

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'électronique

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Spécialité : Télécommunication et réseaux

Filière : Génie Electrique

Présenté par :

**DADOUN OUIZA
GUENDOZ OUIZA**

Thème

Le routage dans les réseaux véhiculaires VANETs

Mémoire soutenu le 04/07/2017 devant le jury composé de :

Mr. ACHOUR Hakim	Maitre de conférence classe A	Président
Mr. LAHDIR Mourad	Maitre de conférence classe A	Encadreur
Mr. ALOUACHE Djamel	Maitre de conférence classe B	Examineur

Année Universitaire 2016-2017

VANET (Vehicular Adhoc Networks) est une nouvelle technologie émergente qui intègre les fonctionnalités de la nouvelle génération de réseaux sans fil pour les véhicules.

Le principal objectif de ce type de réseau est de faire circuler l'information de la manière la plus efficace possible entre les véhicules ; entre les infrastructures et les véhicules afin d'améliorer la sécurité routière et d'apporter le confort aux conducteurs et aux passagers.

Les réseaux véhiculaires reposent sur des protocoles qui assurent l'échange d'informations entre les véhicules qui sont caractérisés par une forte mobilité ; l'amélioration de la communication revient à la détermination de l'efficacité de ces protocoles.

Notre but dans ce mémoire était d'analyser le protocole de routage AODV (Ad-Hoc On-demand Distance Vector) en terme de débit de transmission, le taux de pertes de paquets et le délai bout en bout.

Nous avons simulé ce protocole sous le simulateur réseaux NS2 dans deux milieux différents urbain et autoroute, et ce, pour déterminer l'environnement auquel il est adapté.

Mots clés : VANET, Protocole de Routage, NS2.

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur Mr : LAHDIR.M, pour sa disponibilité son précieux conseil et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.



Liste des figures

Chapitre I :

<i>Figure I.1 : Mode avec infrastructure</i>	4
<i>Figure I.2 : Le mode sans infrastructure (IBSS)</i>	5
<i>Figure I.3 : Un exemple de réseau Ad hoc</i>	6
<i>Figure I.4 : Architecture générale d'un réseau de capteur</i>	7
<i>Figure I.5 : Un exemple d'un réseau VANET</i>	9
<i>Figure I.6 : Les éléments constituant le véhicule intelligent</i>	10
<i>Figure I.7 : Communication Basé sur les RSU</i>	11
<i>Figure I.8 : Communication véhicule à véhicule</i>	13
<i>Figure I.9 : Communication véhicule à station de base</i>	13
<i>Figure I.10 : Communication hybride</i>	14
<i>Figure I.11 : Les différents types de mode de transmission dans les VANETs</i>	16
<i>Figure I.12 : Ensemble de technologies de communication pour VANETs</i>	19

Chapitre II :

<i>Figure II.1 : Classification des protocoles de routage dans les VANETs</i>	26
<i>Figure II.2 : Procédure de découverte de route dans AODV</i>	29
<i>Figure II.3 : Sélection des MPRs dans OLSR</i>	31
<i>Figure II.4 : Un exemple de zone dans le protocole ZRP</i>	33
<i>Figure II.5 : Exemple de Greedy Forwarding</i>	35
<i>Figure II.6 : Perimeter Forwarding ou règle de la main droite</i>	36
<i>Figure II.7 : La sélection des intersections dans GyTAR</i>	38
<i>Figure II.8 : Echanges de données entre deux intersections</i>	39
<i>Figure II.9 : Réseaux véhiculaire basés sur les groupes</i>	40
<i>Figure II.10 : Sélection de relais dans IVG</i>	42

Chapitre III :

<i>Figure III.1 : Format d'un Message RREQ</i>	46
<i>Figure III.2 : Format d'un message RREP</i>	46
<i>Figure III.3 : Architecture basic de NS2</i>	52
<i>Figure III.4 : Flot de simulation avec NS2.....</i>	53
<i>Figure III.5 : Visualisation de la topologie et le déroulement de trafic avec Nam</i>	5
<i>Figure III.6 : Exemple de deux lignes de fichier trace du protocole AODV</i>	56
<i>Figure III.7 : Exemple d'un graph sous NS2.....</i>	57

Figure III.8 : Modèle de propagation Two-ray ground 60

Chapitre IV :

Figure IV.1 : Disposition des nœuds 69

Figure IV.2 : L'évolution de débit de transmission de paquet des nœuds au cours du temps 70

Figure IV.3 : La position des nœuds a 16.17s de la simulation 71

Figure IV.4 : L'évolution des pertes des paquets de transmission au cours du temps 72

Figure IV.5 : L'évolution de délai de bout en bout au cours du temps 73

Figure IV.6 : Disposition initiale des nœuds 75

Figure IV.7 : L'évolution de débit de transmission de paquet des nœuds au cours du temps 76

Figure IV.8 : L'évolution des pertes des paquets de transmission au cours du temps 77

Figure IV.9 : L'évolution de délai de bout en bout au cours du temps 77

Figure IV.10 : Communication à multi saut entre les nœuds 5 et 16..... 78

Liste des tableaux

Tableau III.1 : Le format du fichier trace ancien format	55
Tableau III.2 : Liste des composants disponibles dans NS	59
Tableau IV.1 : Paramètres de simulation dans le scenario 1	68
Tableau IV.2 : Paramètres de simulation dans le scenario 2.....	74

SOMMAIRE

Introduction	1
Chapitre I : Vue d'ensemble sur les réseaux VANET	
1. Préambule	3
I.2. les réseaux sans fils	3
<i>I.2.1. Réseaux avec infrastructure (cellulaire)</i>	3
<i>I.2.2. Réseaux sans infrastructure (Ad Hoc)</i>	4
I.3. les réseaux Ad Hoc	5
<i>I.3.1. Définition d'un réseau Ad Hoc</i>	5
<i>I.3.2. Les réseaux de capteurs sans fils RCSF</i>	6
<i>I.3.3. Caractéristiques des réseaux Ad Hoc</i>	7
I.4. les Réseaux véhiculaires Ad Hoc	8
<i>I.4.1. Définition d'un réseau VANET</i>	8
<i>I.4.2. Historique</i>	9
<i>I.4.3. le nœud de réseau VANET</i>	10
<i>I.4.4. Systèmes de transport intelligents (ITSs)</i>	11
<i>I.4.5. Station de bord de la route (RSU)</i>	11
<i>I.4.6. OBU (On-Board Unit)</i>	12
I.5. Architecture de communication des VANETs	12
<i>I.5.1. La communication inter-véhicule (V2V)</i>	12
<i>I.5.2. La communication entre véhicules et les stations de bord de la route (V2I)</i>	13
<i>I.5.3. La communication hybride</i>	14
I.6. Caractéristiques des réseaux VANETs	14
I.7. Modes de transmission dans les réseaux VANETs	15
<i>I.7.1. Le mode Unicast (Point-à-point)</i>	16
<i>I.7.2. Le mode Broadcast (diffusion)</i>	16
<i>I.7.3. Le mode Multicast (multipoint)</i>	16
<i>I.7.4. Le mode Géocast</i>	16
I.8. Applications des VANETs	17
<i>I.8.1. Application pour la prévention et la sécurité routière</i>	17
<i>I.8.2. Application pour la gestion du trafic routier</i>	17
<i>I.8.3. Application pour le confort</i>	17
I.9. Technologies de communication et de diffusion pour les VANETs	17
I.10. Normes et standardisation dans les réseaux VANETs	19
<i>I.10.1. WAVE (Wireless Access in Vehicular Environment) et IEEE 802.11p</i>	19
<i>I.10.2. DSRC (Dedicated Short Range Communications)</i>	20
<i>I.10.3. ISO : TC204 /WG16-CALM</i>	21
<i>I.10.4. Le consortium Car-to-Car (C2C-CC)</i>	21
I.11. Projets de recherche dans le cadre des réseaux véhiculaires	21

12. Discussion	23
-----------------------------	-----------

Chapitre II : Le routage dans les réseaux VANETs :

II.1. Préambule	24
II.2. Concepts de base du routage	24
<i>II.2.1. Le routage</i>	24
<i>II.2.2. Définition d'un protocole de routage</i>	24
<i>II.2.3. Objectif du routage</i>	24
II.3. Problématique de routage dans les réseaux ad-Hoc Véhiculaire	24
II.4. Les protocoles de routage dans les réseaux VANETs	25
II.4.1. Les protocoles basés sur la topologie	27
<i>II.4.1.1. Les protocoles réactifs (On-Demand Driven)</i>	27
<i>II.4.1.1.1. Le protocole DSR (Dynamic Source Routing)</i>	27
<i>II.4.1.1.2. Le protocole AODV (Ad-hoc On Demand Distance Vector)</i>	28
<i>II.4.1.2. Les protocoles proactifs (Table-Driven)</i>	29
<i>II.4.1.2.1. Le protocole GSR (Global State Routing)</i>	30
<i>II.4.1.2.2. Le protocole OLSR (Optimized Link State Protocol)</i>	30
<i>II.4.1.2.3. Le protocole DSDV (Destination-Sequenced Distance-Vector)</i>	31
<i>II.4.1.3. Protocoles hybrides</i>	32
<i>II.4.1.3.1. Le protocole Le protocole ZRP (Zone Routing Protocol)</i>	32
II.4.2. Les protocoles de routage avec localisation géographique	33
<i>II.4.2.1. Les protocoles géographiques tolérant aux délais (DTN)</i>	34
<i>II.4.2.1.1. Le protocole VADD (Vehicle-Assisted Data Delivery)</i>	34
<i>II.4.2.2. Les protocoles géographiques non tolérant aux délais (Non-DTN)</i>	34
<i>II.4.2.2.1. Le protocole GPSR (Greedy Perimeter Stateless Routing)</i>	35
<i>II.4.2.2.2. Le protocole GyTAR (Greedy Traffic Aware Routing protocol)</i>	36
<i>II.4.2.3. Les protocoles géographiques Hybride</i>	39
<i>II.4.2.3.1. Le protocole GéoDTN+NAV (Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments)</i>	39
II.4.3. Les protocoles de routage basés sur les clusters (groupes)	40
<i>II.4.3.1. Le protocole CBLR (Cluster Based Location Routing)</i>	40
<i>II.4.3.2. Le protocole CBDRP (Cluster-Based Directional Routing Protocol)</i>	41
II.4.4. Les protocoles Géo-caste	41
<i>II.4.4.1. Le protocole IVG (Inter-Vehicle Geocast)</i>	41

II.4.4.2. Le protocole DRG (Distributed Robust Geocast)	42
II.4.5. Les protocoles Broad-caste	42
II.4.5.1. Le protocole UMB (Urban Multihop Broadcast Protocol)	43
II.4.5.2. Le protocole DVCAST (Distributed vehicular broadcast protocol)	43
II.5. Discussion	43
Chapitre III : Application	
III.1. Préambule	44
III.2. Etude détaillé du protocole de routage AODV	44
III.2.1. Le mécanisme de fonctionnement du protocole.....	44
III.2.2. Génération et acheminement des requêtes RREQ.....	45
III.2.3. Génération et acheminement des réponses RREP.....	46
III.2.4. Génération et acheminement des erreurs RERR	47
III.2.5. Maintenance des routes	47
III.2.6 Gestion des numéros de séquence	47
III.2.7. Avantages d'AODV	48
III.2.8. Inconvénients d'AODV.....	48
III.2.9. Métriques de performances	48
III.3. Les outils de simulations.....	49
III.4. Le choix de simulateur	51
III.4.1. Le simulateur du réseau NS2.....	51
III.4.1.1. Présentation	51
III.4.1.2. L'outil de visualisation (NAM)	53
III.4.1.3. Le Fichier trace (résultat de la simulation : out.tr)	54
III.4.1.4. Les outils d'analyse de fichier trace	57
III.5. Les composants d'un réseau sous NS2.....	58
III.6. Les différents modèles de propagation radio sous NS2.....	59
III.6.1. Le modèle de propagation en espace libre (Free space model).....	60
III.6.2. Two-ray ground reflection model (modèle à deux rayons, directs et par le sol) ..	60
III.6.3. Le modèle Shadowing.....	60
III.7. Modèles de mobilité sous NS2.....	61
III.7.1. Le Random Waypoint (RWP).....	61
III.7.2. Le Trajectory Based Mobility Model	61
III.8. Le processus de simulation dans NS2	61

III.9. Création d'un scénario dans NS 62
III.10. Discussion..... 66

Chapitre IV : Simulations et interprétations des résultats

IV.1. Préambule 67
IV.2. Scenarios et paramètres de simulation..... 67
 IV.2.1 Scénario 1 67
 IV.2.1.1. Résultats et interprétations de scénario 1 70
 IV.2.2. Scénario 2 73
 IV.2.2.1. Résultats et interprétations de scénario 2 74
IV.3 : Discussion 78
Conclusion 79
Annexe
Liste des abréviations
Bibliographie.....

