 4 catégories, cette catégorisation crée des groupes d'utilisateurs similaires avec des scénarios de déplacement similaires en utilisant les données collectées dans les statistiques.

Les quatre catégories sont :

- 0 : Employés à plein temps ;
- 1 : Employés à mi-temps ;
- 2 : Etudiants ou collégiens ;
- 3 : Non employés.

Le paramètre clés dans ce modèle est le but de déplacement des personnes qui est équivalent à l'activité dans notre cas, ce paramètre collecté dans les statistiques permet de classer les activités en 9 catégories :

- | | |
|-------------------------|------------------------|
| 1. Travail | 2. Relatif au travail |
| 3. Ecole | 4. Passager de service |
| 5. Achats | 6. Raison sociale |
| 7. Raisons personnelles | 8. Retour à la maison |
| 9. Autre | |

Dans ce modèle, le jour (24 heures) est divisé en 12 périodes, ces périodes sont utilisées pour récolter des informations obtenues du questionnaire de déplacement. La simulation d'une journée pour un individu particulier consiste à une séquence d'activités caractérisées par une durée et une localisation. Les chemins pris pour se déplacer d'un emplacement d'une activité à un autre associé au temps passé dans chaque activité forme les données de sortie exigées.

Le concept d'activité ou le but de déplacement est essentiel. Chaque activité a un horaire de début, une durée et une localisation.

L'activité courante est sélectionnée en se basant sur l'activité précédente et la période de temps courant. La probabilité de transition d'une activité à une autre utilise la matrice de transition d'activité dont le format est présenté dans la table.III.1.

Une fois la prochaine activité est sélectionnée et sa durée est déterminée en utilisant la matrice de la durée présentée dans la table.III.2 et la matrice d'Activité, la localisation de l'activité sera sélectionnée. Comme l'emplacement actuel de l'utilisateur et de l'activité sont connus, la route intermédiaire et la distance complète sont déterminées à partir de la table.III.3. En utilisant la vitesse moyenne, le temps total et le temps passé dans chaque cellule intermédiaire seront calculés. L'utilisateur reste dans la cellule destinataire pour la durée de l'activité et le processus se répète.

3.1.1 Les données d'entrées :

L'activité courante, la durée, l'emplacement et le chemin pris par l'utilisateur sont déterminés dans différentes tables qui sont des données en entrées pour le modèle étudié.

3.1.2 Matrice de transition d'activité :

La matrice de transition d'activité est l'une des tables de données extraite du questionnaire de déplacement.

La matrice de transition est une distribution empirique de transitions d'une activité à une autre, enregistrée sous forme d'un tableau à quatre dimensions, la distribution est indexée par la catégorie de la personne, l'horaire et l'activité précédente. La table Table.III.1 donne un aperçu sur la matrice de transition d'activité.

Groupe	période	activité précédente	activité future	probabilité
0	1	7	9	0.000000
0	1	8	1	0.857143
.....
1	4	5	2	0.052632
.....
1	7	1	5	0.242991
.....
.....
2	7	6	7	0.280488
.....
3	11	9	1	0.000000

Table III.1 : Aperçu de la table de transition d'activité

3.1.3 La matrice des durées :

La matrice des durées est aussi indexée par le type d'utilisateur, l'horaire et l'activité actuelle en donnant la distribution de la probabilité cumulative de différentes durées d'activités.

La durée moyenne de chaque activité est calculée à partir du sondage et regroupée dans cette matrice. Elle indique la durée probable d'une activité (en multiples de 5 minutes) en fonction de la catégorie des personnes et de l'horaire.

Groupe	période	activité précédente	durée	probabilité
0	0	8	220	0.857143
0	2	1	470	0.135279
0	3	1	365	0.318875
1	4	5	2	0.242991
.....
1	10	8	910	0.839506
.....	0.280488
.....
2	3	7	30	0.307692
.....
3	6	8	240	0.307292
3	6	8	255	0.312500

Table.III.2 : Exemple de données de la matrice des durées.

3.1.4 La matrice des distances :

Cette matrice contient la cellule origine, la cellule destinataire et la distance (en km) qui les sépare, elle sert à calculer le temps écoulé dans chaque cellule pour atteindre la destination.

Cellule origine	Cellule destinataire	distance
1	2	2.841
1	3	4.796
1	4	7.450
....
1	40	17.342
1	41	17.890
1	44	19.006
....
5	36	12.819
5	37	14.111
....
43	44	2.452
43	45	3.638
44	45	1.504

Table.III.3 : Exemple de données de la matrice des distances.

3.1.5 Le fichier de chemins :

Ce fichier contient la cellule origine, la cellule destinataire et les cellules intermédiaires ce fichier sert à déterminer le chemin à prendre pour atteindre la destination. Le plus long chemin est constitué de 30 cellules au maximum.

Cellule actuelle	Cellule future	chemin
1	2	1, 2
1	3	1, 2, 3
....
1	43	1, 5, 10, 15, 19, 20, 21, 25, 31, 32, 34, 37, 38, 40, 42, 44, 45, 43
....
....
18	43	18, 33, 21, 25, 31, 32, 34, 37, 38, 40, 42, 44, 45, 43
.....
22	36	22, 26, 27, 28, 30, 35, 36
.....
30	40	30, 28, 25, 31, 32, 34, 37, 38, 40

Table.III.4 : Exemple de données de fichier des chemins.

3.1.6 Déplacement géographique :

La région de Waterloo contient 345 zones de trafic, ces zones ont été regroupées en 45 cellules chaque cellule peut être originaire ou destinataire d'un déplacement. La figure Figure III.1 donne la décomposition en cellules et les chemins existant d'une cellule à une autre.

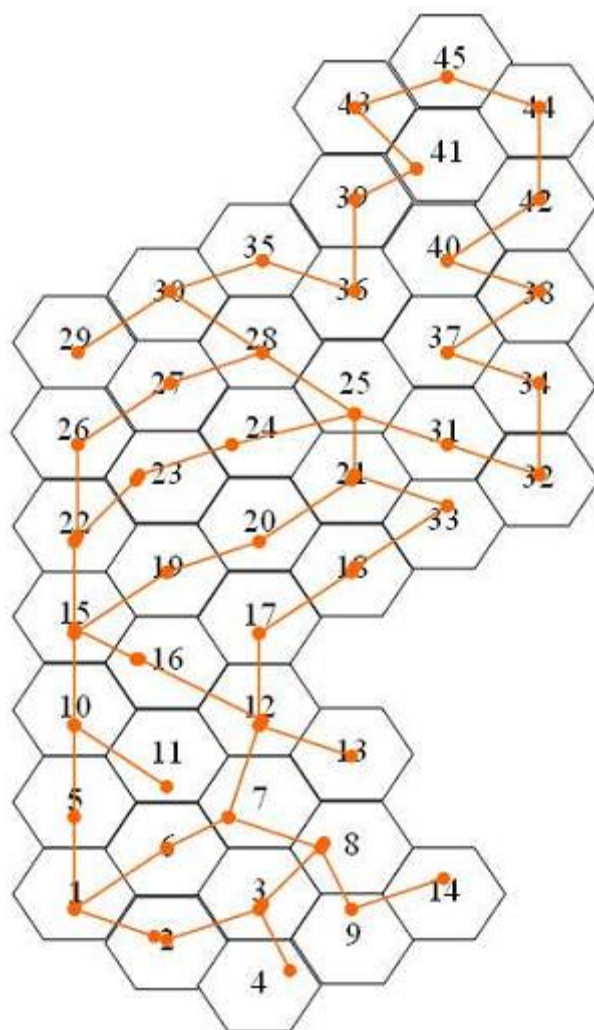


Figure III.1 : La décomposition de la zone simulée en cellules.

3.1.7 Informations sur les utilisateurs :

Dans la théorie de transport, les individus (ou groupes) sont catégorisés selon leurs paramètres par exemple le revenu, la possession de véhicule, etc. pour éviter toute complexité non nécessaire, et différencier entre les groupes, la variable `type` est définie pour chaque utilisateur. L'attribut `type` définit auparavant peut prendre l'une des quatre valeurs basée sur le statut.

Certaines cellules qui sont associées à l'utilisateur sont aussi incluses telle que la maison, l'école, le travail. Le travail peut être localisé dans une cellule fixe ou non, par contre la maison et l'école sont obligatoirement dans des cellules fixes. Ces informations sont utilisées pour déterminer la cellule destinataire qui correspond à l'activité sélectionnée. Ce modèle

incorpore la structure de déplacement quotidien des nœuds qui est complètement ignorée par le modèle de mobilité aléatoire (Random WayPoint). L'activité actuelle, la cellule et la zone de localisation sont aussi maintenues pour chaque utilisateur dans la simulation.

3.2 Le modèle Downtown :

Dans une vraie ville, le trafic n'est pas uniformément distribué ; il y a des zones avec une densité plus élevée de véhicules. Ces zones sont habituellement dans le centre ville, et les véhicules doivent se déplacer plus lentement que dans les périphéries. Les résultats des expériences faites par les chercheurs, ont montré que les véhicules se déplaçant dans le centre ville ont une vitesse aléatoire entre 25 et 60 km/h, alors que la vitesse dans les périphéries est habituellement plus haute.

Le centre-ville est défini par les coordonnées (min-X, max-X, min-Y, max-Y). La figure III.2 montre un exemple de ce modèle. Et à noter que le secteur plus foncé représente le centre ville. Les rectangles foncés représentent des véhicules, les rectangles clairs représentent des véhicules arrêtés au feu rouge, et les croix représentent les voitures endommagées envoyant les paquets d'avertissement. Il y'a pas une distinction entre les différents types des véhicules (voitures, camions ou taxis), mais seulement entre les véhicules normaux et endommagés.

Les nœuds s'arrêtent devant les feux tricolores lors du feu rouge. Les feux tricolores sont placés d'une manière aléatoire. De cette façon, la simulation est plus réaliste puisque dans une ville les feux tricolores ne sont pas systématiquement distribués sur chaque intersection.

L'utilisateur doit initialiser le nombre de nœuds, le temps de simulation, la taille de la carte, la valeur de vitesse maximale, la distance entre les rues consécutives et le nombre de nœuds endommagés.

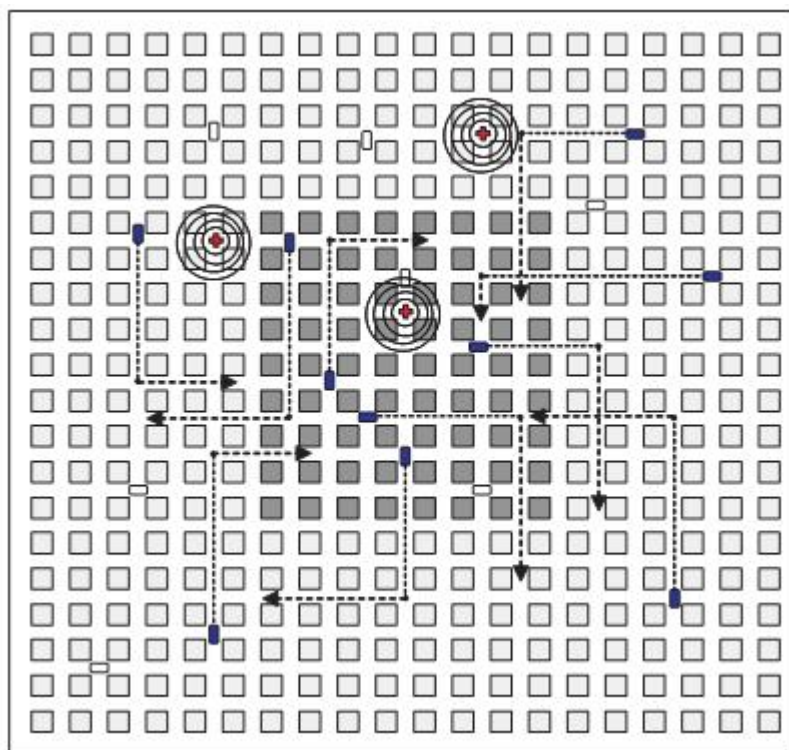


Figure III.2 : exemple de scénario dans le centre ville.