Liste des abréviations

STI: Système de transport intelligent.
VANET: Vehicular Ad-Hoc Network
BSS: Basic Service Set
ESS: Extended Service Set
IBSS: Independent Basic Service Set
MANET : Mobile Ad hoc Net Work
RCSF : réseau de capteurs sans fil
WSN: Wireless Sensor Network
WiMax: World wide Interoperability for Microwave Access
GPS: Global Positioning System
IEEE: Institute of Electrical and Electronics Engineers
WAVE: Wireless Access in Vehicular Environment
DSRC: Dedicated Short Range Communications
V2V: Vehicle to Vehicle
V2I: vihecule to Infrastructure
RSU: Road Side Units
OBU: On-Board Unit
Wi-Fi: Wireless Fidelity
EDGE: Enhanced Data for GSM Evolution
UMTS: Universal Mobile Telecommunication System
LTE: Long Term Evolution
GMR: GEOMobile Radio Interface
DVB: Digital Video Broadcasting
DVBSH: DVB-Satellite services to Handhelds
DSR: Dynamic Source Routing
AODV: Ad-Hoc On- demand Distance Vector
RREQ: Route Request message
RREP: (Route Reply message
RERR: Route Error message
GSR: Global State Routing
OLSR: Optimized Link State Protocol

MPR: Multi-Point Relaying
DSDV: Destination-Sequenced Distance-Vector
ZRP: Zone Routing Protocol
DTN: Delay Tolerant Network
VADD: Vehicle-Assisted Data Delivery
GPSR: Greedy Perimeter Stateless Routing
GyTAR: Greedy Traffic Aware Routing protocol
GLS: Grid Location Service
CBLR: Cluster Based Location Routing
CBDRP: Cluster-Based Directional Routing Protocol
IVG: Inter-Vehicle Geocast
DRG: Distributed Robust Geocast
UMB: Urban Multihop Broadcast Protocol
DVCAST: Distributed vehicular broadcast protocol
NS2: Simulateur Réseau
SUMO: Simulator of Urban Mobility
OMNET++: Objectif Modular Network Test-Bed
Veine: Vehicles in Network Simulator
NCTUns: National Chiao Tung University Network Simulator
OTCL: Object Tools Command Language
NAM: Network Animator
UDP: User Datagram Protocol
TCP: Transmission Control Protocol
CBR: Constant Bit Rate
FIFO: First In First Out

Les réseaux sans fil ont connu ces dernières années un essor spectaculaire, ils s'imposent aujourd'hui de façon indéniable. Leur apparition d'un côté et l'avancement des technologies de la communication et de l'information de l'autre, a donné lieu à la naissance de ce qu'on appelle les Systèmes de Transport Intelligents qui ont comme objectif de rendre plus efficace le système routier.

Les réseaux véhiculaires (VANETs) sont considérés parmi les technologies récentes de communication sans fil, de nos jours les véhicules ne sont plus que de simples moyens de transport, ils sont dotés d'une source d'intelligence par leur interaction avec l'environnement routier grâce à des équipements informatiques embarqués dans les véhicules et intégrés dans des stations le long des routes et autoroutes, les véhicules peuvent communiquer les uns avec les autres et avec des équipements fixes ou mobiles. En exploitant leur récentes capacités de communication, la création d'un réseau dédié permet de rendre plus agréable le temps qu'on passe à bord, tout en améliorant la sécurité routière.

La technologie de communication WiMax, les réseaux cellulaires et les réseaux satellitaires sont proposés pour assurer les communications à longue portée. Une nouvelle technologie de communication conçue spécifiquement aux VANETs s'est émergée. Cette technologie, nommée Dedicated Short Range Communications (DSRC), supporte les communications à moyenne et à courte portée qui peuvent être de véhicules-à-infrastructure ou en mode ad hoc de véhicule-à-véhicule.

En effet, en conjuguant les données issues de la cartographie GPS à celles détectées par les capteurs du véhicule, il devient possible de faire de cette interactivité une conduite coopérative.

Dans ce concept, les réseaux ad hoc véhiculaires permettent le partage, de manière collaborative, de différents types de données entre les véhicules. Le principal objectif de ce type de réseau est d'améliorer la sécurité routière afin de diminuer les accidents, d'apporter aux conducteurs et aux passagers quelques applications du confort et de divertissement.

Depuis l'apparition des VANETs, de nombreux standards, applications et mécanismes de routage ont été proposés pour répondre aux spécificités de cette nouvelle classe de réseaux.

En effet, le routage est un élément essentiel dans un réseau VANET, il constitue un grand défi à relever pour sa conception en raison de la forte mobilité des véhicules, de la

diversité spatio-temporelle de la densité du trafic et de la propagation des ondes radio en environnement extérieur défavorable à l'établissement des communications sans fil. Différents protocoles de routage ont été conçus, font l'objet d'un grand effort de tests, d'évaluations et d'améliorations.

Dans le cadre de ce mémoire, l'évaluation de certains aspects de performances de protocoles de routage réactif AODV sera abordée par des simulations sous NS2. Elle permet de dégager l'évolution de quelques métriques comme le débit de transmission, les pertes des paquets et le délai de transmission de bout en bout dans deux environnements de VANETs urbain et autoroute.

Ce mémoire est structuré en quatre chapitres :

Dans le premier chapitre, nous présentons une vue d'ensemble sur les réseaux véhiculaires Ad Hoc VANET, leurs caractéristiques et leurs applications.

Dans le deuxième chapitre, nous donnons un état de l'art des protocoles qui ont été proposés pour assurer le routage dans les réseaux VANET. Nous décrivons leurs fonctionnements et les stratégies utilisées pour l'acheminement des données entre les différents nœuds mobiles.

Une étude détaillée du protocole AODV est donnée dans le troisième chapitre et nous présentons l'outil de simulation Network Simulator NS2, ainsi que les langages et outils graphiques utilisés pour l'interprétation et l'analyse des résultats de simulation.

Le dernier chapitre est consacré à l'interprétation et discussion des résultats obtenus de la simulation.

Nous terminons ce mémoire par une conclusion pour résumer les apports essentiels de notre travail et perspective de recherche.

I.1. Préambule :

Les technologies sans fil ont connu une progression rapide ces dernières années, ce qui a donné naissance à de nouveaux systèmes de communication ; l'idée première a été de rendre les véhicules et les routes plus intelligents, une application du concept d'intelligence consiste à munir les véhicules et les routes de capacités permettant de rendre la conduite plus sûre et de rendre le temps passé sur les routes plus convivial. Parmi ces systèmes de communication ce que nous appelons aujourd'hui les systèmes de transport intelligents ITS (Intelligent Transport System). [1]

L'architecture sur laquelle se base les systèmes de transport intelligents est connue sous le nom de VANET (Vehicular Ad-Hoc Network) ou réseaux Ad-Hoc Véhiculaire ; une technologie émergente intégrant les dernières techniques de communication.

I.2. Les réseaux sans fil :

Un réseau sans fil (en anglais Wireless network) est un ensemble d'appareils connectés entre eux et qui peuvent s'envoyer et recevoir des données sans qu'aucune connexion « filaire » entre eux ne soit nécessaire ; mais qui établit la communication par des ondes radioélectriques (liaison radio ou liaison infrarouge) à la place des câbles habituels (coaxial, paire-torsadée ou fibre optique).

Grâce aux réseaux sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu.

Les réseaux sans fil, peuvent être classés en deux classes : les réseaux avec infrastructure et les réseaux sans infrastructure.

I.2.1. Réseaux avec infrastructure (cellulaire) :

En mode infrastructure, un nœud mobile communique uniquement avec un nœud fixe, appelé point d'accès (AP Access Point) via une liaison sans fil. L'ensemble formé par le point d'accès et les unités mobiles est appelé ensemble de services de base (en anglais *Basic Service Set*, noté BSS) [2] et constitue une cellule. La figure I.1 illustre un exemple d'un réseau mobile avec infrastructure.

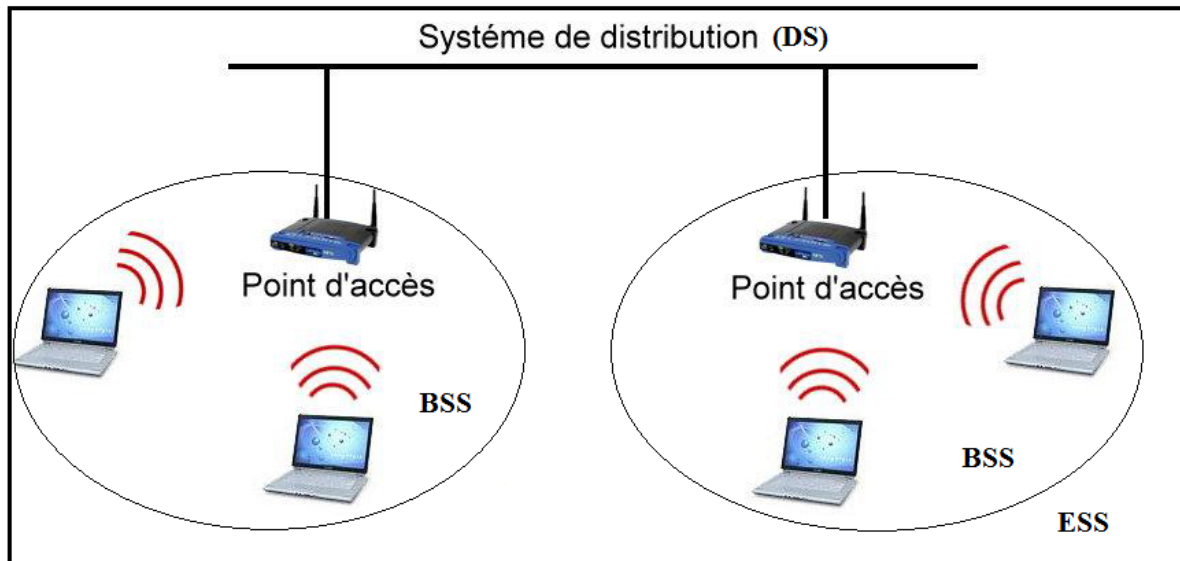


Figure I.1 : Mode avec infrastructure

Les points d'accès se relient entre eux par des liaisons appelées Système de Distribution (notée DS pour Distribution System) afin de constituer un ensemble de services étendus (*Extended Service Set* ou ESS).

I.2.2. Réseaux sans infrastructure (ad hoc) :

Le réseau mobile sans infrastructure également appelé réseau Ad hoc ou IBSS (Independent Basic Service Set) [2] ne comporte pas la notion de point d'accès, capable de s'organiser sans infrastructure définie préalablement.

Toutes les stations du réseau se connectent les unes aux autres afin de construire un réseau point à point (P2P pour peer to peer). Ainsi, chaque machine joue en même temps le rôle de client et le rôle de point d'accès ; ce type de réseau s'appelle le réseau Ad-Hoc.

La figure I.2 illustre le schéma d'un réseau sans infrastructure.

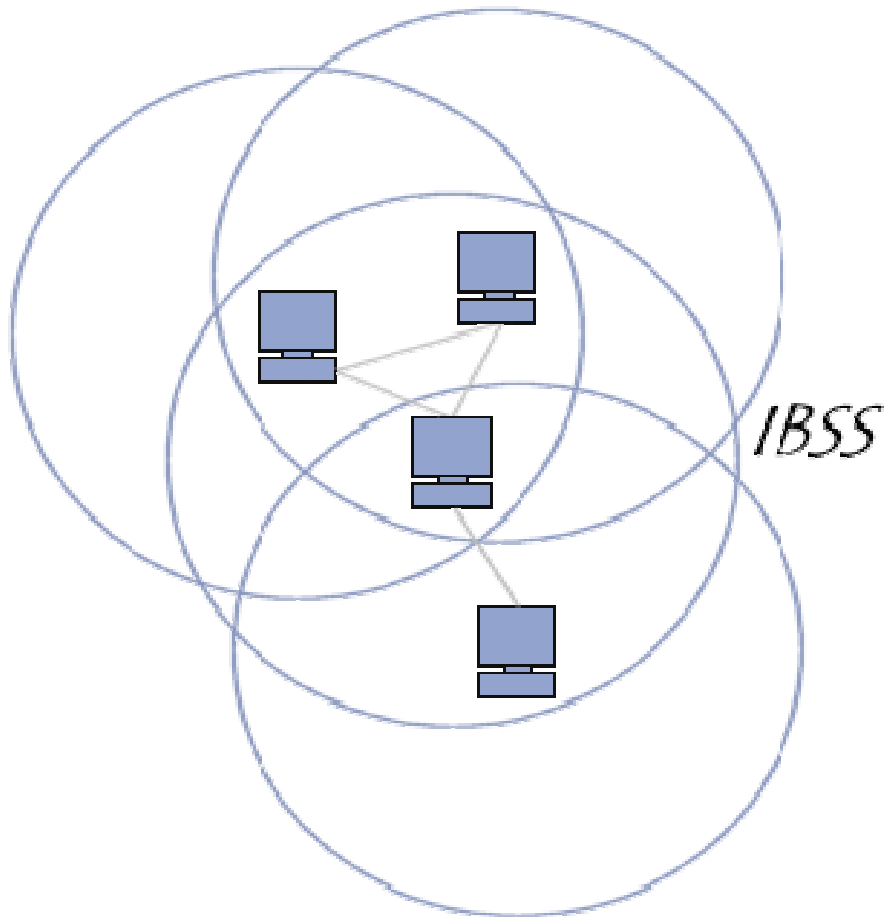


Figure I.2 : Le mode sans infrastructure (IBSS)

I.3. Les réseaux Ad Hoc :

I.3.1. Définition d'un réseau Ad Hoc :

Le réseau mobile ad hoc, appelé également MANET (Mobile Ad hoc Network) est un réseau dont la topologie ne bénéficie d'aucune infrastructure préexistante ou administration centralisée.

Il permet donc à deux nœuds voisins qui sont chacun à portée des ondes l'un de l'autre (condition appropriée de propagation d'ondes radio) de rentrer en communication directement. La figure I.3 illustre un exemple de réseaux Ad-Hoc.