L'utilisation d'une grille pour la simulation de la zone de trafic rend ce modèle plus générique car il n'est pas dédié à une zone spécifique telle que le modèle basé sur l'activité présenté précédemment.

III.4 Critiques des modèles de mobilité étudiés :

Après avoir étudié profondément les deux modèles décrits précédemment, nous avons constaté qu'ils ne considèrent pas quelques cas particuliers des réseaux véhiculaires ce qui nous a incité à parvenir aux critiques suivantes :

4.1 Critiques de modèle de mobilité basé sur l'activité :

- Le paramètre de la vitesse est constant dans toutes les cellules durant toute la simulation et est fixé à 25 km/h. De ce fait le temps de traverser les cellules intermédiaires entre la cellule origine et la cellule destination est le même pour toutes les cellules ce qui n'est pas le cas dans le monde réel.
- ce modèle est développé pour modéliser les déplacements des mobiles dans la zone de Waterloo, ce qui l'a rendu très spécifique à cette zone. Donc ce modèle ne peut pas être généralisé sur n'importe quelle autre région.

4.2 Critiques de modèle de mobilité Downtown :

- Dans ce modèle les nœuds choisissent des destinations aléatoires, or que les nœuds se déplacent d'une position à une autre pour un but précis. Donc le choix d'une destination aléatoire ne reflète pas le vrai mouvement des véhicules dans la vie réelle.

III.5 Amélioration des deux modèles :

5.1 Amélioration du modèle d'activité :

Dans la réalité l'individu change sa vitesse fréquemment, donc la vitesse et le temps qu'il prend pour traverser d'une zone actuelle vers une zone future sont variables durant toute la simulation. En tenant compte de cette réalité nous avons modifié le modèle d'activité d'une façon à ce qu'il génère la vitesse de l'individu aléatoirement dans chaque passage de la zone actuelle vers la zone future et le temps écoulé pour s'approcher plus de la réalité, sachant que la vitesse maximale autorisée dans une ville est fixée à 60 km/h.

5.2 Amélioration du modèle Downtown :

Les utilisateurs se déplacent d'une localisation à une autre d'une façon aléatoire, ce qui n'est pas représentatif de la réalité, car il y a toujours une raison à un déplacement. Pour cela nous avons porté quelques modifications à ce modèle. En s'inspirant du modèle d'activité développé par [26], la destination est précisée en fonction de l'activité de l'utilisateur. Dans notre cas, comme les personnes se déplacent d'un lieu à un autre pour un objectif, nous avons introduit le principe d'activités dans ce modèle qui permettra de s'approcher plus de la réalité. A chaque individu est associé un ensemble d'activités qui définiront son profil de déplacement futur. Et à chaque lieu d'activité est associé des coordonnées symbolisant sa localisation dans la grille.

Nous avons classé ces activités en 4 catégories qui sont les suivantes :

Les études

Le travail

La santé

Le sport

Les informations décrivant ces activités sont décrites dans la figure suivante :

```
Activité : les études
Ecole 1 // l'emplacement de école 1
identifiant = 1
dest_x = 20 // coordonnée X
dest_y = 30 // coordonnée Y

ecole 2 // l'emplacement de école 2
identifiant = 2
dest_x = 80 // coordonnée X
dest_y = 70 // coordonnée Y

activité : travail//
travail 1// l'emplacement de travail 1
identifiant = 3
dest_x = 60 // coordonnée X
dest_y = 70 // coordonnée Y

travail 2 // l'emplacement de travail 2
identifiant = 4
dest_x = 80// coordonnée X
dest_y = 50// coordonnée Y

activité: la santé // l'emplacement de l'activité santé
identifiant = 5
dest_x = 50 // coordonnée X
dest_y = 40 // coordonnée Y

activité: le sport // l'emplacement de l'activité sport
identifiant = 6
dest_x = 20 // coordonnée X
dest_y = 90 // coordonnée Y
```

Figure III.3 Le fichier d'activités.

III.6 Conclusion :

Dans ce chapitre nous avons présenté les modèles Downtown et d'activité en décrivant l'ensemble des données qui participent à l'élaboration de ces modèles telle que la vitesse maximale autorisée, la distance séparant deux rues dans le modèle Downtown, et les matrices d'activités, de durée et les informations concernant les chemins à prendre pour atteindre la destination là où se situe l'activité à effectuer dans le modèle d'activité, ensuite, comme ces modèles n'ont pas pris en considération quelques spécificités des réseaux véhiculaires, nous avons critiqué ces modèles en quelques lignes et enfin nous avons proposé les améliorations à apporter pour rapprocher ces modèles plus de la réalité.

Dans le prochain chapitre nous aborderons la phase d'implémentation de ces modèles.

IV.1 Introduction :

Dans la plupart des simulateurs de VANETs le paramètre mobilité n'est pas pris d'une façon sérieuse. Le modèle Random Waypoint est le plus utilisé, bien qu'il propose des déplacements très simples et aléatoires. Un modèle de mobilité non représentatif de la réalité risque de donner des résultats erronés. Dans ce chapitre nous allons aborder la phase d'implémentation des modèles de mobilité, basé sur l'activité et Downtown. Ces modèles que nous allons intégrer dans le simulateur J-Sim simulent une zone urbaine et respectent les contraintes présentes dans une ville (les routes, les lieux publics, les feux de circulation, les intersections, ...).

IV.2 Environnement de la simulation :

1. Le simulateur J-Sim :

J-Sim [28] (appelé autrefois JavaSim, a été développé par une équipe du laboratoire Distributed Realtime Computing Laboratory (DRCL) de l'université d'État d'Ohio. Ce simulateur, développé entièrement en langage Java, repose sur une architecture logicielle basée sur les composants autonomes, appelée « Autonomous Component Architecture » (ACA). Les entités de base dans ACA sont les composants qui communiquent entre eux par l'envoi et la réception de données à travers les ports. Un composant est une entité indépendante représentant un objet physique (une batterie, un module radio, etc.) ou logique (un protocole de routage, un modèle de mobilité, etc.). Le composant peut être conçu, implémenté et testé individuellement [29] [30].

J-Sim possède plusieurs caractéristiques. Le fait que J-Sim est implémenté en Java, avec son architecture à base de composants, fait de J-Sim un environnement véritablement indépendant de toute plateforme et extensible. J-Sim fournit une interface de script qui permet son intégration avec différents langages de script tels que le Perl, le TCL, ou le python.

Par conséquent, J-Sim est un environnement de simulation qui utilise deux langages dans lequel les classes sont écrites en Java et liées en utilisant Tcl/Java [31]. Cependant, les classes et méthodes sont écrites en Java étant déclarées publiques n'ont pas besoin d'être explicitement exportées pour pouvoir y accéder dans l'environnement de TCL.

IV.1.2 Architecture de J-Sim :

  Le composant :

L'entit   de base dans l'architecture logicielle de J-Sim est le composant. Nous d  finissons une application en r  alisant une composition de composants. Les composants dans J-Sim sont faiblement connect  s. Ils communiquent entre eux en connectant leurs ports ensemble, et sont li  s par des contrats.

  Le port :

Un composant communique avec les autres composants par l'interm  diaire de ses ports. Il peut poss  der plus d'un port. L'interface de programmation entre un composant et son port est bien d  finie. Un composant peut   tre d  velopp   sans la pr  sence d'autres composants. En outre, le m  canisme r  el de communication qu'un composant emploie pour communiquer avec le reste du monde est compl  tement cach   par la notion de port.

  Le contrat

La communication entre composants est d  crite par les contrats. Le contrat impose les conditions sur les ports d'entr  e/sortie des composants afin de les faire communiquer. Un contrat indique comment un initiateur et un r  acteur accomplissent une certaine communication. Il d  crit comment un composant r  pond aux donn  es qui arrivent    chacun de ses ports.

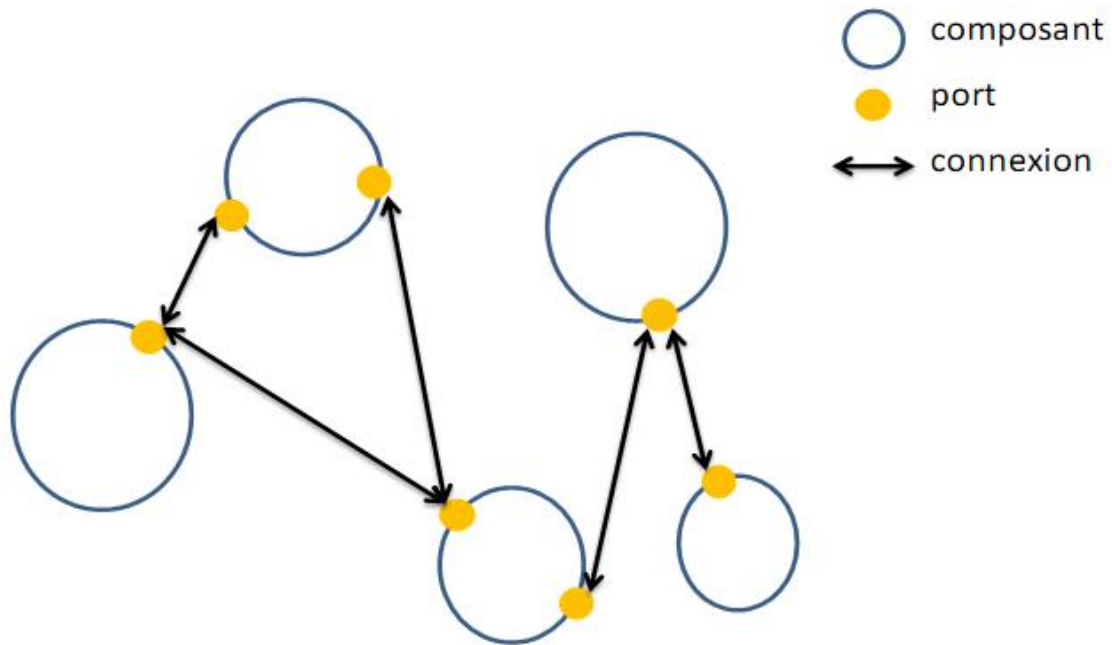


Figure IV .1 : Connexions entre composants dans J-Sim.

1.2 Langage utilisé pour définir un scénario de simulation :

Bien que J-Sim soit écrit en Java, les scénarios de simulation ne sont pas décrits en Java. La combinaison des langages Java et TCL ne facilite pas cette tâche. Pour développer un grand projet, il peut devenir encombrant d'employer des commandes TCL/Java parce que les références des objets Java doivent être stockées dans des variables TCL afin d'y accéder. Pour simplifier la syntaxe des simulations il existe un système appelé RUnTime Virtual (RUV) [32]. Ce système s'appuie sur la similitude entre les systèmes composants et les systèmes de fichiers d'UNIX. L'analogie entre un composant/port et un chemin de fichier permet d'accéder au composant de la même manière que l'on accède à un dossier dans un système de fichiers.

D’ailleurs des commandes systèmes UNIX, telles que `ls`, `cd`, `pwd`, `mkdir`, `cp`, `mv`, et `rm` peuvent être utilisées pour manœuvrer des composants et des ports dans le système. En résumé, des composants peuvent être créés et ils peuvent être liés grâce à une interface de commande qui utilise la syntaxe des invites de commandes (shells).

1.3 Mod  le de transmission dans J-Sim :

Comme la plupart des simulateurs de r  seau, J-Sim poss  de un mod  le propre de transmission.

L'utilisateur peut    tout moment impl  menter son propre mod  le ou modifier celui de J-Sim. Dans J-Sim, le champ de simulation est divis   en plusieurs sous-champs    deux dimensions. Chaque sous-champ est un rectangle de taille $dx \times dy$. Un n  ud d  finit sa port  e de transmission comme suit : il peut communiquer seulement avec les n  uds appartenant    son sous-champ et avec ceux dans les sous-champs voisins. La Figure VI.2 montre un n  ud A et ses n  uds voisins qu'il peut atteindre en un seul saut. Il peut communiquer avec les n  uds appartenant aux neuf sous-champs gris  s : le sous-champ o   il appartient et les huit sous-champs voisins au sien.

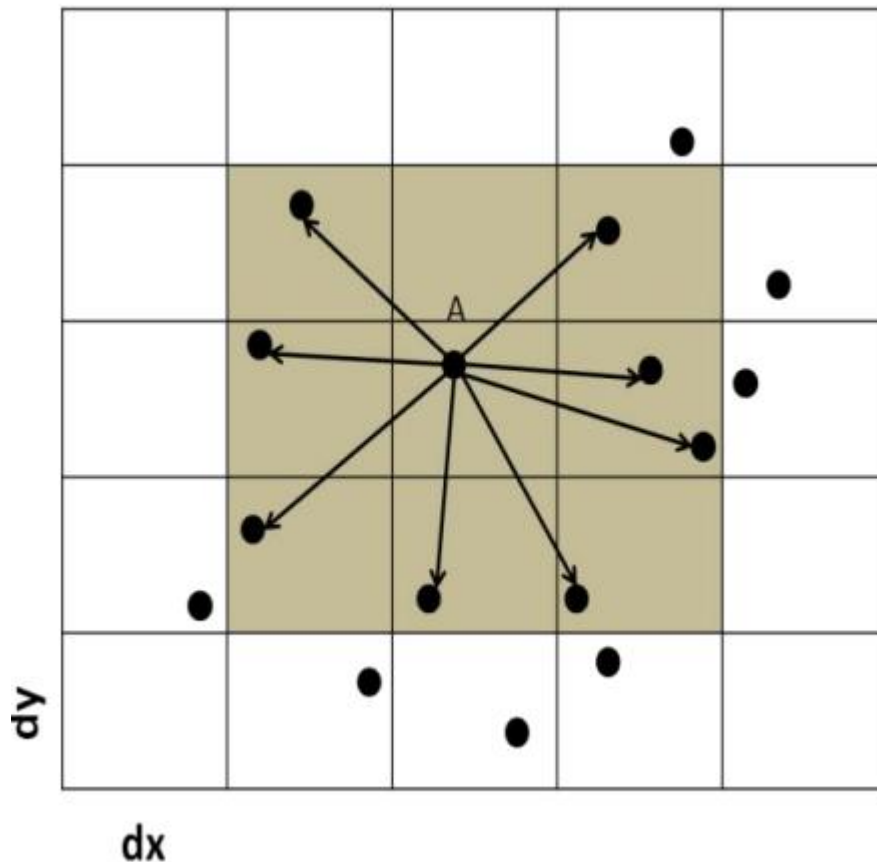


Figure IV.2 : Champs de voisinage dans J-Sim.

1.4 Installation de J-Sim :

Le fait que J-Sim soit d  velopp   en java lui offre l'avantage d'  tre un simulateur multiplateforme. Cela veut dire qu'il s'installe sur toute plateforme (Syst  me d'exploitation) disposant d'une machine virtuelle Java (JVM : Java Virtual Machine). Pour cela avant d'installer J-Sim, on doit disposer de cette machine virtuelle. L'installation de J-Sim est pr  sent  e dans l'annexe.

1.5 Int  gration du code source du protocole dans J-Sim :

Le code source du mod  le    simuler, qui doit   tre   crit en langage Java est copi   dans le r  pertoire /src de J-Sim,    condition de respecter le principe de d  veloppement impos   par le langage Java (notion de packages).

Pour faciliter le d  veloppement du code source, il est n  cessaire de choisir l'un des environnements de d  veloppements pour le langage Java. Il en existe plusieurs mais les plus populaires sont Eclipse IDE [33] et NetBean IDE [34]. Nous avons choisi NetBeans IDE pour sa facilit      cr  er les projets qui nous a permis d'ailleurs d'installer J-Sim dans le r  pertoire /project de NetBeans IDE. L'ensemble de J-Sim et NetBeans combin  s. Nous proc  dons comme suit pour impl  menter notre mod  le:

Nous cr  ons un projet sous NetBeans. Nous cr  ons ensuite les packages (au sens Java) de notre projet qui sont dans le r  pertoire /src.

Une fois le code source est sauvegard  , sous netBeans, nous le compilons par la commande « ant compile » (voir guide d'installation de J-Sim).

1.6 Connexion entre les composants :

A l'aide de TCL, on d  finit les composants puis on les connecte. Tous les composants sont h  berg  s dans un conteneur, qui est    son tour un composant. La d  finition des composants est en fait la cr  ation des objets. Cette cr  ation est r  alis  e par la commande TCL *mkdir*. Chaque composant est une entit   ind  pendante qui fonctionne ind  pendamment des autres entit  s. Les composants poss  dent des ports par d  faut pour qu'ils puissent communiquer entre eux. D'autres ports peuvent   tre cr   s pour un composant. Une connexion

entre deux composants est réalisée par l'intermédiaire de deux ports dédiés, un dans chaque composant (voir la Figure IV.1). Cette connexion est établie par la commande TCL *connect*. Il suffit de suivre un schéma qui indique l'interconnexion entre les différents types de composants que l'on veut utiliser. Toujours dans ce script TCL, on y définit les paramètres globaux (par exemple la taille du champ de simulation), les outils de visualisation des résultats et l'ordonnancement de la simulation.

1.7 Implémentation des modèles dans j-sim :

Pour implémenter les deux modèles dans J-Sim nous avons utilisé l'architecture à base de composants offerte par ce dernier.

Chaque modèle est implémenté sous forme d'un composant. Tous les deux sont mis dans le conteneur « *component* » défini dans J-Sim. Pour connecter ces deux composants aux autres, nous définissons les deux ports pour chaque modèle qui sont :

Le port *query* : établit la connexion du composant avec l'interface « *wirelessphy* » du nœud mobile, cette interface est un composant qui décrit certaines caractéristiques de la couche physique.

Le port *report* : ce port sert à connecter le composant avec le composant *NodePositionTracker* pour obtenir les changements de position effectués par le modèle de mobilité.

IV.3 Implémentation des modèles de mobilité étudiés :

1. Le modèle basé sur l'activité :

En utilisant les détails du modèle décrits dans le chapitre précédent, nous avons intégré ce modèle dans le simulateur J-SIM. La génération des déplacements se fait en développant des événements discrets et en ordonnant ces événements dans une file d'attente en fonction de leurs ordres chronologiques. Les types des événements utilisés sont l'évènement de génération de déplacement et l'évènement de changement de zones. L'occurrence d'un type particulier déclenche la fonction correspondante. La simulation commence avec l'évènement de génération de déplacement où les paramètres d'entrées sont sélectionnés par l'utilisateur.

La fonction de g  n  ration de d  placement cr  e et met en file d'attente l'ensemble des   v  nements de changement de zones. Ces   v  nements correspondent aux zones travers  es par l'utilisateur quand il se d  place pour une activit  . Quand un   v  nement de changement de zones se produit, la fonction de changement de zones met    jour les fonctions de localisation de l'activit  .

La fonction qui g  n  re le d  placement cr  e et met en file d'attente une autre fois l'  v  nement de changement de zones qui correspond    la dur  e de l'activit   destinataire.

1.1 L'  v  nement de g  n  ration de d  placement :

Cet   v  nement produit le mouvement de l'utilisateur durant la simulation.

Les entr  es de la fonction sont : l'activit   pr  c  dente, type de l'utilisateur, l'horaire actuel et la matrice de transition d'activit   sont utilis  s pour d  terminer la prochaine activit  .

La fonction de changement de zones utilise la prochaine activit   et le fichier de chemins pour g  n  rer la localisation de cette activit  . Cette fonction sera aussi employ  e pour d  terminer la dur  e de l'activit   en utilisant la prochaine activit  , le type d'utilisateur, l'horaire actuel et la matrice de dur  e. En utilisant la localisation, la dur  e de la prochaine activit  , le chemin qui y m  ne, la matrice des distances et la vitesse maximale qui permet de g  n  rer une vitesse al  atoire, le temps pass   dans chaque zone sera calcul  .

Le temps pass   dans chaque zone est calcul   en divisant la distance entre deux zones successives sur la vitesse de d  placement dans chaque zone. Pour chaque changement de zone, lui correspond un temps qui   gale    l'horaire actuel plus le temps de la travers  e de la zone. Finalement, l'  v  nement de d  placement sera g  n  r   avec un temps d'  v  nement   gal au temps total de d  placement vers la destination plus la dur  e de l'activit   (une fois l'activit   s  lectionn  e est termin  e une nouvelle activit   sera entam  e).