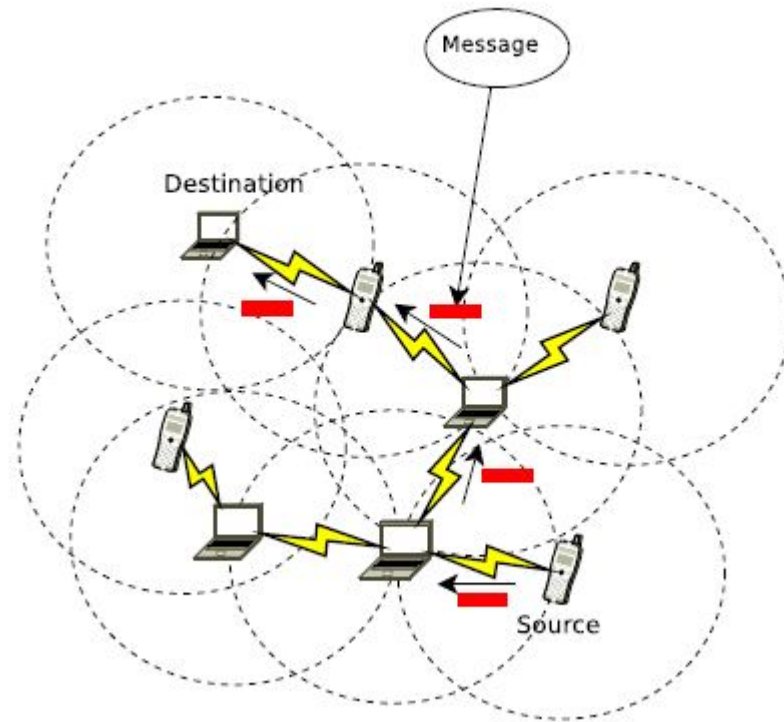


Figure I.3 : Un exemple de réseau Ad hoc.

Dans un réseau ad hoc, deux nœuds d'un réseau s'échangent des données sous forme de paquets à l'aide de nœuds appelés nœuds intermédiaire. Pour cela, le nœud initiateur de l'échange, appelé nœud source, transmet les paquets à ses voisins. Si le nœud n'est pas la destination du paquet, un nœud intermédiaire transmet un paquet à ses voisins, et les voisins transmettent à leurs voisins, jusqu'à ce que le paquet ait atteint la destination. [3]

I.3.2. Les réseaux de capteurs sans fil RCSF :

Un réseau de capteurs sans fil (RCSF ou WSN : Wireless Sensor Network) est un type particulier des réseaux ad hoc ; il est constitué d'un ensemble de nœuds capteurs déployés dans une zone d'intérêt. Ils sont capables de réaliser les fonctions de capture, de traitement et ne sont pas intégrés à une quelconque architecture de communication préexistante, mais ils communiquent à l'aide d'un réseau Ad Hoc sans fil.

L'alimentation électrique de chaque capteur est assurée par une batterie individuelle dont la consommation pour la communication et le calcul lié au traitement de l'information doit être optimisée. la figure I.4 illustre une architecture générale d'un réseau de capteurs.

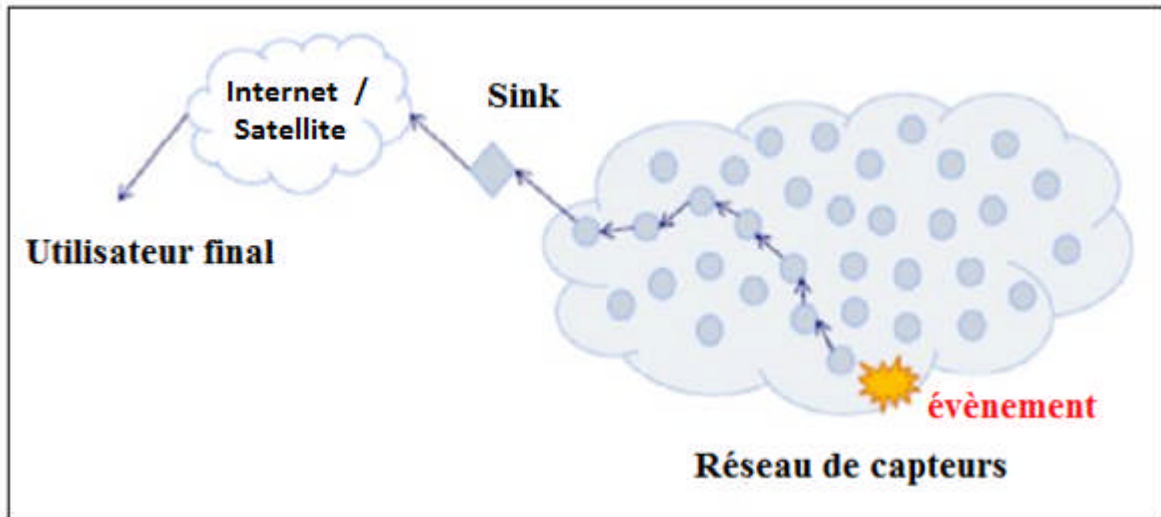


Figure I.4 : Architecture générale d'un réseau de capteur.

I.3.3. Caractéristiques des réseaux Ad Hoc : Les réseaux mobiles Ad Hoc présentent plusieurs caractéristiques :

➤ **Topologie dynamique (Mobilité) :**

Une particularité très importante qui distingue les réseaux mobiles Ad Hoc des réseaux filaires est la mobilité de ses nœuds. Les nœuds se déplacent indépendamment les uns des autres et de manière imprévisible. Un nouveau terminal peut intégrer le réseau et en ressortir à n'importe quel moment. [4]

➤ **Absence d'infrastructure :**

Les nœuds d'un réseau Ad Hoc travaillent en mode peer to peer totalement distribué, ce qui leur permet de se déplacer librement. Ces nœuds agissent en tant que routeurs pour relayer des communications ou générer leurs propres données.

➤ **Vulnérabilité (Une sécurité physique limitée) :**

Les réseaux sans fil sont de nature plus sensible aux problèmes de sécurité. Le principal problème pour les réseaux ad hoc ne se situe pas au niveau du support physique mais principalement dans le fait que tous les nœuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau.

➤ **Liaisons sans fil (Une bande passante limitée) :**

Les technologies de communication sans fil sont indispensables à la mise en place d'un réseau ad hoc. Malgré des progrès très importants, leurs performances restent et resteront en dessous de celles des technologies des réseaux filaires.

La bande passante varie d'un type de réseau ad hoc à un autre. Par exemple, les performances attendues dans les réseaux de capteurs ne seront pas les mêmes que celles attendues dans les réseaux ad hoc d'entreprise. [5]

➤ **Autonomie des nœuds (Des contraintes d'énergie) :**

La consommation d'énergie constitue un problème pour des équipements fonctionnant grâce à une alimentation électrique autonome. Ces équipements intègrent des modes de gestion d'énergie et il est important que les protocoles mis en place dans les réseaux ad hoc prennent en compte ce problème.

I.4. Les Réseaux Véhiculaires Ad hoc :

I.4.1. Définition d'un réseau VANET:

Les réseaux sans fil véhiculaires aussi appelés VANETs (Vehicular Ad-hoc Network) constituent une forme particulière des réseaux mobiles Ad-Hoc (MANET), où les nœuds mobiles sont des véhicules. [6]

Ils permettent d'établir des communications entre véhicules ou bien avec une infrastructure située aux bords de routes dans le but d'améliorer la sécurité routière ou de permettre l'accès à Internet pour les passagers.

Pour la mise en place d'un tel réseau, certains équipements électroniques doivent être installés au sein de véhicules (de calculateurs, de cartes réseaux, de capteurs, d'un système de localisation GPS et bien sûr une plateforme de traitement). Plusieurs technologies peuvent être mises en œuvre pour l'établissement des communications véhiculaires, tel : les réseaux sans-fil de type 802.11, WIMAX, Bluetooth...etc.

Un exemple de réseau VANET urbain est illustré dans la figure 1.5.

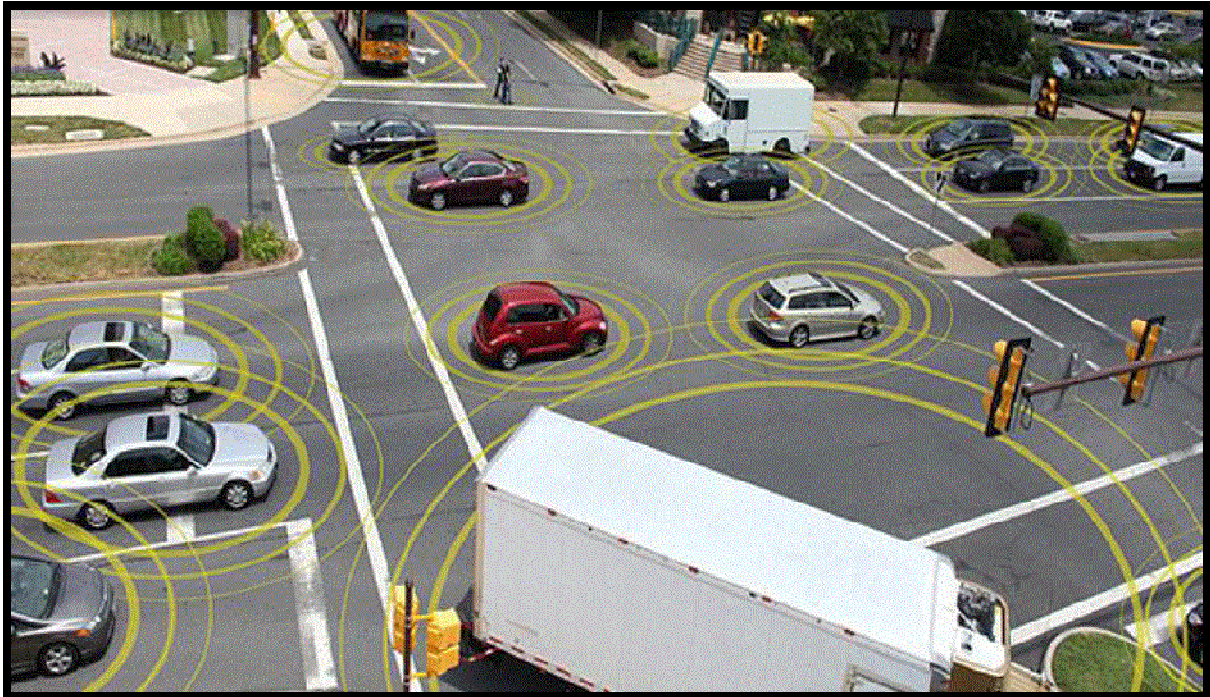


Figure I.5: Un exemple d'un réseau VANET

I.4.2. Historique :

C'est au début des années 80 que les premières études, portant sur des communications entre véhicules et infrastructure, ont été menées au Japon à travers le projet JSK (Abbréviation japonaise pour "Association of Electronic Technology for Automobile Traffic and Driving"). [7][8]

La recherche sur ces réseaux a vraiment commencé à croître au début des années 2000, avec la création des projets tels que «CarTalk 2000» en Europe qui tente de résoudre des problèmes liés à la sécurité ou au confort des passagers par l'intermédiaire de communications inter-véhiculaires [4], «Advance Safety Vehicle 3» au Japon, et «Vehicle Safety Communication» aux États-Unis. [8]

Depuis 2002, avec le développement rapide de technologies sans fil, le nombre de publications dans le domaine des réseaux de véhicules a rapidement augmenté, divers workshops ont été créés, comme ACM International Workshop on Vehicular Ad Hoc Network en 2004 ou International Workshop on Intelligent Transportation en 2003.

Chapitre I : Vue d'ensemble sur les réseaux VANETs

En 2009, l'Institute of Electrical and Electronics Engineers (IEEE), a publié le standard 802.11p qui définit l'accès sans fil dans les réseaux véhiculaires mobiles (Wireless Access in Vehicular Environment (WAVE)). [7][8]

Très vite, les chercheurs se sont intéressés à un nouveau type d'adressage et de routage des paquets de données. Plutôt que d'utiliser la méthode IP, il a été mis en place une méthode géographique appelée **GeoCast**, basée sur les standards DSRC et IEEE 802.11, pour fournir une communication multi-sauts sans fil, et ce, sans l'aide d'infrastructure (par comparaison avec le Global Positioning System (GPS) qui nécessite un réseau de satellite).

GeoCast permet ainsi d'étendre les communications de courte portée, fournie par DSRC, à des communications multi-sauts [6]. Geocast a d'abord été conçu pour les MANETs, mais a très vite été étendu à d'autres réseaux tels que les VANETs, les réseaux de capteur sans fil (Wireless Sensor Networks WSN), ou les réseaux maillés (Mesh Networks). Ce type d'adressage est en effet bien adapté aux réseaux mobiles dynamiques, ayant des changements de topologies fréquents, avec ou sans infrastructure, et supportant des connexions intermittentes. [8]

I.4.3. Le Nœud de réseau VANET :

Un nœud d'un réseau VANET est un véhicule équipé de plusieurs terminaux tels que les calculateurs, les interfaces réseaux ainsi que des capteurs capables de collecter les informations et de les traiter. On parle de « véhicule intelligent ». La figure I.6 modélise un véhicule intelligent.

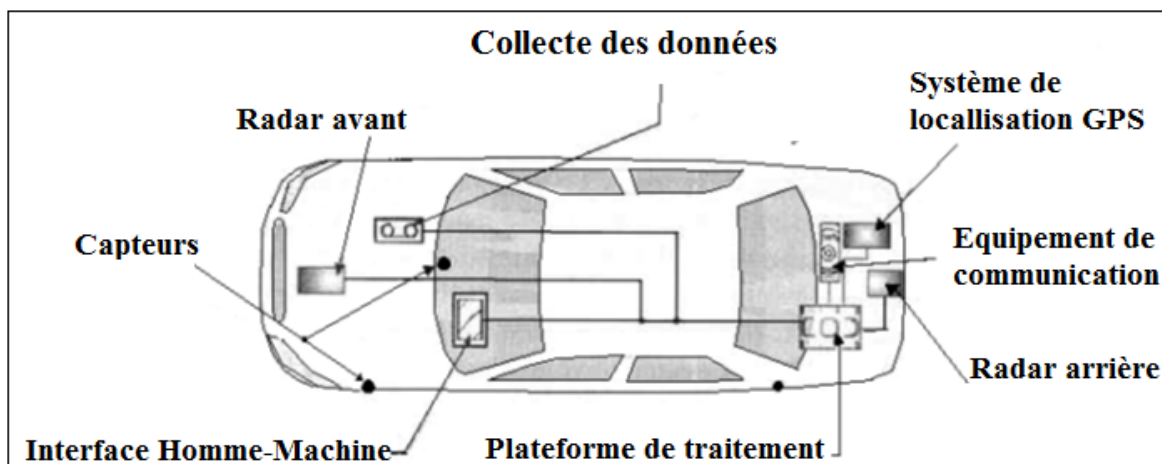


Figure I.6 : Les éléments constituant le véhicule intelligent.

I.4.4. Systèmes de transport intelligents (STIs) :

Les Systèmes de Transport Intelligents (STI) désignent les applications des nouvelles technologies de l'information et de la communication au domaine des transports [9], pour les rendre plus automatisés, permettant d'augmenter les performances au delà des limites sensorielles des conducteurs humains.

Un STI a la possibilité d'accroître la sécurité du trafic, son efficacité, le confort des voyageurs.

I.4.5. Station de bord de la route (RSU) :

Les Stations de bord de la route sont des stations de bases, ou points d'infrastructure RSU (Road Side Units), ils permettent d'établir une communication de l'infrastructure vers les véhicules (I2V) et des véhicules vers l'infrastructure (V2I).L'objectif du système coopératif résultant est d'assurer la continuité du message d'information de véhicules à véhicules en passant par les unités de bord de la route.

Les entités installées au bord des routes diffusent aux véhicules les informations sur l'état du trafic et sur les conditions météorologiques ; elles peuvent être aussi utilisées comme point d'accès au réseau pour connecter les véhicules aux autres réseaux à infrastructures comme internet.

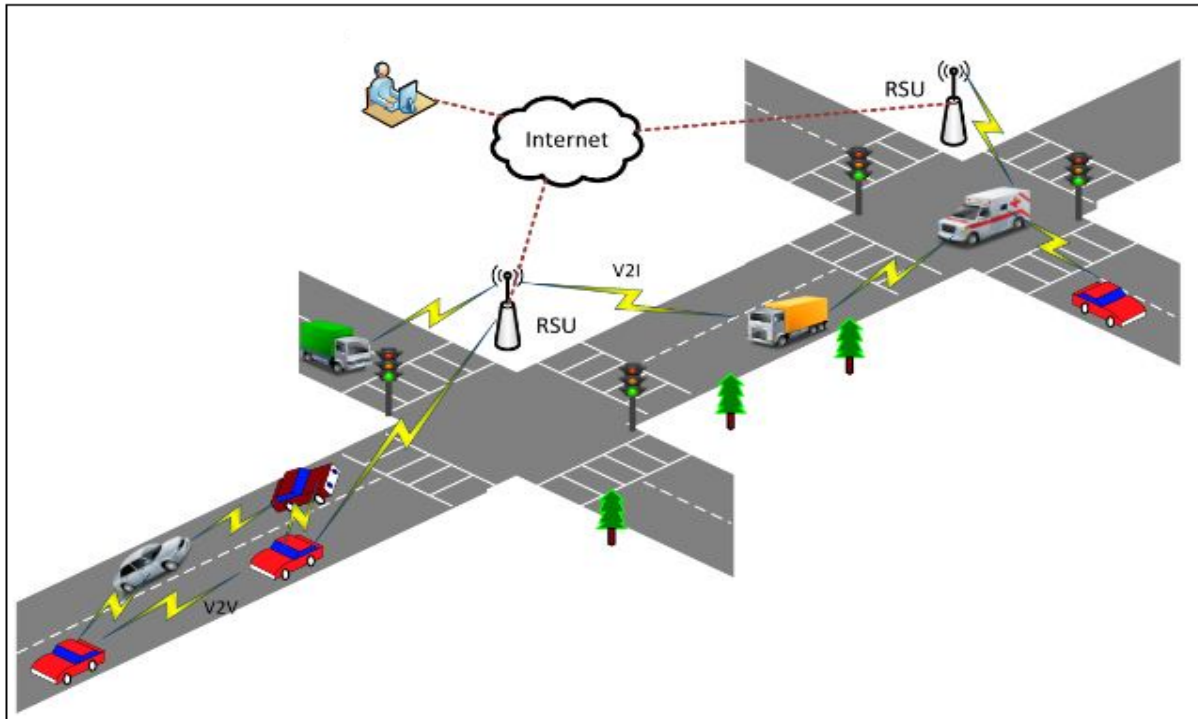


Figure I.7 : Communication Basé sur les RSU.

I.4.6. OBU (On-Board Unit) :

Ce sont des unités embarquées dans les véhicules intelligents, elles regroupent un ensemble de composants matériels et logiciels de hautes technologies (GPS, radar, caméras, différents capteurs et autres). Leurs rôles sont d'assurer la localisation, la réception, le calcul, le stockage et l'envoi des données sur le réseau. Ce sont des émetteurs-récepteurs qui assurent la connexion du véhicule au réseau.

I.5. Architecture de communication des VANETs :

Pour concevoir une architecture de communication de réseaux de type VANETs, nous abordons trois modes essentiels :

I.5.1. La communication inter-véhicule (V2V) :

On parle de communication véhicule à véhicule (V2V, Vehicle to Vehicle) lorsqu'au moins deux véhicules communiquent ensemble par l'intermédiaire de leur OBU (On Board Unit). Ce mode de communication est vu comme un cas particulier des réseaux mobiles

ad hoc où les contraintes d'énergie, de mémoire et de capacité sont relaxées et où le modèle de mobilité n'est pas aléatoire mais prévisible avec une très grande mobilité.

Ce mode de communication utilise le multicast/ Broadcast en multi sauts pour transmettre les informations relatives aux trafics à un groupe de récepteurs. Cette architecture peut être utilisée dans le scénario de diffusion d'alertes (freinage d'urgence, collision, ralentissement...) ou pour la conduite coopérative.

C'est une communication moins coûteuse et plus flexible car aucune infrastructure n'est utilisée et aucune installation n'est nécessaire sur les routes et tous les véhicules sont équipés pour communiquer directement entre eux n'importe où.

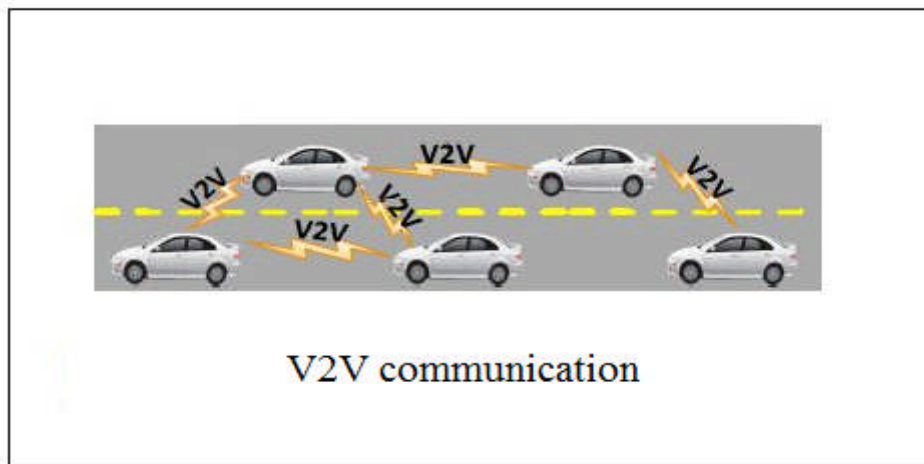


Figure I.8 : Communication véhicule à véhicule.

I.5.2. La communication entre véhicules et les stations de bord de la route (V2I) :

Dans ce mode de communication V2I (Véhicule to Infrastructure) des points d'infrastructures RSU (Road Side Units) sont déployés aux bords des routes, toute communication doit passer par eux. V2I est une architecture dédiée à la remontée d'information depuis les véhicules vers l'infrastructure RSU, en utilisant soit des points d'accès Wifi, soit des connexions 3G.

Ils peuvent aussi offrir aux utilisateurs plusieurs services concernant le trafic, accès à internet, échange de données de voiture-à-domicile et même la communication de voiture-à-garage pour le diagnostic distant.

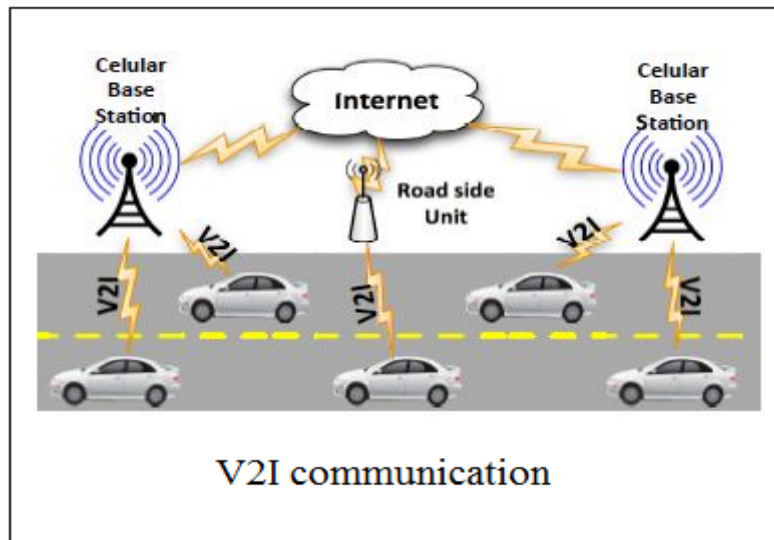


Figure I.9 : Communication véhicule à station de base.

I.5.3. La communication hybride :

La combinaison de ces deux types de communications, V2V et V2I, permet d'obtenir une communication hybride très intéressante. En effet, les portées des infrastructures étant limitées, l'utilisation de véhicules comme relais permet d'étendre cette distance.

Dans un but économique en évitant de multiplier les bornes (RSU, Road Side Units) à chaque coin de rue, l'utilisation de sauts par véhicules intermédiaires prend toute son importance. [10]

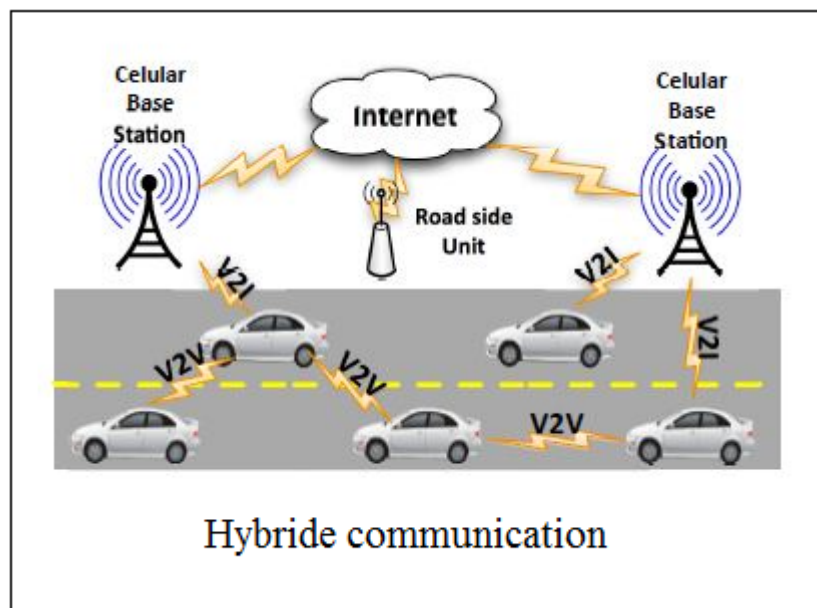


Figure I.10 : Communication hybride

I.6. Caractéristiques des réseaux VANETs : Les réseaux véhiculaires ont des caractéristiques spécifiques qui les distinguent de réseaux ad hoc mobiles parmi ces caractéristiques :

➤ **Ils n'ont pas de problème d'énergie ou de stockage :**

Les équipements du réseau VANET disposent suffisamment d'énergie qui peut alimenter les différents éléments électroniques d'une voiture intelligente. Donc, les nœuds sont censés avoir une grande capacité de traitement et de stockage de données ; contrairement au contexte des réseaux Ad Hoc Mobiles où la contrainte d'énergie représente un défi pour les chercheurs.

➤ **Ils sont très dynamiques (forte mobilité) :**

La vitesse des véhicules varie selon l'environnement allant de 0 à 160 km/h en moyenne. Ainsi si deux véhicules sont en sens inverse et qu'ils roulent tous les deux à des vitesses excessives, le temps d'interconnexion pourra être très court.

L'impact de la mobilité sur la connectivité du réseau reste l'une des difficultés majeures des réseaux véhiculaires.