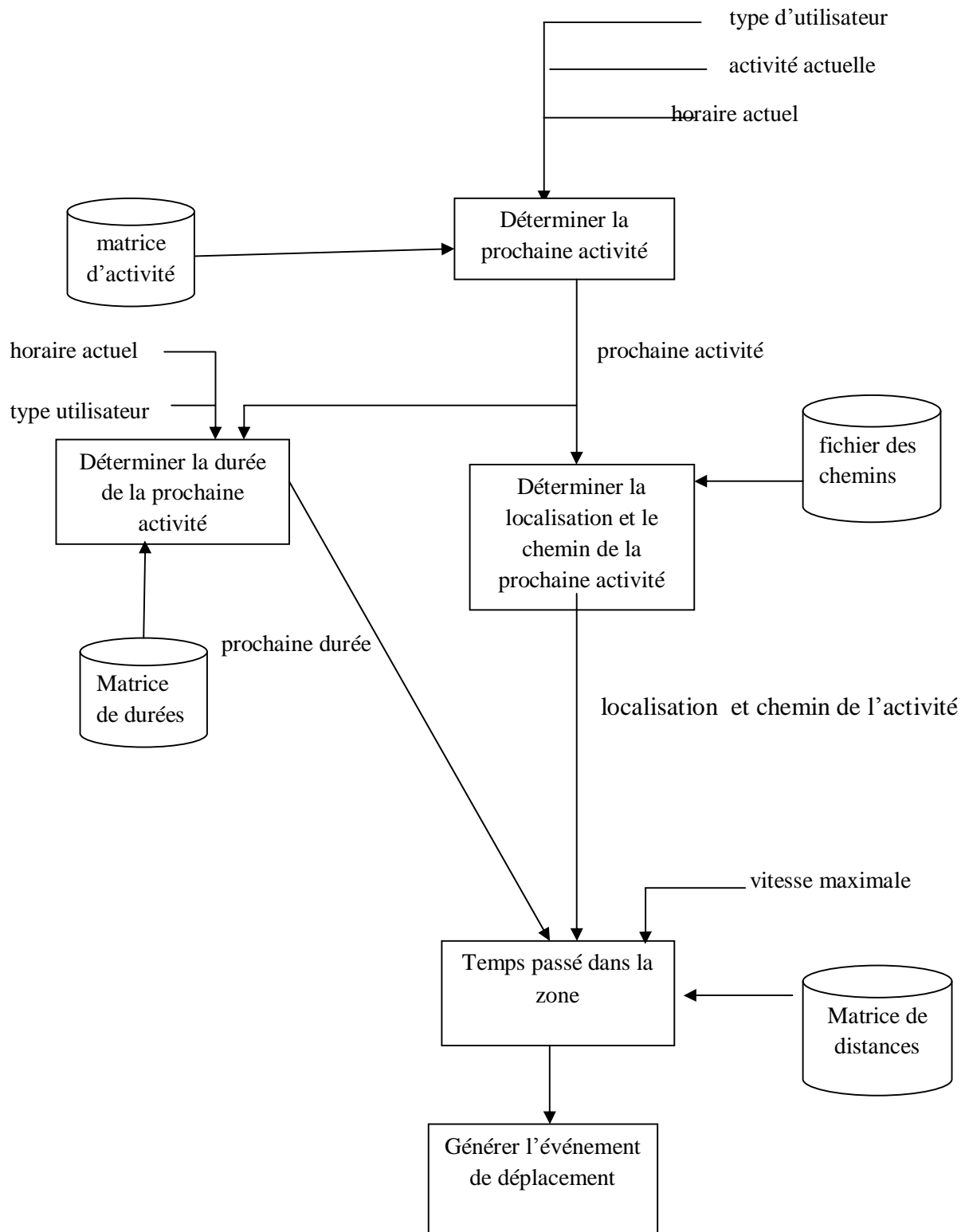


Figure IV.3 : Fonctionnement du m  d  le d'activit  

La d  termination de l'activit   suivante de l'utilisateur conduit    la g  n  ration de la mobilit   quotidienne, comme le montre la figure suivante :

1.2 La description des classes :

Notre mod  le est impl  ment   en langage java et compos   des classes suivantes :

   La classe *MobilityEvent* :

Cette classe est utilis  e pour rassembler les classes permettant de g  n  rer la classe *EventTraversZone* qui produit l'  v  nement de d  placement men   par un utilisateur qui sera mis dans une file d'attente.

   La classe *EventTraversZone* :

Quand l'  v  nement *EventTraversZone* se produit (c.  .d, l'  v  nement est tir   de la file d'attente d'  v  nements, et sa m  thode *Handler()* est appel  e), les m  thodes *setCurLocation()* et *setCurZoneEntryTime()* de la classe *Utilisateur* sont appel  es pour mettre    jour l'emplacement courant et le temps d'entrer dans la zone pour un utilisateur.

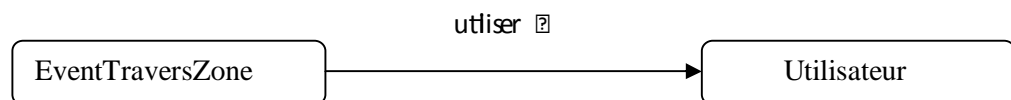


Figure IV.4 : Sch  ma fonctionnel de la classe *EventTraversZone*.

   La classe Utilisateur :

Ce mod  le de mobilit   d  crit le mouvement individuel des utilisateurs, donc certaines informations doivent   tre maintenues pour chaque utilisateur. Pour diff  rencier entre les utilisateurs la variable type d'utilisateur est d  finie. Aussi l'emplacement de chaque utilisateur est maintenu. L'encapsulation des donn  es sp  cifiques de chaque utilisateur permet de simuler plusieurs utilisateurs en m  me temps.

   La classe ZoneGeography :

Cette classe contient des informations concernant une zone, tel que le chemin et la distance qui sont lus    partir des fichiers externes *shortestdist.txt* et *shortestpath.txt*. Cela est n  cessaire pour cr  er les chemins incluant les zones interm  diaires pour tous les d  placements g  n  r  s, cela permet de conna  tre    tout moment la zone o   se trouve l'utilisateur.

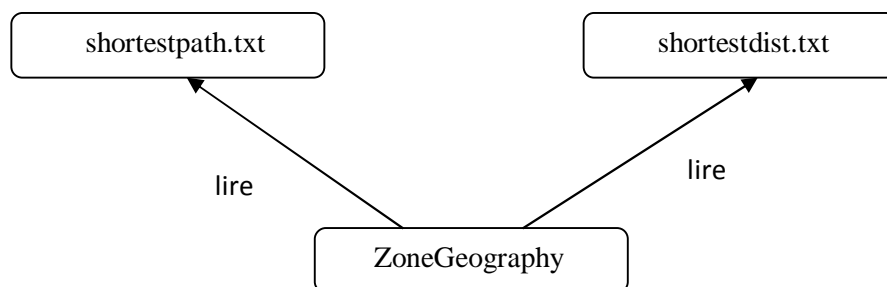


Figure IV.5 : Sch  ma fonctionnel de la classe ZoneGeography.

   La classe Mouvement :

Cette classe permet de r  cup  rer les param  tres d'entr  e dans le fichier mobilite.ini. Pour g  n  rer les d  placements la m  thode *redmov()* appelle la m  thode *setCurCellEntryTime()* de la classe *Utilisateur* , la m  thode *addEvent()* et la m  thode *handler()* de la classe *EventDepl* pour   crire les r  sultats de la simulation dans le fichier trace.

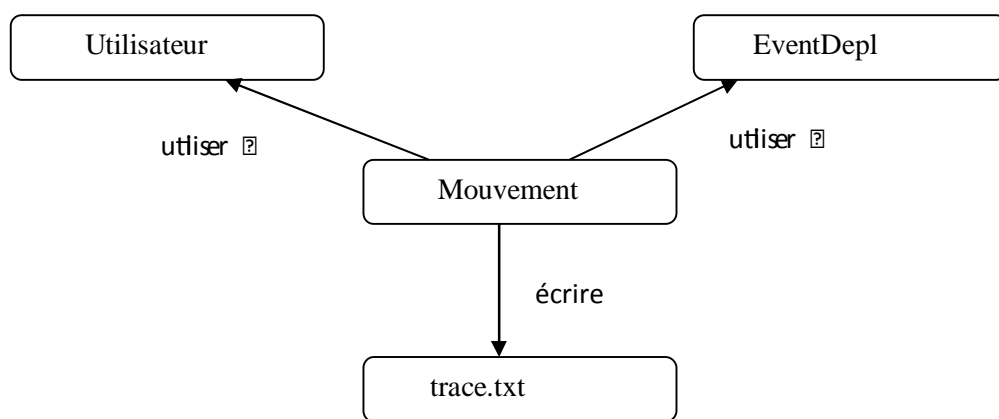


Figure IV.6 : Sch  ma fonctionnel de la classe Mouvement

   La classe Deplcement :

Cette classe permet de r  cup  rer les informations concernant les activit  s et leurs dur  es en initialisant la matrice d'activit   et la matrice de dur  e avec la m  thode *initDeplcementMatrices()* et en d  terminant la prochaine activit   avec la m  thode *getNextActivity()* et sa dur  e avec la m  thode *getNextDuration()*.

   La classe EventScheduler :

Elle utilise la m  thode *initEventScheduler()* pour r  cup  rer un   v  nement et l'ajouter dans la file des   v  nements dans un ordre chronologique ascendant avec la m  thode *AddEvent()*.

   La classe principale Mobilit   :

Elle permet d'initialiser l'ensemble des utilisateurs et de g  n  rer les d  placements caract  ris  s par une zone origine, une zone destinataire, l'activit   courante, le temps   coul   et la vitesse de l'utilisateur dans chaque zone dans le fichier trace.

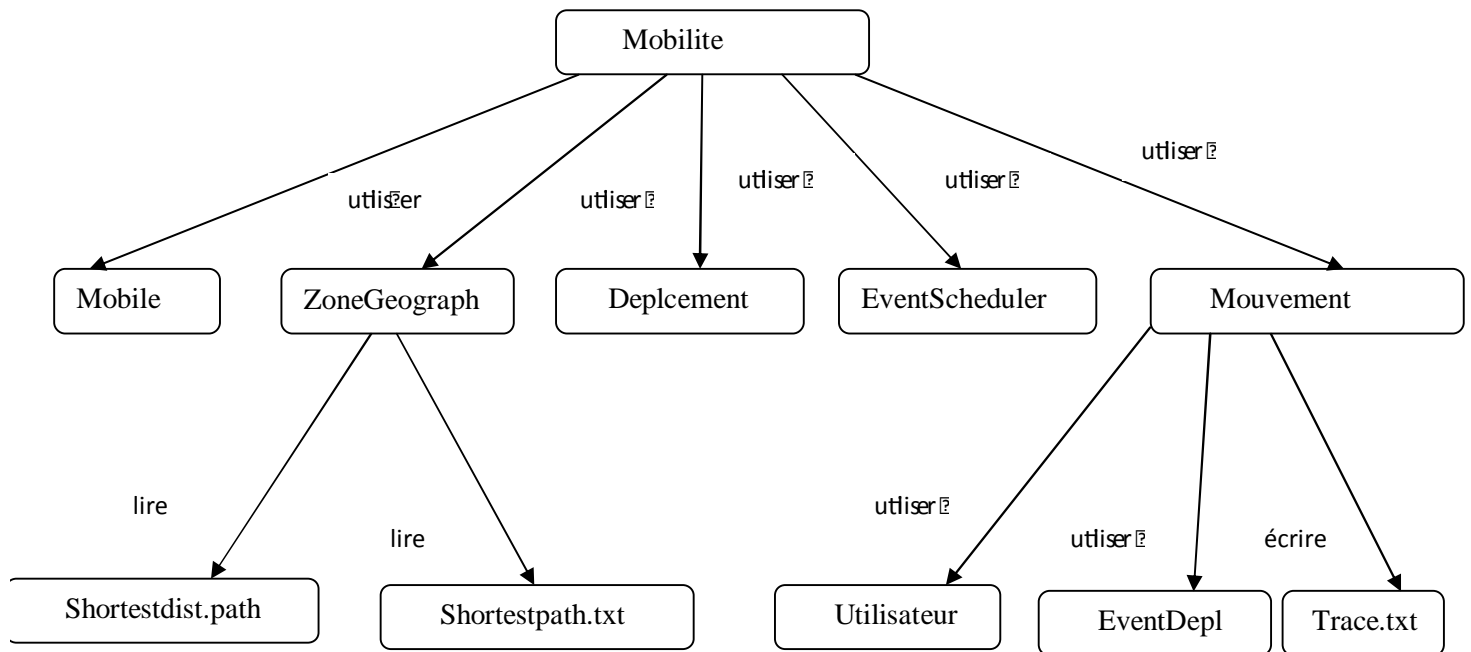


Figure IV.7 : Le sch  ma fonctionnel de la classe Mobilit  

2 Le mod  le Downtown:

En utilisant les param  tres d'entr  es tels que le nombre de n  uds, la vitesse maximale autoris  e dans la ville, le temps de la simulation, la taille de la carte et le fichier contenant les activit  s...etc, l'  v  nement de la simulation se produit.

La position initiale des n  uds est d  termin  e al  atoirement.    chaque fois qu'un n  ud atteint sa destination, une nouvelle destination est choisie selon l'activit   future tant que le temps de la simulation n'est pas termin  .

L'ensemble des   v  nements produit seront mis dans la liste des   v  nements.

2.1 L'  v  nement de g  n  ration de d  placement :

Cet   v  nement repr  sente le d  placement des n  uds durant la simulation.