➤ **Environnements de communication très variés :**

Les réseaux véhiculaires peuvent fonctionner dans trois types d'environnement : autoroutier, urbain, rural ; ils sont appelés à passer d'un environnement urbain caractérisé par de nombreux obstacles à la propagation des signaux à un environnement rurale ou autoroutier présentant des caractéristiques différentes ; contrairement aux autres réseaux sans fil qui ont un environnement spécifique selon leur fonctionnement.

➤ **Information de localisation disponible :**

Actuellement, la majorité des véhicules neufs possède un système de géolocalisation, le système peut obtenir la position géographique du véhicule, ainsi que sa direction, s'il est en mouvement.

➤ **Faible connectivité du réseau :**

La densité d'un réseau de véhicules est très variable. Une forte densité de véhicules permet au réseau d'être connexe, et donc il existe toujours un chemin entre deux nœuds qui souhaitent communiquer. A l'inverse, une faible densité de véhicules a pour conséquence un taux élevé de ruptures de communications, un délai d'acheminement plus long si le véhicule conserve le paquet, voire une impossibilité pour deux véhicules de communiquer. [3]

I.7. Modes de transmission dans les réseaux VANETs :

Les échanges de données entre un nœud et ses voisins dans un réseau véhiculaire peuvent être transmis suivant quatre façons :

I.7.1. Le mode Unicast (Point-à-point) :

Dans ce mode l'information est envoyée d'un nœud source vers un nœud destinataire bien déterminé.

I.7.2. Le mode Broadcast (diffusion) :

On dit qu'un message est inondé ou diffusé en broadcast si la source envoie un message à tous les nœuds du réseau.

I.7.3. Le mode Multicast (multipoint) :

Le nœud envoie un message à un ensemble de nœuds du réseau où le message peut être reçu par n'importe quel nœud se trouvant dans ce groupe.

I.7.4. Le mode Géocast :

La transmission dans ce mode est multicast où le message est reçu par chaque nœud d'une zone géographique déterminée.

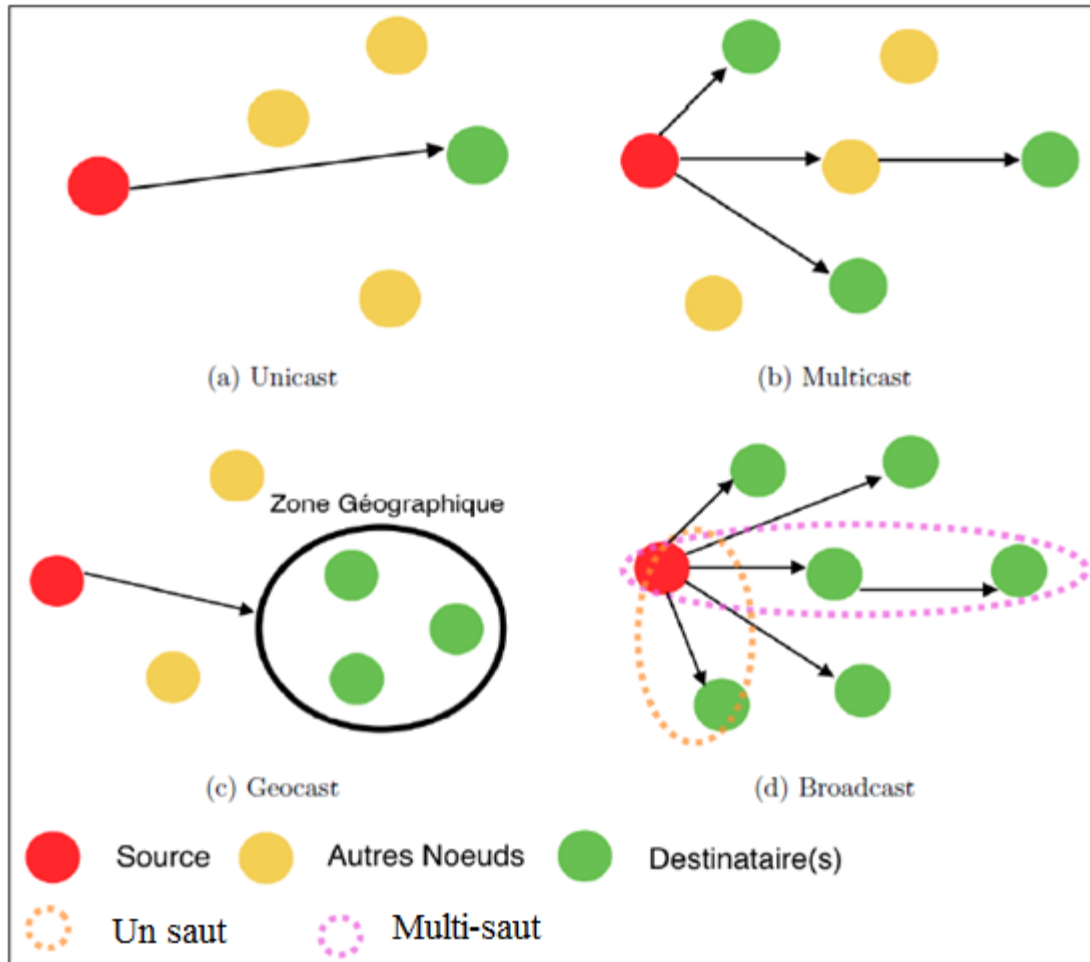


Figure I.11 : Les différents types de mode de transmission dans les VANETs.

I.8. Applications des VANETs : Nombreuses sont les applications proposées pour les réseaux véhiculaires. Elles peuvent être classifiées en trois grandes catégories :

I.8.1. Application pour la prévention et la sécurité routière :

C'est le type d'applications le plus important qui visent à améliorer la sécurité des déplacements et faire face aux accidents routiers en avisant les véhicules de toute situation dangereuse.

Ces applications se basent en général sur une diffusion, périodique ou non, de messages informatifs permettant aux conducteurs d'avoir une connaissance de l'état de la route et des véhicules voisins, alertes en cas de violations des feux de circulation, notification en cas de freinage urgent.

I.8.2. Application pour la gestion du trafic routier :

Les applications de gestion de trafic sont axées sur l'amélioration des conditions de circulation, elles visent à équilibrer la circulation des véhicules sur les routes pour une utilisation efficace de la capacité des routes et des carrefours dans le but de réduire les embouteillages, les risques d'accidents, la durée des voyages et la consommation d'énergie.

I.8.3. Application pour le confort :

En plus des services de sécurité et de gestion du trafic routier, les réseaux véhiculaires contribuent aussi à l'amélioration du confort des usagers en rendant les voyages plus agréables.

Grace à l'accès à internet les conducteurs et les passagers bénéficient de plusieurs services tel que : les jeux/discussions distribués, la gestion des parkings, la messagerie, le chat inter – véhicule, télécharger des fichiers MP3, envoyer des cartes à des amis, etc.

I.9. Technologies de communication et de diffusion pour les Vanets :

Les réseaux véhiculaires mobiles sont en théorie capables d'utiliser un grand nombre de technologies de transmission, dont celles issues de la 3ème génération (3G) et 4ème génération (4G Long Term Evolution (LTE)) de téléphonie mobile, le Wi-Fi, le WiMax, et le DSRC/WAVE [10]. Les réseaux mis en place pour les véhicules peuvent utiliser une ou plusieurs de ces technologies. [8]

➤ **Wifi (Wireless Fidelity) :**

Cette technologie utilise potentiellement plusieurs normes telles que 802.11 a/b/g/n, peut être utilisé dans les VANET. La couverture va jusqu'à une centaine de mètres en fonction de la norme et de la configuration utilisée. Le débit peut aller de 1Mbit/s jusqu'à 11Mbit/s pour 802.11b, de 6Mbit/s jusqu'à 54Mbit/s pour 802.11a/g, jusqu'à 300Mbit/s pour 802.11n. [11]

➤ **Bluetooth :**

Bluetooth est une norme sans fil (802.15.1) spécialement conçu pour les communications à courte portée Il fonctionne dans la bande des 2,4 GHz. Cependant, cette technologie est limitée par la portée de communication limitée.

➤ **3G et 4G LTE :**

Les VANETs utilisent les connexions ad-hoc. Afin de pouvoir se connecter au réseau global, et donc à Internet, le réseau cellulaire mobile peut être utilisé. Ce dernier utilise un certain nombre de transmissions, parmi lesquelles : 3G, 4GLTE. La technologie 2G/3G offre un débit de téléchargement maximal de 384Kbit/s (pour de vitesses de 3 km/h à 100 km/h) et de 144Kbit/s (<250 km/h). Avec UMTS il est de 2Mbit/s ; la 4G offre un débit de 100Mbit/s pour les utilisateurs à forte mobilité. Ainsi, puisque les réseaux véhiculaires mobiles sont hautement dynamiques, ces technologies sont peu adaptées. De plus la nécessité d'appartenance au réseau, via une carte SIM, et donc un forfait avec un opérateur, n'est pas intéressant. [8][11]

➤ **WiMax :**

Le WiMax est définie par la norme IEEE 802.16. Sa couverture peut atteindre 60 km. Son débit minimum est de 110Mbit/s et le maximum de 365Mbit/s dans la version 2 de la norme peut être utilisé dans les VANET. De son côté, WiMax permet une transmission portant sur une zone géographique étendue. Il reste donc intéressant comme moyen de communication entre les véhicules et les infrastructures, afin de permettre un accès à Internet au réseau ad-hoc formé par les véhicules. Cependant WiMax n'est pas non plus idéal dans les situations de très grande mobilité. [11]

➤ **Satellite :**

Les satellites peuvent présenter de grands avantages en raison de leur large couverture. Parmi les technologies satellite que nous pouvons utiliser, « GEO Mobile Radio Interface (GMR) » et « DVB-Satellite services to Handhelds (DVBSH)» En revanche, ils sont connus pour leurs coûts potentiellement élevés. [11]

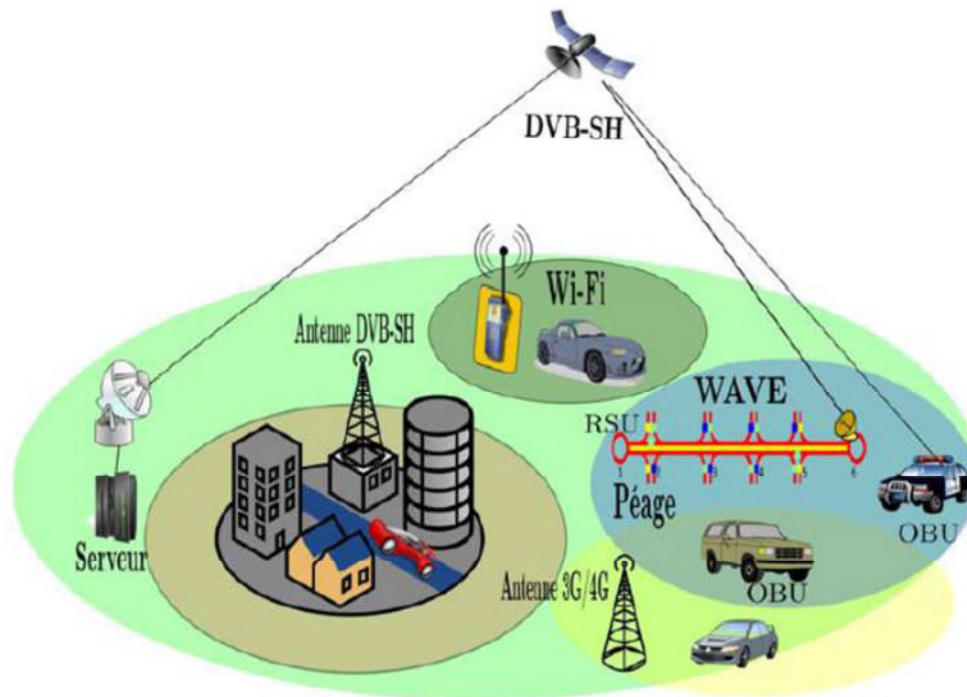


Figure I.12 : Ensemble de technologies de communication pour VANETs.

I.10. Normes et standardisation dans les réseaux VANETs :

I.10.1. WAVE (Wireless Access in Vehicular Environment) et IEEE 802.11p :

Le groupe de travail IEEE a défini un standard dédié aux communications inter véhicules, appelé WAVE (Wireless Ability in Vehicular Enveronment) aussi connu sous le nom de IEEE 802.11p ; plus tard IEEE a entrepris des travaux pour un nouveau standard connu sous le nom de IEEE 802.11p/WAVE (Wireless Access in Vehicular Enveronments), cette norme utilise le concept de multicanaux afin d'assurer les communication pour les applications de sécurité et les autres services du Système de Transport Intelligents (STI).[8]

La norme IEEE 802.11p est une norme pour la couche d'accès au médium, couche mac, se basant principalement sur la norme 802.11a et fonctionnant dans la bande de fréquence DSRC.

WAVE s'appuie sur le protocole IEEE802.11p au niveau de la couche MAC et la couche physique. Au niveau de la couche MAC, 802.11p est basé sur CSMA/CA comme protocole 802.11a.

I.10.2. DSRC (Dedicated Short Range Communications) :

Un spectre de fréquences dédié aux communications sans fil dans les STI (Système de Transport Intelligent), appelé DSRC (Dedicated short-range communications) a été attribué dans différentes régions du monde ; fonctionnant dans la bande des 5.9 GHz aux Etats-Unis ou les 5.8 GHz en Europe et au Japon, a été spécifiquement conçu pour les communications véhiculaires dont l'objectif est d'assurer des services tel que : le système de télépéage sur autoroute, l'accréditation et la surveillance des opérations des véhicules commerciaux.

Cette bande de fréquence étant trop étroite pour supporter l'évolution envisagée pour les applications dans les réseaux véhiculaires.

Aux Etats-Unis, en 1999 la FCC (Federal Communications Commission) a accordé une bande passante de 75MHz dans la gamme de fréquences [5,850-5,925] GHz pour les communications à courte portée dédiées aux STI ce qui a donné naissance à la technologie DSRC. [13]

Le standard DSRC permet d'établir une communication véhicule-à-véhicule (V2V) ainsi qu'une communication véhicule-à infrastructure (V2I). Il est compatible avec les contraintes des réseaux de véhicules fortement dynamiques. En effet, il offre une fiabilité de communication ainsi qu'une faible latence lors de l'établissement de la communication.

Les caractéristiques de DSRC sont :

- Il supporte une vitesse des véhicules dépassant 200km/h.
- Il offre une portée radio variant entre 300 et 1000 mètres.
- Il garantit un temps de latence pour l'établissement de la communication ne dépassant pas 50 ms.
- Il permet un débit théorique (bande passante) allant de 3 à 27 Mbps dans un rayon de 1000mètre. [14]

I.10.3. ISO : TC204 /WG16-CALM :

L'organisation International de normalisation (ISO) :TC204/WG16 a fixé un ensemble de règles sous forme d'un référentiel relatif à l'interface radioélectrique nécessaire en courte ou en longue portée connu sous le nom de CALM (Continuous Air interface, Long and Medium Range). [12]

L'objectif de CALM consiste à développer une plateforme logicielle embarquée dans les véhicules qui assurera l'interface entre plusieurs technologies de communication permettant ainsi une mobilité sans coupure. C'est elle qui choisira automatiquement de passer par du WIFI, du WIMAX ou du DSRC en fonction de la disponibilité des réseaux et des messages à transmettre.

I.10.4. Le consortium Car-to-Car (C2C-CC) :

Le CAR 2 CAR Communication Consortium (C2C-CC) est une organisation non lucrative, née de l'industrie, initiée par les constructeurs automobiles européens et supportée par les équipementiers, les organismes de recherche et autres partenaires.

L'objectif de C2C-CC est d'améliorer la sécurité et l'efficacité du trafic routier à l'aide de systèmes de transport intelligents, équipés de solutions de communication entre véhicules (V2V) et entre véhicules et infrastructures (V2I). Il soutient la création d'une norme européenne destinée aux véhicules communicants de toutes marques. En tant que contributeur clé, le consortium CAR 2 CAR Communication collabore étroitement avec les organismes de normalisation internationaux et européens. [13]

I.11.Projets de recherche dans le cadre des réseaux véhiculaires :

Plusieurs projets de recherches de développement et de standardisation de la communication automobile ont été lancés partout dans le monde entier aux USA, en Europe et au Japon parmi ces projets on va citer quelques-uns :

- **Se Ve Com (Secure Vehicular Communication) :** Est un projet financé par l'union européenne s'intéresse à la sécurité des futurs réseaux de véhicules, sa problématique est de définir une architecture des VANET, incluant la sécurité et l'anonymat de la communication véhicule à véhicule et véhicule à infrastructure.

- **NoW (Network on Wheels)** : Est un projet de recherche allemand, fondé par des constructeurs automobiles et des acteurs de télécommunication et des universités. Son objectif est de résoudre les questions techniques clés relatives aux protocoles de communication et à la sécurité des données dans les communications V2V et de présenter les résultats au consortium européen de normalisation Car 2 Car Consortium Communications.
- **VII (Vehicle-Infrastructure Integration)**: Est un projet américain issu de la collaboration du département US DOT (United States Department of Transportation) avec plusieurs constructeurs automobiles. Son objectif est de fournir une variété d'applications de sécurité des véhicules et d'applications à intérêt commercial pour les constructeurs automobiles en utilisant la technologie DSRC et le système GPS. [13]
- **Fleet Net** : Son objectif principal est le développement d'une plate-forme basée sur la technologie UMTS pour les communications inter-véhicules et l'accès à Internet.
- **VSC (Vehicle Safety Communication)**: Est un projet américain qui réunit sept constructeurs automobiles dont Mercedes-Benz, Ford et Toyota avec US DOT (United States Department of Transportation) ; a pour objectif d'estimer le potentiel des applications de sécurité basées sur la communication inter véhicules et de déterminer les besoins en communication des applications de sécurité de haute priorité. [13]
- **Car TALK 2000** : Est un projet européen qui s'est focalisé sur les systèmes d'aide à la conduite, basés sur la communication inter-véhicules ; son but principal est de travailler sur la conduite coopérative et les réseaux autonomes.

12. Discussion :

Nous avons présenté dans ce chapitre les réseaux sans fil, les réseaux Ad Hoc et une vue d'ensemble sur les réseaux Ad Hoc véhiculaires VANET, leurs caractéristiques et leurs applications. Pour que les véhicules puissent communiquer entre eux, un protocole de routage doit être défini. En effet, quand les terminaux ne sont pas à une portée de transmission radio directe, le routage est exigé pour établir la communication entre les véhicules ; la conception d'un protocole de routage efficace pour VANETs est un défi majeur.

Les protocoles déjà présents dans les réseaux Ad Hoc Mobiles ne sont pas tous adaptés pour les VANETs à cause des caractéristiques bien spécifiques telles que la forte mobilité qui entraîne une topologie très dynamique et la propagation des signaux radio.

Dans le chapitre suivant, nous présenterons les différents protocoles de routage dans les réseaux véhiculaires Ad hoc (VANET).

II.1. Préambule :

VANET (Vehicular Ad hoc Network) est un réseau sans infrastructure fixe, constitué d'un ensemble de véhicules dénommés nœuds mobiles. Ces nœuds établissent entre eux une communication dite à un-saut lorsqu'ils sont à portée radio l'un de l'autre. Une communication dite multi-sauts peut également être établie entre deux nœuds distants (nœuds n'étant pas dans une même zone de transmission), réalisable grâce à la mise en place d'un chemin de routage reliant le nœud source au nœud destinataire et impliquant un ou des nœuds intermédiaires, dits aussi nœuds relais. [15]

II.2. Concepts de base du routage :

II.2.1. Le routage :

La fonction de routage est l'une des plus importantes dans les réseaux de communication, que ce soit dans les réseaux filaires ou dans les réseaux sans fil. Le routage sert à faire communiquer des nœuds distants. C'est un mécanisme qui permet l'établissement et le maintien de routes optimales entre les nœuds d'un réseau. [4]

II.2.2. Définition d'un protocole de routage :

Est un ensemble de règles de communication respectées par tous les systèmes interconnectés afin de permettre la liaison entre systèmes émetteurs et systèmes récepteurs. [16]

II.2.3. Objectif du routage :

Le routage dans les réseaux VANET est un problème très difficile à gérer et un axe de recherche pour beaucoup de chercheurs. Pour que les véhicules puissent communiquer entre eux, un protocole de routage doit être défini. En effet quand les terminaux ne sont pas à une portée de transmission radio directe, le routage est exigé pour établir la communication entre les véhicules.

II.3. Problématique de routage dans les réseaux ad-Hoc Véhiculaire :

Les réseaux véhiculaires ont comme caractéristique principale une forte mobilité qui entraîne une topologie très dynamique. En effet, dans les VANETs, la vitesse peut être beaucoup plus élevée dans certains environnements de communication comme les autoroutes. Relativement la vitesse, la distance entre deux nœuds peut augmenter très rapidement et être

supérieure à la portée de transmission des nœuds qui est assez limitée [17]. En conséquence, le lien entre les nœuds en question sera rompu et ils ne pourront plus communiquer directement. Ceci peut se produire fréquemment et affecter considérablement le bon acheminement des paquets dans le réseau. Une solution pour améliorer la connectivité du réseau consiste à utiliser les véhicules participants comme relais, établissant ainsi des communications multi-sauts.

II.4. Les protocoles de routage dans les réseaux VANETS :

Les protocoles de routage définissent la façon dont les différentes entités du réseau échangent en permanence des informations relatives à la topologie et à l'état des liens afin de maintenir les tables de routage à jour et de déterminer les meilleures routes en fonction de l'encombrement du réseau permettant d'assurer la connectivité du réseau.

Pour s'avérer efficace dans les VANETS, un protocole de routage se doit d'assurer un temps de connexion minimal tout en utilisant de façon optimale les ressources du réseau.

Afin de présenter les principaux protocoles de routages dans les VANETS, nous avons choisi de commencer par faire une classification des différents protocoles existants. [8]

On distingue principalement quatre catégories de protocoles de routage dans les VANETS :

- Les protocoles qui se basent la topologie.
- Les protocoles qui se basent sur la position géographique.
- Les protocoles basés sur les clusters.
- Les protocoles basés sur le mode de diffusion.

Un grand nombre de protocoles de routage existent dans les VANET, La figure II.1 présente les principaux protocoles.

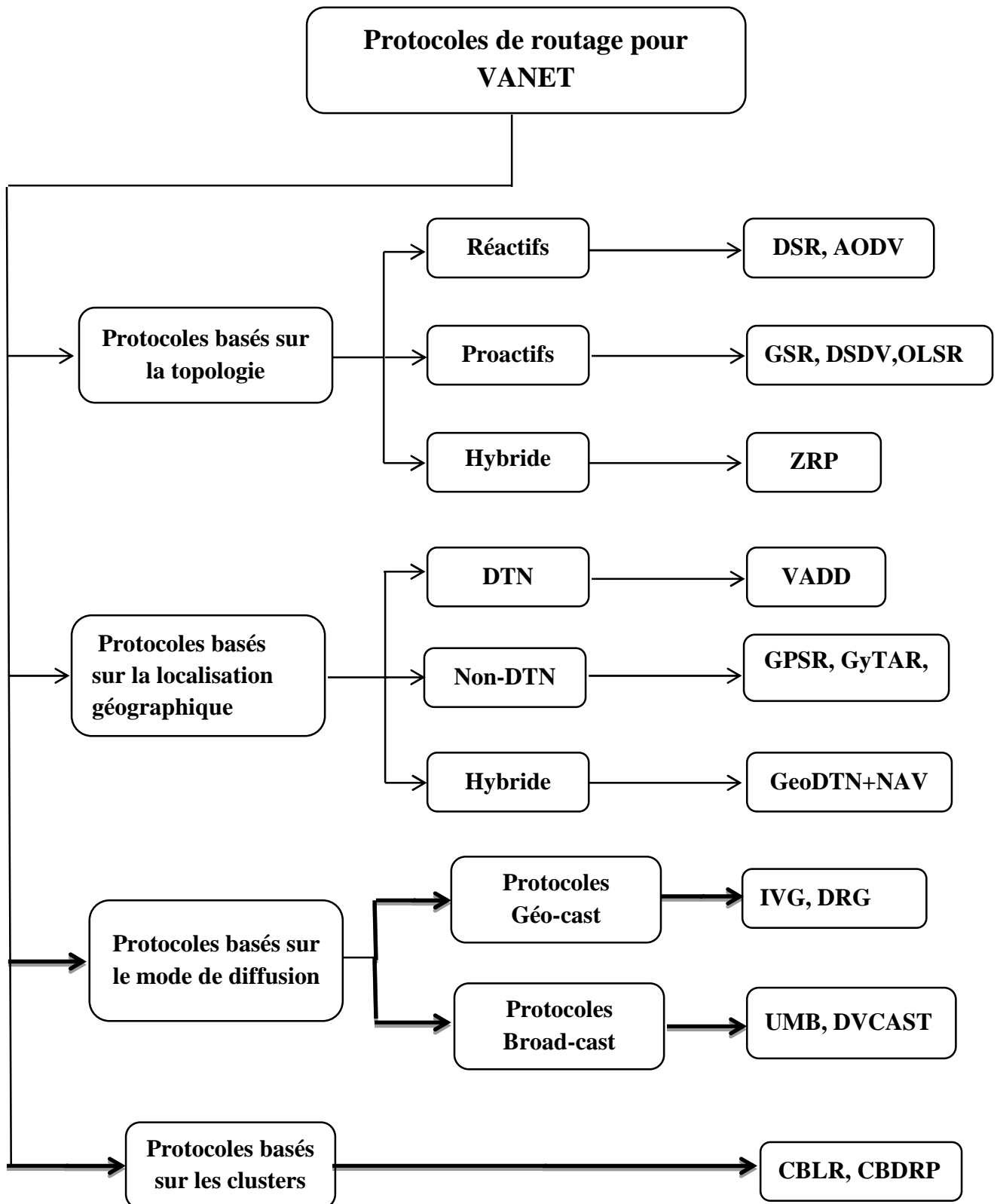


Figure II.1 : Classification des protocoles de routage dans les VANETs.