Le nombre de n  uds, la vitesse maximale autoris  e, le fichier contenant les activit  s...etc; sont utilis  s en entr  e pour g  n  rer l'  v  nement de d  placement.

Au d  but de la simulation, les n  uds sont plac  s al  atoirement en supposant que l'emplacement initial est la maison. Apr  s avoir d  termin   la position initiale, la destination est s  lectionn  e selon les activit  s   voqu  es pr  c  demment et la vitesse de d  placement sera choisie aussi. La direction du n  ud peut   tre soit UP, DOWN, LEFT ou RIGHT suivant les coordonn  s de la destination.

A chaque intervalle de temps, on compare la position actuelle avec la destination. Si la position est la m  me, donc la position est atteinte sinon on choisit une autres vitesse en prenant en consid  ration la vitesse des n  uds qui sont devant et derri  re le n  ud concern  , la pr  sence des feux tricolores qui sont utilis  s pour marquer une pause (pour la couleur rouge)ou continuer le mouvement (pour la couleur verte) et les intersections .

La fonction permettant le changement de vitesse contr  le le mouvement des n  uds, si la vitesse de n  ud concern   est plus   lev  e que celle de n  ud devant, cela conduit    v  rifier la distance entre les deux n  uds. Si elle est inf  rieure    la distance minimale qui est 5 m  tres il aura un risque de collision.

La fonction qui d  finit les intersections permet de contr  ler si le n  ud se trouve pr  s d'une intersection. Si le cas se pr  sente, la fonction tricolore() est utilis  e afin d'  viter toute collision en changeant ou en restant dans la m  me direction.

Dans le cas o   le temps d'arriver    la destination est sup  rieur    celui de la dur  e de la simulation, le mouvement de g  n  ration de d  placement sera interrompu.

Une fois le n  ud a atteint sa prochaine destination, une autre activit   sera choisie pour la g  n  ration d'un nouveau d  placement. Ce processus continue jusqu'   l'ach  vement de temps de la simulation. Un sch  ma r  capitulatif de ce mouvement est pr  sent   ci-dessous.

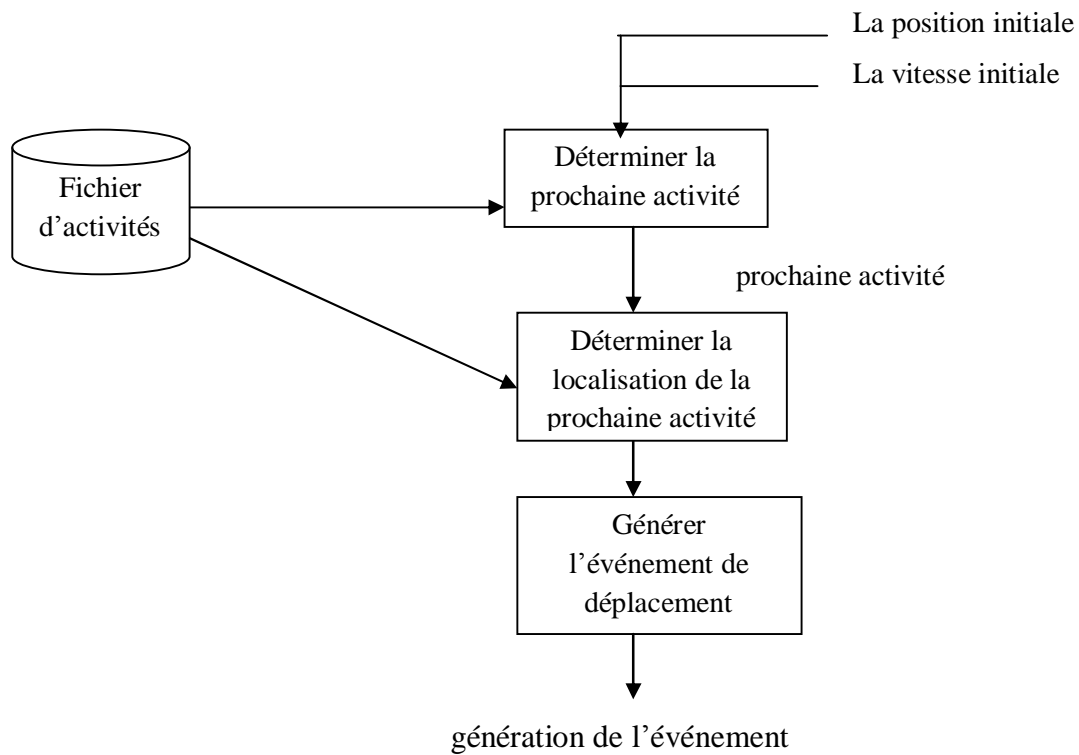


Figure IV.8 : sch  ma fonctionnel de mod  le Downtown

2.2 La description des classes :

2.2.1 La classe Control :

Cette classe est constitu  e de la fonction d  but() qui permet de cr  er le fichier trace contenant les d  placements, et fait appel aux fonctions initializeSimulation(), generateStaticScenario() et startSimulation() de la classe simulation. La premi  re fonction initialise la simulation avec les param  tres. La deuxi  me fonction permet de d  terminer la position des n  uds .et la troisi  me g  n  re le d  placement des n  uds.

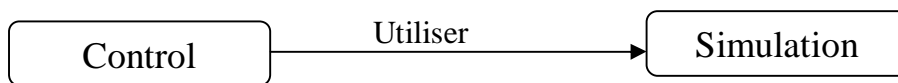


Figure IV.9 : Sch  ma fonctionnel de la classe Control.

2.2.2 Classe simulation :

La simulation commence par la d  termination de la carte de simulation en d  finissant les routes horizontales et verticales. On d  finit   galement la distance entre les routes, puis on g  n  re al  atoirement la position initiale des n  uds. Cette position est consid  r  e comme lieu d'habitation. Enfin l'  v  nement de g  n  ration de d  placement est d  clench  .

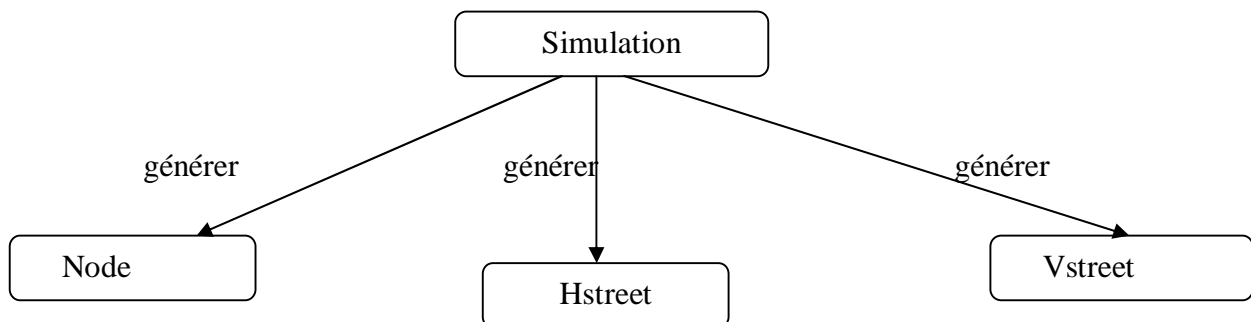


Figure IV.10: Sch  ma fonctionnel de la classe Simulation.

2.2.3 La classe Vstreet/Hstreet :

Ces deux classes sont héritées de la classe Street. Les contrôles d'ajout ou de suppression d'un nœud sont effectués avec la fonction Add(), et Remove(). Quand un nœud essaie de changer la vitesse, il doit vérifier la vitesse des nœuds situés devant et derrière lui afin d'éviter toute collision.

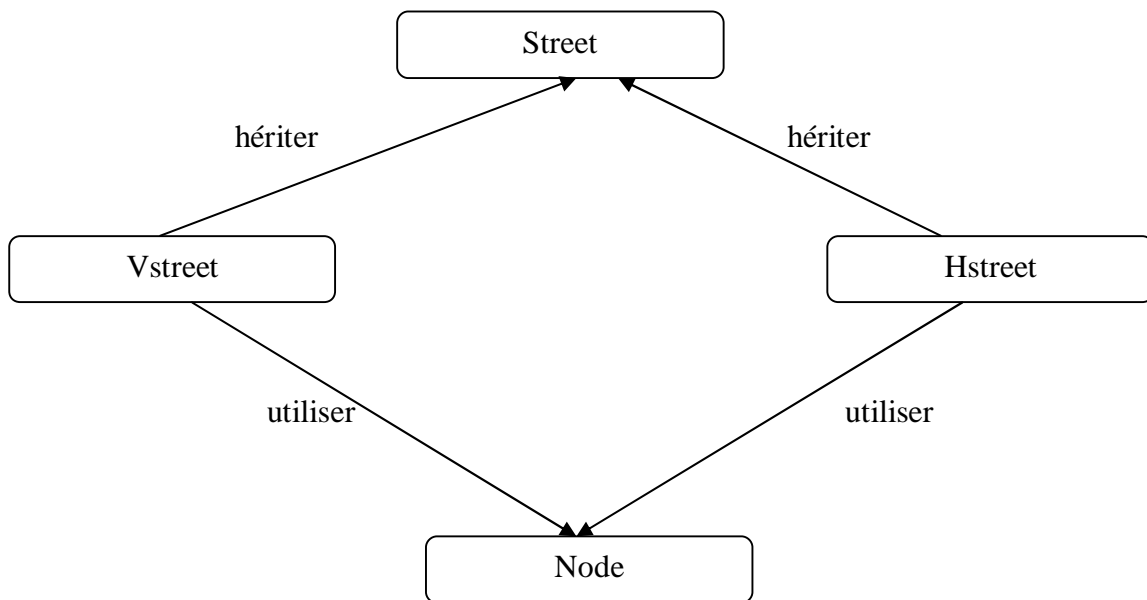


Figure IV.11: Schéma fonctionnel de la classe Vstreet/Hstreet.

2.2.4 La classe Node :

Les informations concernant les activit  s sont obtenues    partir de fichier destination.ini. La classe Node utilise ces informations pour d  terminer la prochaine activit  . La position future de n  ud est d  finie en g  n  rant la classe Position. La classe Node g  n  re en sortie l'  v  nement de d  placement dans le fichier trace.txt.