II.4.1. Les protocoles basés sur la topologie :

Les protocoles de routage basés sur la topologie utilisent les informations sur les liens qui existent entre les nœuds pour l'acheminement des paquets. Cette famille de protocoles peut être divisée en trois catégories : proactifs, réactifs et hybrides. [8]

Chaque nœud utilise comme données l'état de ses connexions avec ses nœuds voisins ; cette information est ensuite transmise aux autres nœuds pour leur offrir une connaissance plus précise sur la topologie du réseau. [18]

Les protocoles basés sur la topologie génèrent un nombre conséquent de messages de contrôle pour l'établissement et le maintien des routes. [8]

II.4.1.1. Les protocoles réactifs (On-Demand Driven) :

Les protocoles de routage réactifs dits aussi protocoles de routage à la demande sont les plus récents proposés dans les réseaux sans fil. [8] Le principe de ces protocoles est de lancer le processus de recherche de routes uniquement en cas de besoin. Cela permet d'économiser de la bande passante et de l'énergie [14]. Lorsqu'un paquet doit être envoyé d'un nœud source vers un nœud cible, le protocole de routage va rechercher un chemin jusqu'à la destination, une fois ce chemin trouvé, il est inscrit dans la table de routage et peut être utilisé tant que la destination est joignable ou jusqu'au moment où la route devient inutile. [18]

L'étape de découverte de route est initiée par la diffusion d'un message de recherche de route. Tout nœud qui reçoit ce message et qui ne dispose pas d'informations à propos de la destination diffuse à son tour le message. Ce mécanisme est appelé mécanisme d'inondation. [16]

II.4.1.1.1. Le protocole DSR (Dynamic Source Routing) :

DSR pour Dynamic Source Routing est un protocole de routage réactif destiné aux réseaux ad hoc. Il est basé sur la technique "routage à la source" [19] dans laquelle la source inclut dans l'entête du paquet la route complète par laquelle un paquet doit passer pour atteindre sa destination. Les nœuds entre la source et la destination « nœuds intermédiaires » n'ont pas besoin de maintenir à jour dans leurs tables les informations sur la route traversée puisque la route complète est insérée dans l'entête du paquet de données avant sa transmission et cela permet de résoudre facilement le problème des boucles.

DSR est composé de deux processus : la découverte de route (Route Discovery) et la maintenance de route (Route Maintenance).

- **Découverte de routes :**

Permet de chercher les routes nécessaires à la demande lorsqu'on ne possède pas de route vers le destinataire, le nœud mobile déclenche une découverte de routes en diffusant un paquet de requête de route (Route Request) [19] contenant l'adresse de la destination recherchée, une liste dans laquelle les adresses des nœuds traversés sont conservées ainsi qu'un identifiant de requête. En recevant une requête de route, un nœud vérifie qu'il ne l'a pas déjà traitée grâce à l'identifiant de requête et en vérifiant que son adresse n'est pas déjà dans le chemin. [20]

Il va alors s'enregistrer dans le chemin et la propager à tous ses voisins. Lorsque la destination de la requête est atteinte, elle renvoie un paquet de réponse de route (Route Reply) vers la source en utilisant le chemin construit lors de la propagation de la requête.

- **Maintenance de routes :**

Permet de s'assurer de la maintenance des routes tout au long de leur utilisation. Si la transmission s'avère impossible, le nœud mobile qui a détecté la défaillance envoie un paquet d'erreur (Route Error) vers la source [20], indiquant quel nœud est à l'origine du problème. La source supprime alors les routes de son cache contenant le lien erroné. Si la source dispose d'une autre route vers la même destination elle poursuit l'émission avec sa nouvelle route. Dans le cas contraire, elle effectue une nouvelle découverte de routes. [19]

II.4.1.1.2. Le protocole AODV (Ad-hoc On Demand Distance Vector):

AODV (Ad-hoc On-Demand Distance Vector) est un protocole réactif à vecteur de distance destiné aux réseaux mobiles Ad hoc [21]. Dans ce protocole, les nœuds se réfèrent à leurs tables de routage comme une base pour acheminer les paquets. Ces tables de routage contiennent des informations qui sont mises à jours à partir des données contenues dans les paquets de contrôle. [18]

Lorsqu'un nœud source désire envoyer des données vers un destinataire, il vérifie tout d'abord dans sa table de routage s'il existe une route valide vers ce destinataire. Si la route n'est pas trouvée, le nœud source lance la procédure de découverte de route en diffusant en broadcast un paquet RREQ (Route Request message) à la recherche d'un chemin vers le destinataire.

A la réception de ce paquet, le nœud répond par un paquet RREP (Route Reply message) s'il est lui-même le destinataire ou s'il possède dans sa table une route vers la destination. Dans le cas contraire, c'est-à-dire si la table de routage ne contient pas de route vers le destinataire, le nœud rediffuse le RREQ. Une fois la route trouvée, le nœud source transmet les paquets de données en transitant de proche en proche et chaque nœud détermine le prochain relais à partir de sa table de routage. En cas de rupture de route, le nœud intermédiaire envoie un paquet RERR (Route Error message) pour informer la source qui décide ou non de recommencer l'envoi du paquet. [9]

En plus des messages (RREQ, RREP, RERR), l'AODV utilise des paquets contrôle **HELLO** qui permettent de vérifier la connectivité des routes.

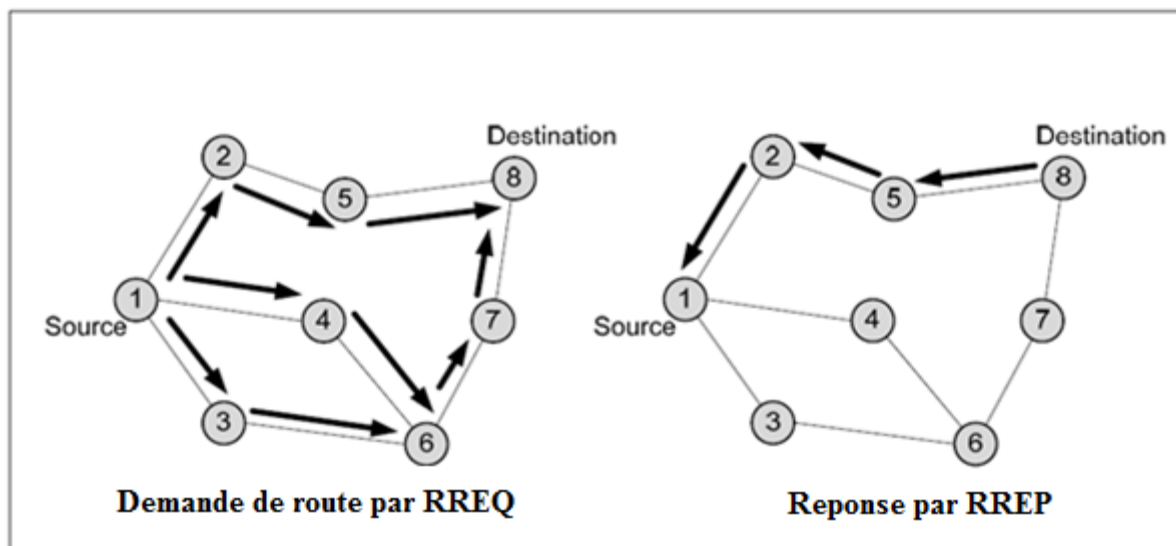


Figure II.2 : Procédure de découverte de route dans AODV.

L'AODV utilise le principe de numéro de séquence afin d'éviter le problème des boucles infinies et des transmissions inutiles de messages sur le réseau, en plus il permet de maintenir la consistance des informations de routage.

II.4.1.2. Les protocoles proactifs (Table-Driven) :

Le principe de base des protocoles proactifs est de calculer les routes à l'avance, il maintient à jour ces tables de routage, de tel sorte que lorsqu'un nœud désire envoyer un paquet à un autre nœud, une route soit immédiatement connue. [22]

Chaque nœud garde une image de la topologie de tout le réseau, Cette image est mise à jour par un envoi continu de paquet de contrôle. Un des avantages de ces protocoles est la disponibilité immédiate de la route lors du besoin. Cependant, la bande passante diminue le trafic généré par l'échange de paquets de contrôle.

II.4.1.2.1. Le protocole GSR (Global State Routing) :

Le protocole GSR est un protocole proactif. Il utilise les idées du routage basées sur l'état des liens (Link State, LS), et les améliore en évitant le mécanisme inefficace d'inondation des messages de routage. GSR utilise une vue globale de la topologie du réseau, comme c'est le cas dans les protocoles basés sur LS. Le protocole utilise aussi une méthode, appelée la méthode de dissémination. [16]

GSR est un protocole de routage géographique qui combine le routage basé sur la localisation avec le routage basé sur la topologie des routes pour construire une connaissance adaptée à l'environnement urbain [18]. Le principe de GSR est que le véhicule source désirant envoyer des données vers un véhicule cible, calcule le chemin de routage le plus court à partir des informations géographiques d'une carte routière et en utilisant les algorithmes de recherche du plus court chemin, par exemple « Dijkstra. » A partir du chemin de routage calculé, le véhicule source sélectionne ensuite une séquence d'intersections par lesquelles le paquet de données doit transiter afin d'atteindre le véhicule destinataire. Cette séquence d'intersections est constituée d'un ensemble de points géographiques fixe de passage du paquet de données pour envoyer les messages d'une intersection à une autre.

II.4.1.2.2. Le protocole OLSR (Optimized Link State Protocol):

OLSR pour Optimized Link State Protocol est un protocole de routage proactif, considéré comme une optimisation du protocole à état des liens filaires pour les réseaux mobiles Ad Hoc [19]. Il est utilisé dans les réseaux denses et peu mobiles. Il a pour objectif de fournir des routes de plus court chemin vers une destination en termes de nombre de sauts en utilisant l'algorithme de « Dijkstra » [14]. Son innovation réside dans sa façon d'économiser les ressources radio lors des diffusions, ceci est réalisé grâce à l'utilisation de la technique des relais multipoints (MPR : Multi-Point Relaying), donc le principe est que chaque nœud construit un sous ensemble appelé MPR, parmi ses voisins, qui permet d'atteindre tous ses voisins à deux sauts, les nœuds de cet ensemble servent à acheminer et retransmettre les

messages qu'ils reçoivent. Les voisins d'un nœud qui ne sont pas MPRs, lisent et traitent les paquets mais ne les retransmettent pas.

Le processus de construction des routes dans OLSR passe par les étapes suivantes [14] :

- Découverte du voisinage.
- Sélection des relais multipoints.
- Annonce des MPRs et diffusion des voisinages.
- Calcul des tables de routage.

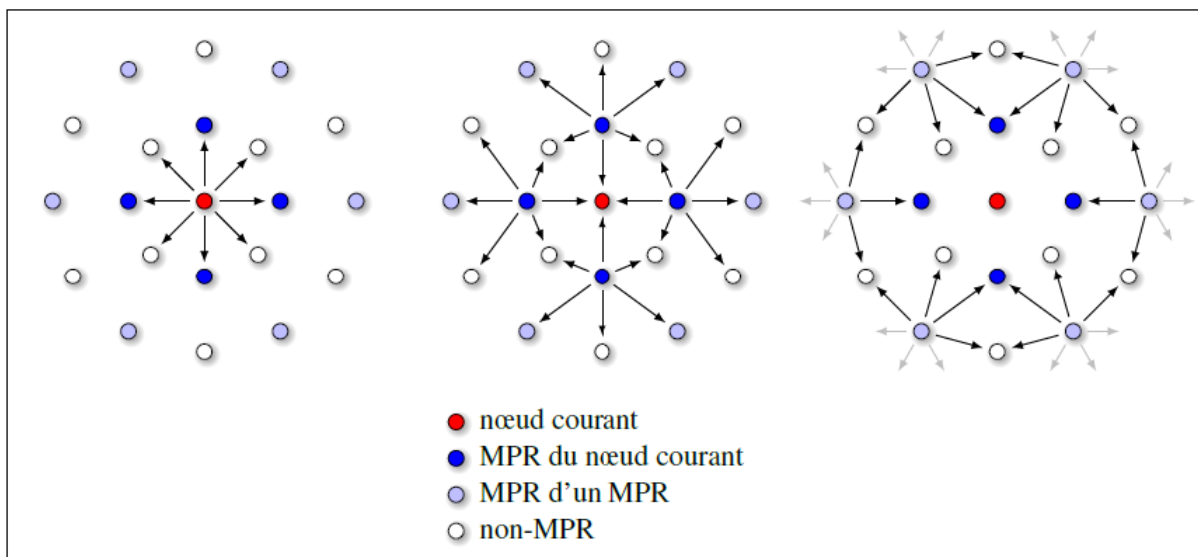


Figure II.3 : Sélection des MPRs dans OLSR.

II.4.1.2.3. Le protocole DSDV (Destination-Sequenced Distance-Vector) :

L'algorithme DSDV a été conçu spécialement pour les réseaux mobiles. Chaque nœud maintient une table de routage qui contient des informations sur les destinations accessibles dans le réseau. [21] Ces informations comprennent : le nœud suivant utilisé pour atteindre la destination, le nombre de sauts qui sépare le nœud de la destination et le numéro de séquence (SN) qui correspond à un nœud destination. Ce numéro de séquence permet de distinguer les nouvelles routes des anciennes et d'éviter la formation de boucles de routage. Les mises à jour des tables sont transmises périodiquement à travers le réseau. Quand un nœud reçoit un paquet de mise à jour, il le compare avec les informations existantes dans sa table de routage. Toute entrée dans la table est mise à jour si l'information reçue est plus récente (ayant un

numéro de séquence plus grand), ou si elles ont le même numéro de séquence mais avec une distance plus courte.

II.4.1.3. Protocoles hybrides :

Les protocoles hybrides combinent les deux approches proactives et réactives. Ils utilisent un protocole proactif pour apprendre le proche voisin ainsi ils disposent des routes immédiatement dans le voisinage au-delà de cette zone prédéfinie le protocole hybride fait appel aux techniques du protocole réactif pour chercher des routes. L'avantage des protocoles hybrides est le fait qu'ils s'adaptent mieux aux réseaux de grandes tailles, Cependant, ce type de protocole cumule les inconvénients des protocoles proactifs et ceux des protocoles réactifs, tels que l'échange de paquets de contrôle réguliers et inondation de l'ensemble de réseau pour chercher une route vers un nœud éloigné.