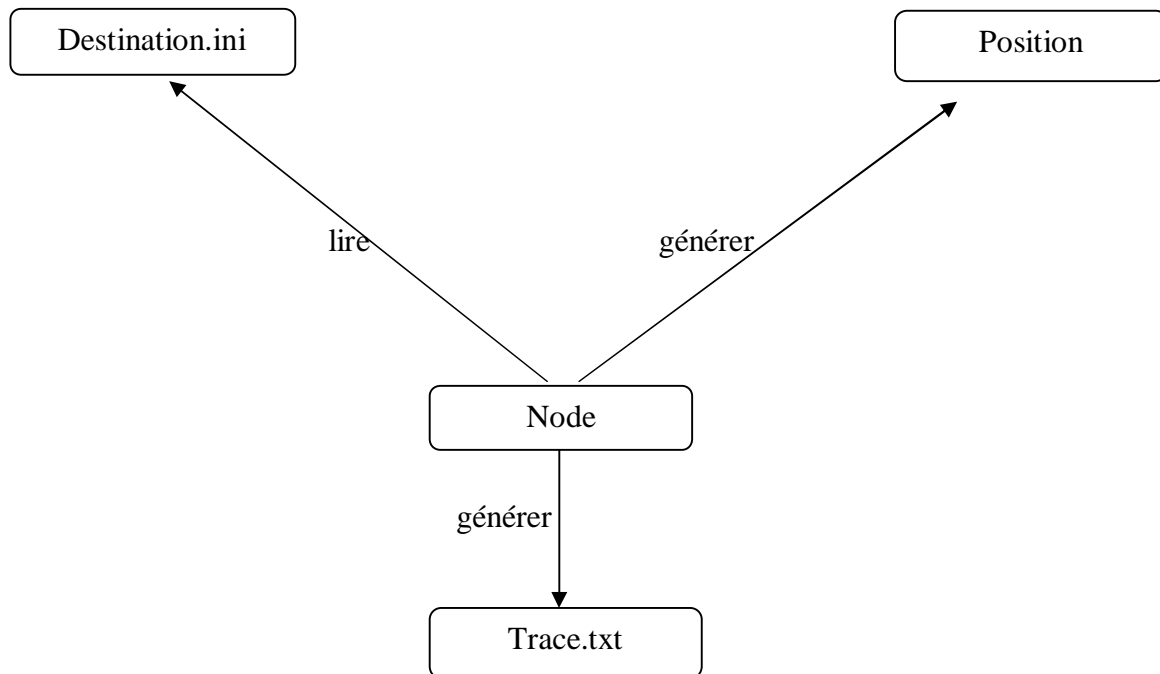


Figure.IV.12: Sch  ma fonctionnel de la classe Node.

2.2.5 La classe principale Mobility :

Elle appelle la m  thode debut() de la classe Control qui    son tour appelle les fonctions des autres classes tel qu'il est mentionn   auparavant. L'appel de ces classes permet de g  n  rer le fichier trace.

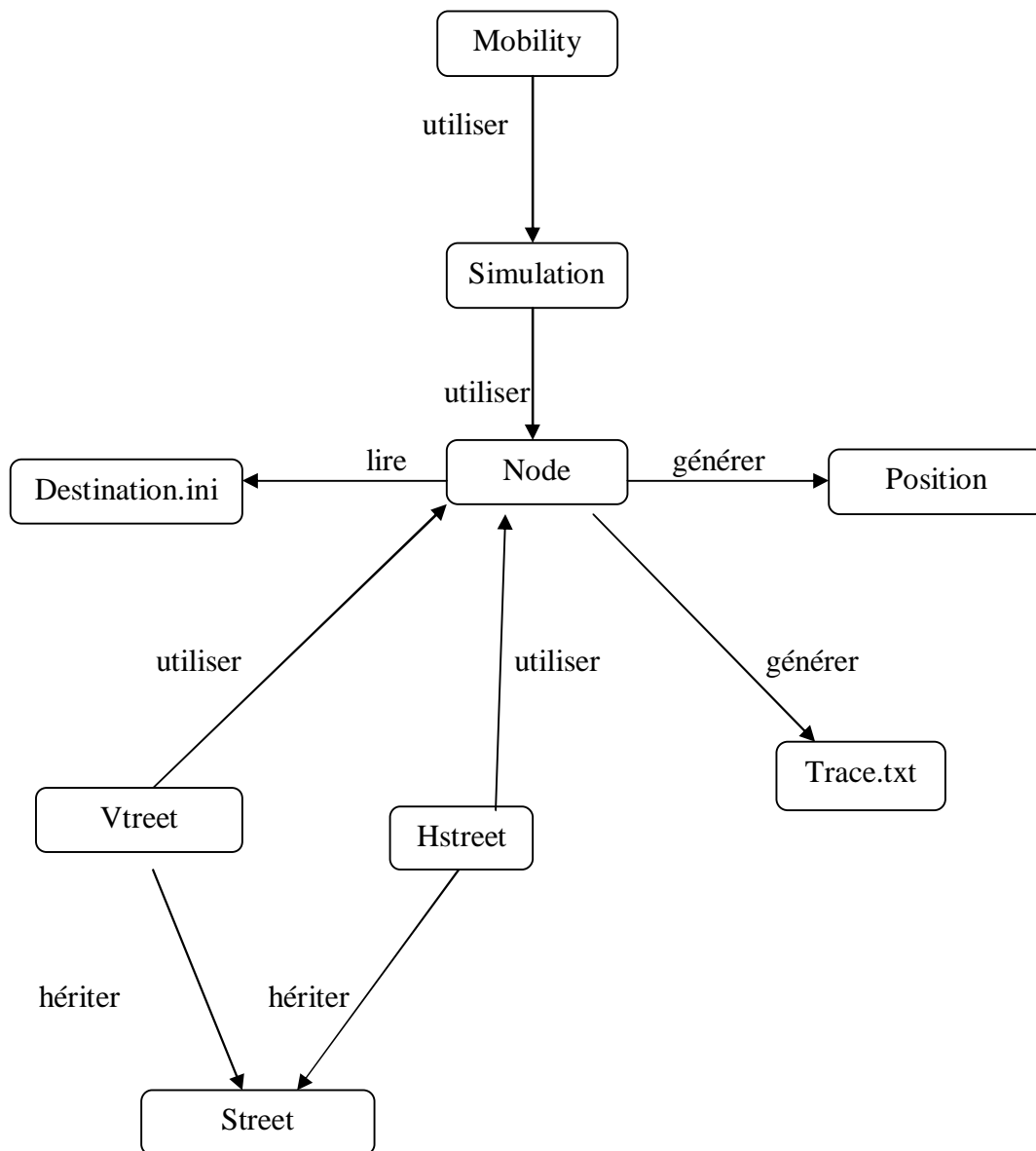


Figure IV.13 : sch  ma global de la classe Mobility.

IV.4 Ex  cution et simulation :

Pour lancer la simulation, nous avons deux possibilit  s :

Charger le fichier de configuration de m  d  le d'activit   ou Downtown via l'interface RUV, ou utiliser la commande « java drcl.ruv.System -u nom_fichier_de_configuration ».

Le nom du fichier de configuration est de format *.tcl

IV.5 Tests des mod  les de mobilit   :

5.1 Le mod  le bas   sur l'activit   :

1.1 Les param  tres de la simulation :

1. Le temps de la simulation :

Ce param  tre permet de sp  cifier le temps de simulation.

SIMULATION DURATION 1440 s (une journ  e)

2. La zone de la simulation:

La surface de la simulation est divis  e en 45 zones, chacune pouvant renfermer un lieu de travail, un segment de route, ...etc.

3. Nombre d'utilisateurs : Nous prenons un exemple de 10 utilisateurs.

4. Type d'utilisateur :

Les utilisateurs simul  s sont des travailleurs.

5. Emplacement initial des utilisateurs :

L'emplacement initial des utilisateurs est la maison situ  e dans des zones choisies d'une fa  on al  atoirement.

1.2 Les r  sultats de la simulation :

Apr  s le d  roulement de la simulation le fichier trace sera g  n  r  . Les r  sultats de la simulation sont pr  sent  s dans ce qui suit :

L'utilisateur 2 passe de la zone 45    la zone 17 au temps 390 mn.

user	de	��	la vitesse est:	le temps de traverser	temps total
2	44	42	43.89500600112207	4.37213759570317	394.3721375957032
2	42	40	35.48788251548792	3.586012773358909	397.5860127733589
2	40	38	7.744164964094446	13.984722756053005	410.98472275605303
2	38	37	12.273698493949647	14.709502607465685	424.7095026074657
2	37	34	36.041124838097495	4.631375983680625	428.63137598368064
2	34	32	24.23869258797202	3.1536354414566015	431.15363544145663
2	32	31	35.280273840955886	5.0050631918568955	436.0050631918569
2	31	25	9.068880396327577	30.182336522030027	466.18233652203
2	25	21	20.72946004025331	4.628195805086085	470.6281958050861
2	21	33	4.446821922648452	80.60588128667838	550.6058812866784
2	33	18	35.88284704027979	12.211405619741576	562.2114056197415
2	18	17	47.21764787614674	3.2555624202885327	565.2555624202886

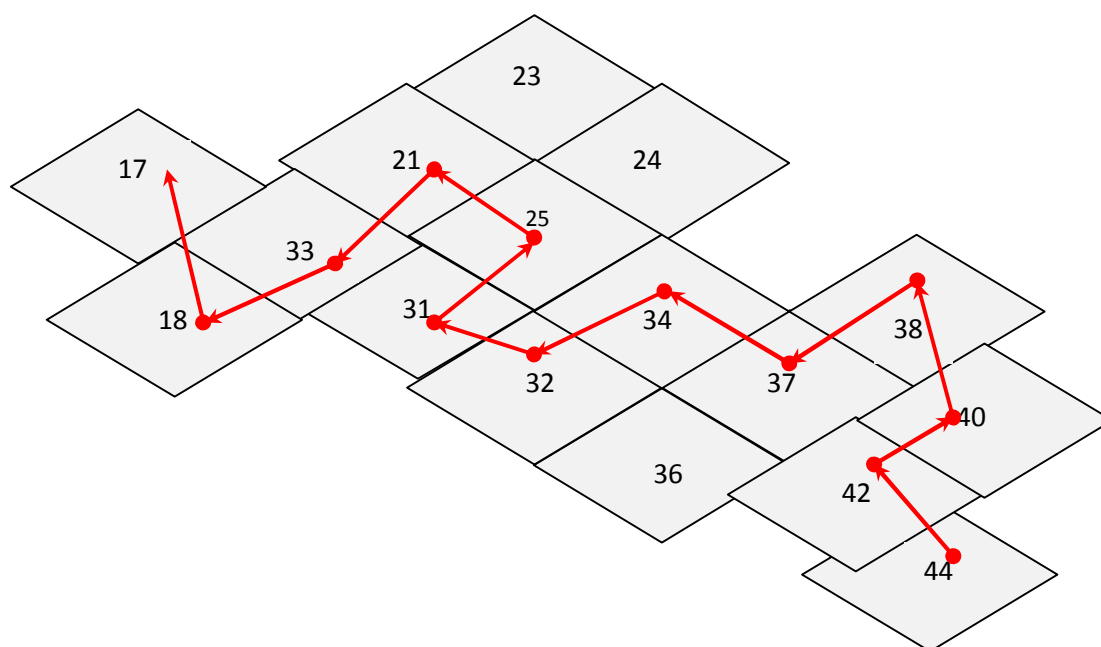


Figure IV.14 : Repr  sentation de l'itin  raire de l'utilisateur num  ro (2).

5.1 Description de l  v  nement :

L  utilisateur (2) se d  place selon sa prochaine activit  , dans cet exemple son activit   future sera (5 : Raison sociale), en commen  ant le mouvement    partir de la zone actuelle qui est (44 c  est la localisation de la maison sachant que c  est l  emplacement initial de l  utilisateur) et finira en atteignant la zone destinataire (17) l      o   il effectue son activit  . Mais avant d  atteindre la zone destinataire il passe par plusieurs zones interm  diaires dans lesquelles la vitesse de d  placement et le temps de passage d  une zone    une autre sont calcul  s.

Au temps 390 mn, l  utilisateur commence le d  placement de la zone 44    la zone 42 avec une vitesse de 43.89500600112207 km/h et une dur  e de 4.37213759570317 mn, puis de la zone 42    la zone 40 avec une vitesse de 35.48788251548792 km/h et une dur  e de 3.586012773358909 mn, ensuite, de la zone 40    la zone 38 avec une vitesse de 7.744164964094446 km/h et une dur  e de 13.984722756053005 mn et ainsi de suite jusqu      l  atteinte de la destination.

L  utilisateur effectuera un autre d  placement pour une autre activit   tant que le temps de la simulation n  est pas fini, c  est la m  me chose pour tous les autres utilisateurs.

2 le mod  le Downtown :

2.2 Les param  tres de la simulation :

1. Le temps de la simulation :

Ce param  tre permet de sp  cifier le temps de simulation.

Maxtime 60 mn.

2. La zone de la simulation:

La zone de la simulation est une zone urbaine sous forme d  une grille de dimension 100X100m.

3. Nombre de n  uds : Nous prenons un exemple de 50 n  uds.

4. Type de n  ud :

Les n  uds sont des travailleurs.

3. Emplacement initial des n  uds :

L'emplacement initial des n  uds est la maison situ  e dans des points choisis al  atoirement dans la grille.

2.2 Les r  sultats de la simulation :

L'  v  nement de la simulation est g  n  r   dans le fichier trace. Les r  sultats sont affich  s comme suit :

Les param  tres d'entr  e sont les suivants :

nodes number= 20, max time= 60.0, max_speed= 60.0, max X= 100.0,
max Y= 100.0, d= 20.0, alpha= 0.5, delta= 5.0

NODE 9 MOVEMENTS

#Node 9: TARGET REACHED (20.0, 30.0) (42.40219773865657 Km/h) activite: etude dans
l'  cole 1 direction : DOWN

at 2.4098420225013286 "\$node_(9) setdest 20.0 30.0 42.40219773865657"

#Node 9: feu rouge jusqu'   4.160963827285221s

#Node 9: feu vert

#Node 9: NEW TARGET (60.0, 70.0)

at 4.2 "\$node_(9) setdest 20.0 30.0 0.0 activite: travail 1 direction : RIGHT UP"

#Node 9: SPEED CHANGE (0.0 Km/h --> 59.48896544688752 Km/h)

at 4.45 "\$node_(9) setdest 40.11861153781857 30.0 59.48896544688752

#Node 9: SPEED CHANGE (59.48896544688752 Km/h --> 58.58416893538733 Km/h)

at 4.65 "\$node_(9) setdest 60.0 30.0 58.58416893538733

#Node 9: SPEED CHANGE (58.58416893538733 Km/h --> 35.87835115928491 Km/h)

#Node 9: TARGET REACHED (60.0, 70.0) (35.87835115928491 Km/h)

at 6. #Node 9: NEW TARGET (20.0, 90.0)

at 6.85 "\$node_(9) setdest 60.0 70.0 35.87835115928491" 739987315808788 "\$node_(9)
setdest 60.0 70.0 35.87835115928491" activite: travail 1

activite: Sport direction : UPLEFT"

#Node 9: SPEED CHANGE (35.87835115928491Km/h --> 0.0 Km/h)

at 10.25 "\$node_(9) setdest 60.0 73.299224946584943 0.0

#Node 9: SPEED CHANGE (0.0 Km/h --> 37.204424522461515 Km/h)

at 10.3 "\$node_(9) setdest 60.0 78.299224946584943 37.204424522461515

#Node 9: SPEED CHANGE (37.204424522461515 Km/h --> 0.0 Km/h)

at 12.4 "\$node_(9) setdest 60.0 78.299224946584943 0.0

#Node 9: SPEED CHANGE (0.0 Km/h --> 35.84462832590119 Km/h)

at 12.45 "\$node_(9) setdest 60.0 80.0

#Node 9: SPEED CHANGE (35.84462832590119 Km/h --> 0.0 Km/h)

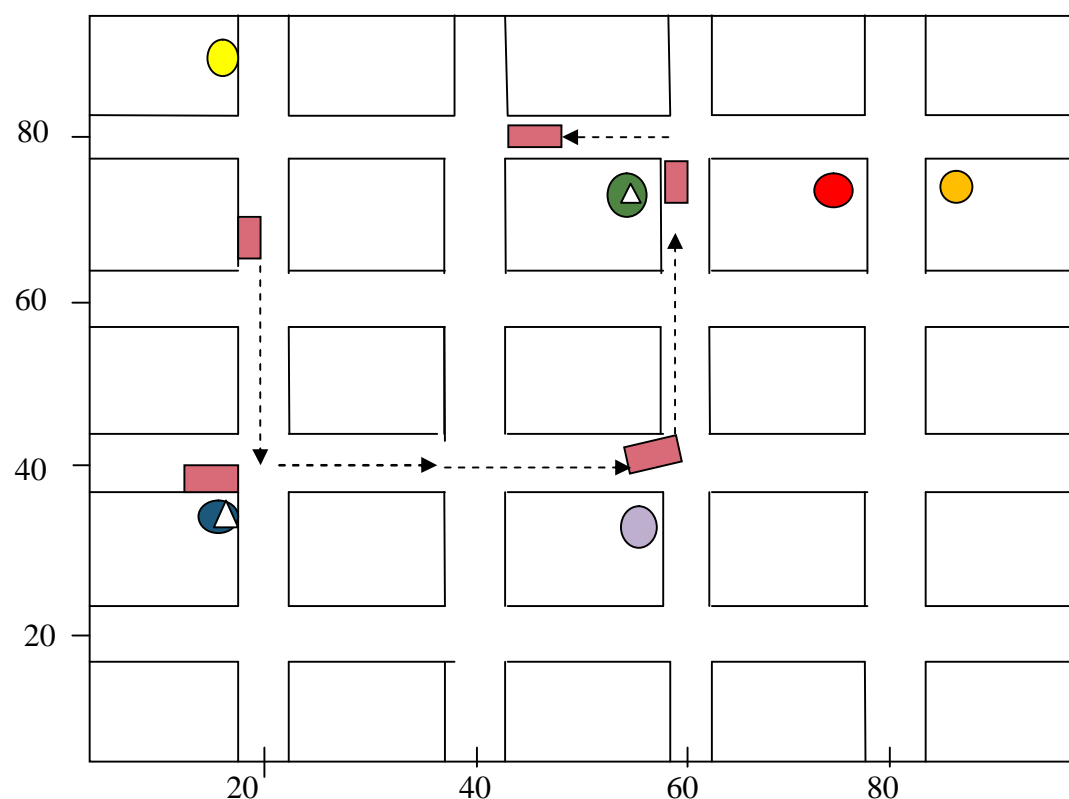
at 15.2 "\$node_(9) setdest 60.0 80.0 0.0

#Node 9: SPEED CHANGE (0.0 Km/h --> 53.8451113305605 Km/h)

at 16.05 "\$node_(9) setdest 55.258344839306986 80.0 53.8451113305605

#Node 9: SPEED CHANGE (53.8451113305605 Km/h --> 0.0 Km/h)

60.0 "\$node_(9) setdest 55.258344839306986 80.0 0.0



- : Localisation de l'  cole 1
-    : Localisation de l'activit   travail 1
- : Localisation de l'activit   sant  
-    : Localisation de l'activit   travail 2
-    : Localisation de l'  cole 2
-    : Localisation de l'activit   sport
-    : Future position atteinte.

Figure IV.15 : repr  sentation de l'itin  raire de l'utilisateur (9)

2.3 Description de l'evenement :

Nous pr  sentons le d  placement de n  ud (9) durant toute la simulation (60 minutes) en passant d'une position    une autre selon l'activit   choisie.

#Node 9 (20.0, 68.38405498806262) --> (20.0, 40.0) 42.40219773865657 Km/h

activite: l'ecole.

\$node_(9) set X_ 20.0

\$node_(9) set Y_ 68.38405498806262.

   chaque fois que le n  ud atteint sa destination il choisit une nouvelle position selon sa prochaine activit  . La position initiale est (20.0, 68.38405498806262), le n  ud essaie d'atteindre la position (20.0, 40.0), en changeant    chaque fois la vitesse et la position.

   2.4098420225013286s le n  ud atteint sa destination avec une vitesse de 42.40219773865657 Km/h. Mais avant d'arriver    cette position le n  ud passe par les feux tricolores. Il s'est arr  t   au feu rouge pour un temps de 4.160963827285221s puis, il a repris son mouvement avec le feu vert. La prochaine position    atteindre est (40.0, 30.0) et ainsi le processus se r  p  te jusqu'   la fin de la simulation.

IV.6 Conclusion :

Dans ce chapitre nous avons mis en   uvre les am  liorations des mod  les de mobilit     tudi  s sous J-SIM, nous avons int  gr   la variation de vitesse des mobiles dans chaque zone, et le temps pass   dans cette zone dans le mod  le bas   sur l'activit  . Le mod  le de mobilit   Downtown est am  lior   de fa  on    ce que la destination sera choisie en fonction des activit  s des n  uds. Ces am  liorations sont r  alis  es afin de rendre ces deux mod  les plus proches de la r  alit  .

Les réseaux ad hoc véhiculaires constituent un nouveau type de réseaux issu des réseaux ad hoc mobiles (MANETs). Leur particularité provient des communications qui peuvent s'instaurer entre véhicules ou bien avec une infrastructure de stations de base. Les réseaux ad-hoc véhiculaires (VANETs) ont attiré progressivement l'attention de la communauté de recherche et d'industrie. Un des défis posés par l'étude de VANETs est la définition d'un modèle de mobilité générique fournissant une description précise et réaliste de mobilité véhiculaire. Pour modéliser la mobilité dans les VANETs plusieurs modèles de mobilités sont proposés, certains prennent en considération les contraintes de l'environnement à simuler (zone rurale, zone urbaine, le type des routes,...etc).

La principale motivation pour la réalisation de ce travail était d'étudier et de simuler les modèles de mobilité réalistes d'un réseau ad-hoc véhiculaire en l'adaptant au simulateur J-Sim qui est dédié aux réseaux de capteurs. Ceci est justifié par le fait qu'un réseau de capteurs peut servir d'ossature de base à un VANET. On peut imaginer un réseau VANET reposant sur des capteurs installés dans les véhicules.

Dans le premier chapitre nous avons présenté les réseaux véhiculaires VANETs et les réseaux ad hoc mobile en général, ce qui nous a permis d'approfondir nos connaissances.

L'étude menée sur les modèles de mobilité des MANETs et particulièrement dans VANET dans le deuxième chapitre nous a été utile pour améliorer les deux modèles de mobilité : le modèle basé sur l'activité et le modèle Downtown. Cette amélioration a fait l'objet d'une simulation présentée dans le chapitre trois et les résultats obtenus, dans le chapitre quatre.

Comme perspectives de notre travail le modèle peut être utilisé dans la simulation des protocoles de routage pour tester leurs performances et dans les applications de localisation dans les réseaux cellulaires pour la réservation des ressources ou les systèmes de prédiction des mouvements.

- [1] N. Baccour, “*Etude comparative de deux simulateurs pour les réseaux sans fil Ad Hoc*”, Mémoire de fin d’études, école Nationale d’Ingénieurs de Sfax, Département d’Informatique et de Mathématiques Appliquées, Juillet 2005.
- [2] Nouha BACCOUR SELLAMI, « *conception d’une nouvelle stratégie de routage dynamique pour les réseaux mobiles ad hoc* », thèse de doctorat, juin 2006.
- .
- [3] A. Delye, V. Gauthier, M. Marot, and M. Becker, “Etat de l’art sur les réseaux de Capteur”, Rapport de Recherche INT N-05001RST GET-INT, UMR5157 SAMOVAR, Institut National des Télécommunications, Evry, France, 2005.
- [4] CHEE-YEE CHONG AND SRIKANTA P. KUMAR, “*Sensor Networks: Evolution, Opportunities, and Challenges*”, *Proceedings of the IEEE*. 2003, Vol. 91, pages 1247-1256.
- [5] I.F. Akyildiz, W. Su*, Y. Sankarasubramaniam, E. Cayirci, “*Wireless sensor networks: a survey*”, Broadband and Wireless Networking Laboratory, School Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332,
- [6] Archana Bharathidasan, Vijay Anand Sai Ponduru, “*Sensor Networks: An Overview*”, Department of Computer Science University of California, pages 20- 23, May 2003.
- [7] Laura Raileanu et Fatemi Nastaran, « Les Réseaux de Senseurs », Haute école d’ingénierie et de gestion du Canton de Vaud, 10/01/2006.
- [8] J. Choi, S. Jung, Y. Kim, and M. Yoo, "A Fast and Efficient Handover Authentication Achieving Conditional Privacy in V2I Networks," in *Proceedings of the 9th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking and Second Conference on Smart Spaces*, St. Petersburg, Russia, pages. 291-300, 2009.
- [9] Q. Xu and D. Jiang, "Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum", Vehicular Technology Conference, 2003, pages. 2451- 2455, Apr 2003.
- [10] Patrick Marlier, Moez Jerbi, and Sidi Mohammed Senouci, " Experimental Assessment of V2V and I2V Communications " International Workshop on Mobile Vehicular Networks (MoVeNet 2007) October 12, 2007, Pisa, Italy.

- [11] Broch J, Maltz D, Johnson D, Hu Y, Jetcheva J, "Multi-hop wireless ad hoc network routing protocols". In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 85–97, 1998.
- [12] Cholatip YAWUT, "Adaptation à la mobilité dans les réseaux ad hoc", thèse de doctorat, 28/09/2009.
- [13] E. M. Royter, P. M. Melliar-Smith and L. E. Moser, "An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks," *Proceedings of the IEEE International Conference on Communications*, pages 857-861, June 2001.
- [14] Hong X, Gerla M, Pei G, Chiang C. A group mobility model for ad hoc wireless networks. In *Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM)*, August 1999.
- [15] T. Camp, J. Boleng, and V. Davies, A Survey of Mobility Models for Ad Hoc Network Research, in *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, , pages 483-502, 2002.
- .
- [16] F. Bai, N. Sadagopan and A. Helmy, "IMPORTANT: A Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Ad Hoc Networks," *Proceedings of the IEEE International Conference on Computer Communications*, pages 825-835, March-April 2003.
- [17] V. Davies, "Evaluating mobility models within an ad hoc network", thèse de master, Colorado School of Mines, 2000.
- [18] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *Proc. of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, Cologne, Germany, Sep. 2005.
- [19] Atulya Mahajan, Niranjan Potnis, Kartik Gopalan, and An-I A. Wang," Mobility Models For Vehicular Ad-Hoc Network Simulations", *Dept. of Computer Science, Florida State University*, 2006.
- [20] Arentze, T.A. and H.J.P. Timmermans, ALBATROSS: A Learning Based Transportation Oriented Simulation System. Eindhoven, the Netherlands, European Institute of Retailing and Service Studies, 2000.
- [21] <http://www.doc-etudiant.fr/Informatique/Reseaux-informatiques/Cours-Generalites-sur-les-reseaux-sans-fil-116584.html>.

- [22] Donggen Wong, “Conjoint approaches to developing activity based models”, thèse de doctorat, 1998.
- [23] Zhongwei Sun “Travel Information Impact on Activity travel Patterns”, Eindhoven University of Technology Date of Defence, 19 October 2009.
- [24] Simon H. A, “Invariant of human behavior. “*Annual Review of Psychology*, 41, pages 1-19, 1996.
- [25] J. Scourias and T. Kunz. An activity-based mobility model and location management simulation framework. In *2nd ACM Int. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, page 61–68, 1999.
- [26] M. DAOUI, “Réservation de Ressources et Prédiction de Cellules dans un Réseau Mobile de Troisième Génération. “, thèse de doctorat, 2009.
- [27] Tranplan Associates. Waterloo region travel survey 1987: An overview of the survey findings. Technical report, Regional Municipality of Waterloo, Department of Planning and Development, October 1989.
- [28] <http://sites.google.com/site/jsimofficial/>
- [29] H.-Y. Tyan and J. C. Hou. JavaSim: A component-based compositional network simulation environment. In *Proc. of Western*.
- [30] j -Sim. <http://www.j-sim.org/>
- [31] Tcl Developer Site, 2011, <http://www.tcl.tk/>.
- [32] kamal BEYDOUN, “conception d’un protocole de routage hiérarchique pour les réseaux de capteurs “, thèse de doctorat, 2009.
- [33] <http://www.eclipse.org/>.
- [34] <http://www.netbeans.org/>.
- [35] A. Mahajan, N. Potnis, K. Gopalan, and A.-I. A. Wang, “Urban mobility models for vanets”, in *Proc. of the IEEE Workshop on Next Generation Wireless Networks (WoNGeN)*, Dec. 2006.
- [36] D. B. J. Amit Kumar Saha, “Modeling mobility for vehicular ad-hoc networks”, in *VANET 2004*, Oct, <http://www.cs.rice.edu/~amsaha/Resume/vanet2004/vanet2004.pdf>

- [37] Xavier Campos Ríos , Diego Pastor Pérez, “ Performance evaluation of manhattan downtown scenario for vehicular ad hoc networks with CityMob and NCTUns”, juin 2011.

Annexe

Installation du simulateur JSIM sous la plateforme WINDOWS

1. Préparation d'environnement :

Etape1 :

Pour installer le simulateur JSIM, il faut déjà avoir installé un logiciel spécial depuis le site web de la société Sun Microsystems, son nom complet est *Java 2 Software Development Kit (J2SDK)*. Pour des raisons de stabilité, il est hautement recommandé d'installer la version *J2SDK 1.4* ou bien des versions ultérieures. Ce logiciel peut être téléchargé depuis le site suivant :

<http://java.sun.com/j2se>

Etape2 :

La compilation des codes sources pour le simulateur JSIM peut se faire soit via le compilateur *Apache Ant*, soit par la commande « *make* ».

Le premier compilateur est valable depuis le lien : <http://ant.apache.org/>

Pour compiler via la commande « *make* », on doit installer l'utilitaire *GNU-Make makefiles* in Java. Cet utilitaire permet d'installer des commandes propres aux plateformes LINUX sous WINDOWS.

Etape3 :

La version du simulateur J-SIM la plus utilisée est J-SIM version1.3. Elle peut être téléchargée depuis le lien : http://www.j-sim.org/cgi-bin/j-sim_downloader.cgi

Il faut noter que la version J-Sim version1.3 n'est pas compatible avec la version J2SDK 5.0.

Pour des questions d'interopérabilités, on doit installer le patch4 ou bien des versions de patch ultérieures. Le patch que nous avons installé est *patch4-version1.3* valable sur :

<http://www.jsim.org/patch.html>

2. Installation de JSIM :

Etape4 :

L'installation du JSIM se fait sur plusieurs étapes:

On décompresse sous la racine C :\\ le package téléchargé J-Sim_v1.3.tar. On obtient par la suite un dossier nommé jsim-1.3.

Remarque : la décompression des fichiers .tar sous WINDOWS se fait avec le logiciel WINRAR.

On modifie le fichier setcpath.bat placé sous C:\\jsim-1.3 comme suit :

- On affecte à la variable %J_SIM% le chemin d'installation du simulateur JSIM
- On affecte à la variable % JAVA_HOME % le chemin d'installation de J2SDK
- On initialise une variable % ANT_HOME % au chemin d'installation du compilateur

Apache Ant et on ajoute à la variable %PATH% le chemin vers les commandes de compilation si on le choisit comme compilateur pour JSIM.

Si on a choisit de compiler les codes sources du simulateur JSIM avec la commande « *make* », on doit donc ajouter le chemin d'installation de à la variable %PATH%.

On ajoute à la variable %CLASSPATH% le chemin vers les classes du simulateur JSIM.

Si le compilateur choisi pour le simulateur JSIM, est *Apache Ant*, le fichier *setcpath.bat* doit comprendre les données suivantes :

```
@echo off
set JAVA_HOME=C:\\jdk
set J_SIM=C:\\jsim-1.3
set ANT_HOME=C:\\ant
set PATH=%PATH%;c:\\jdk\\bin;c:\\ant\\bin
set CLASSPATH=.;c:\\jsim-1.3\\classes;c:\\jsim-1.3\\jars\\tcl.zip;c:\\jsim-1.3\\jars\\jython.jar
```

Si la compilation pour le simulateur JSIM est suivant la commande « *make* », le fichier *setcpath.bat* doit être de la forme suivante :

```
@echo off
set JAVA_HOME=C:\\jdk
set J_SIM=C:\\jsim-1.3
set PATH=%PATH%;c:\\jdk\\bin;c:\\UnxUtils_1\\usr\\local\\wbin
set CLASSPATH=.;c:\\jsim-1.3\\classes;c:\\jsim-1.3\\jars\\tcl.zip;c:\\jsim-1.3\\jars\\jython.jar
```

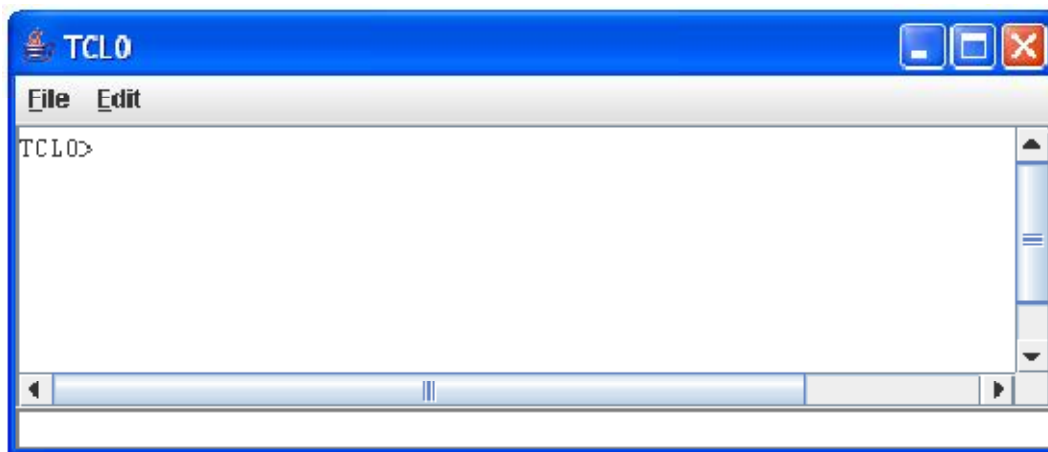
1. On ouvre maintenant la fenêtre des invites des commandes DOS, et on se place sous le répertoire jsim-1.3. pour cela, on tape les commandes suivantes :
2. On exécute maintenant le fichier setcpath.bat en l'appelant sous la fenêtre d'invite des commandes.

```
cd C:\\  
cd jsim-1.3
```

3. On compile tous les codes sources du simulateur JSIM en tapant la commande « *make* » avec l'utilitaire *GNU-Make makefiles in Java* ou la commande « *ant compile* » avec le compilateur *Apache Ant*.
4. On décompresse maintenant le dossier *patch1.3-4.tar.gz*, déjà téléchargé, sous le répertoire jsim-1.3 et on accepte les mises à jour des fichiers existants.
5. On recompile de nouveau les codes sources et on ignore tous les avertissements.

Etape5 :

Pour vérifier si l'installation de JSIM a réussi, vous devez avoir une fenêtre pop up intitulé TCL0 qui apparait en tapant la commande suivante : `java drcl.ruv.System`



Etape 6 : à chaque simulation un script TCL, on doit changer le chemin courant vers où est placé le script, puis taper la commande suivante :

```
java drcl.ruv.System « nom de votre script.tcl »
```