II.4.1.3.1. Le protocole Le protocole ZRP (Zone Routing Protocol) :

Le protocole de routage ZRP (Zone Routing Protocol) est un protocole hybride qui combine les deux approches proactives et réactive. Le protocole ZRP divise le réseau en différentes zones. Pour chaque nœud, il définit une zone de routage exprimée en nombre de sauts maximal σ . Ainsi, la zone de routage d'un nœud inclut tous les nœuds qui sont à une distance au maximum de σ sauts. Les nœuds qui sont exactement à σ sauts sont appelés nœuds périphériques. [23]

Le routage au sein d'une zone se fait de manière proactive, via le protocole IARP (Intrazone Routing Protocol) et le routage vers les nœuds extérieurs de la zone se fait de façon réactive, grâce au protocole IERP (Interzone Routing Protocol) [22]. En plus de ces deux protocoles, ZRP utilise le protocole BRP (Bordercast Routing Protocol), dont le but de construire la liste des nœuds périphériques d'une zone et les routes permettant de les atteindre, en utilisant les données de la topologie fournies par le protocole IARP.

La procédure de recherche des chemins s'effectue comme suit :

-Si le nœud destinataire se trouve dans la zone du nœud source, le chemin est supposé déjà connu. Si non, une demande d'établissement de route RREQ est initiée vers tous les nœuds périphériques. Ces derniers vérifient si la destination existe dans leurs zones. Dans ce cas la source recevra alors un paquet RREP contenant le chemin menant à la destination. [22] Dans le cas contraire, les nœuds périphériques diffusent la requête à leurs propres nœuds

périphériques qui, à leur tour, effectuent le même traitement. Un exemple de zone dans le protocole ZRP est donné à la figure II.4.

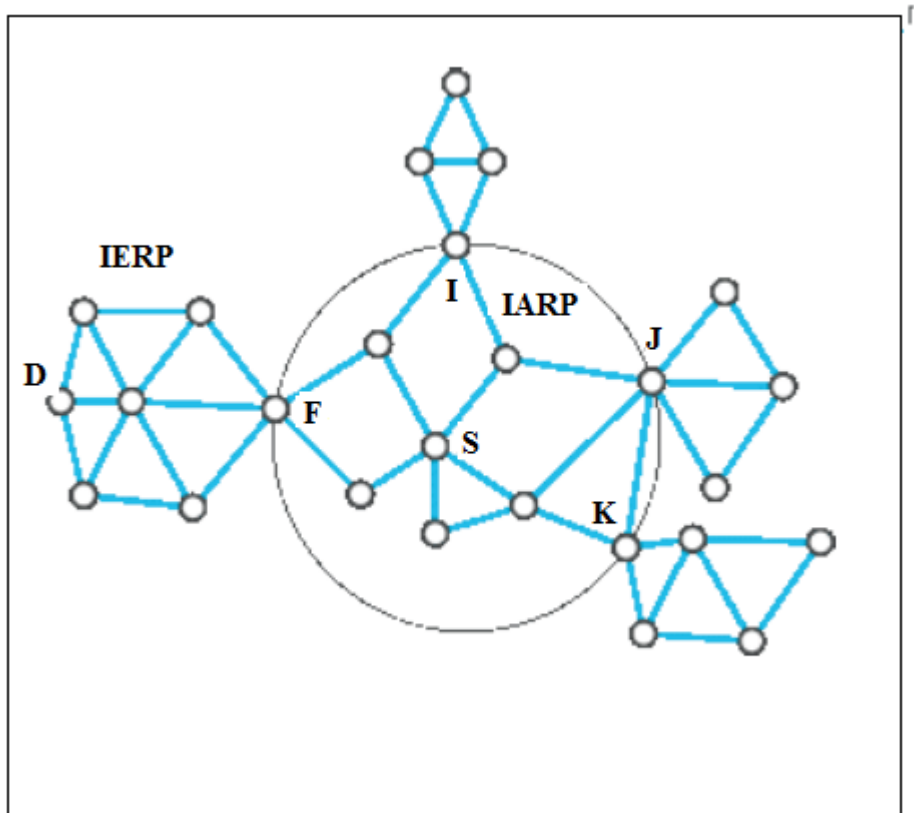


Figure II.4 : Un exemple de zone dans le protocole ZRP.

II.4.2. Les protocoles de routage avec localisation géographique :

Les protocoles de routage géographique (ou basés sur la position) utilisent des coordonnées géographiques (par exemple, fournies par un système de géolocalisation tel que le GPS afin de trouver un chemin vers la destination. Chaque nœud source inclut l'identifiant et la position de la destination dans l'entête de tout paquet à envoyer, les nœuds recevant ce paquet utilisent les informations géographiques incluses dans ce dernier et celles disponibles dans leurs tables de routage pour retransmettre le paquet et répètent le même mécanisme jusqu'à ce que celui-ci atteigne la destination. Le fait que ni les tables de routage, ni les activités de découverte de route ne sont nécessaires rend le routage géographique très attractif pour les réseaux dynamiques, tels que les réseaux véhiculaires.

L'avantage majeur de ces protocoles par rapport aux protocoles basés topologie, est qu'ils réduisent considérablement les paquets de contrôles, particulièrement dans les réseaux larges et dynamiques.

Le routage géographique se divise en trois catégories (tel qu'illustré à la figure II.2) les réseaux non-tolérants au délai (non-Delay Tolerant Network) (non-DTN), les réseaux tolérants au délai (Delay Tolerant Network)(DTN) et les réseaux hybrides. [24]

II.4.2.1. Les protocoles géographiques tolérant aux délais (DTN) :

Ce type de protocoles DTN appelé en anglais *Delay Tolerant Network* prennent en considération la déconnectivité. [21] (tient compte du cas où le réseau peut être partitionné).

On considère dans ce cadre que si aucune route complète n'existe à un instant donné entre deux nœuds, les paquets sont mis en mémoire jusqu'à la prochaine disponibilité ; ou on peut également faire transiter un message dans le réseau de proche en proche entre les nœuds accessibles dans l'espoir que l'un d'entre eux sera à un moment donné à portée du destinataire.[25]

II.4.2.1.1. Le protocole VADD (Vehicle-Assisted Data Delivery) :

Le protocole VADD (Vehicle -Assisted Data Delivery), est un protocole de routage qui prend en considération le contexte des réseaux de véhicules et exploite le mouvement prévisible des véhicules pour décider de retransmettre ou non le message.

Il utilise particulièrement les informations sur le trafic routier au niveau d'une route (vitesse maximale, densité moyenne), pour estimer le délai mis par un paquet pour parcourir un tel segment. Par conséquent, les paquets seront acheminés le long d'un chemin ayant le plus faible délai de bout en bout. [26]

Les chercheurs proposent des variantes de VADD, qui déterminent quel lien garantit le délai de livraison de données le plus faible. Le choix du prochain lien est basé sur la localisation dans la première variante (Location VADD), et sur la direction dans l'autre variante (Direction VADD). [27]

II.4.2.2. Les protocoles géographiques non tolérant aux délais (Non-DTN) :

Les protocoles de routage géographiques de type non-DTN ne prennent pas en considération la connectivité intermittente et ne sont pratiques que pour les VANET très denses. [28]

II.4.2.2.1. Le protocole GPSR (Greedy Perimeter Stateless Routing) :

Greedy Perimeter Stateless Routing, GPSR, est un protocole de routage géographique conçu pour les réseaux ad hoc véhiculaires qui exploite la correspondance entre la position géographique et la connectivité dans un réseau sans fil afin de prendre des décisions de transfert de paquets.[19]

Un nœud choisit le prochain relais en fonction des informations de position qu'il reçoit périodiquement de ces voisins, et de la destination (messages « Beacon » contenant l'adresse du nœud et sa position (x, y)). Sa particularité consiste à récupérer les informations des véhicules voisins par un équipement de positionnement GPS, si à l'extérieure ou par d'autre moyen comme les stations de base dans le cas des tunnels.

La stratégie de transmission combine une stratégie d'acheminement à vide (Greedy Forwarding) qui est utilisée à chaque fois que possible et une stratégie d'acheminement périmétrique (Perimeter Forwarding) utilisée dans les régions où la première méthode ne peut pas être appliquée. [20]

Greedy Forwarding :

Un nœud peut prendre une décision optimale et locale pour choisir le prochain nœud à qui transférer le paquet reçu. Le nœud suivant est toujours le voisin le plus proche géographiquement de la destination. Cette technique est répétée de manière récursive jusqu'à ce que la destination soit atteinte.

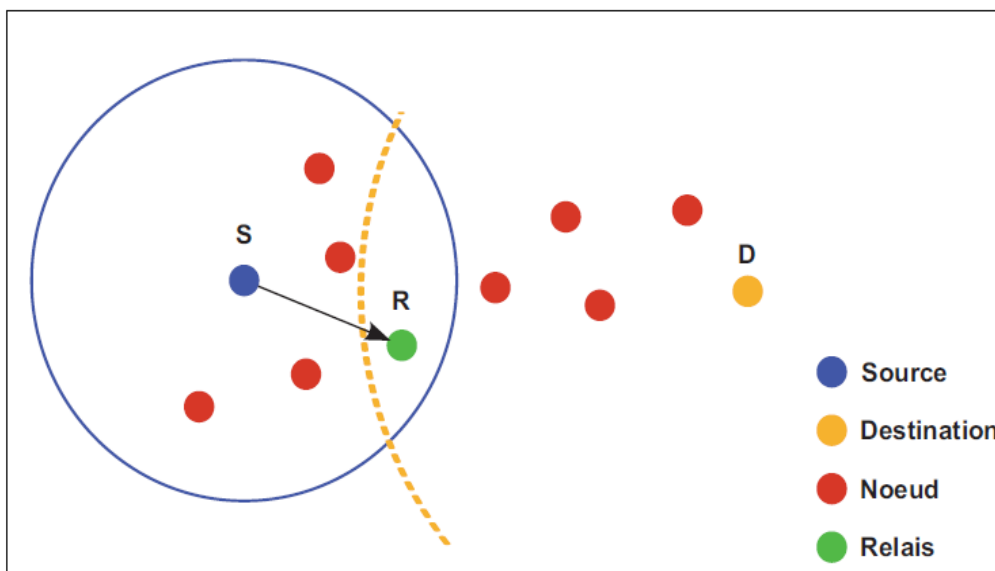


Figure II.5 : Exemple de Greedy Forwarding.

Perimeter Forwarding:

Parfois, le nœud source se trouve plus proche de la destination que tous ses voisins, il est le maximum local dans son entourage par rapport à la destination finale qui reste inaccessible en un seul saut. Le nœud est, dans ce cas, face à un "trou". Pour pallier à ce problème, la transmission de périmètre « Perimeter Forwarding » est utilisée pour router les paquets autour du trou. Ces paquets se déplacent, parcourant le chemin dans le sens inverse des aiguilles d'une montre par rapport au segment [Source-Destination] en partant du nœud source jusqu'au nœud le plus proche de la destination. [14] [29]

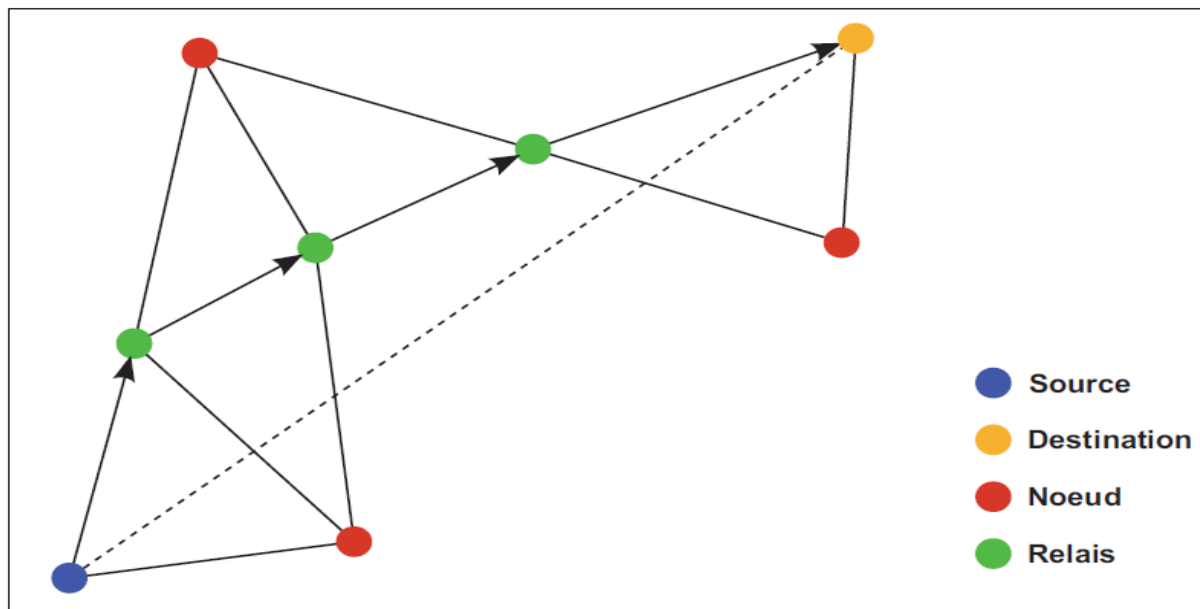


Figure II.6 : Perimeter Forwarding ou règle de la main droite.

II.4.2.2.2. Le protocole GyTAR (Greedy Traffic Aware Routing protocol) :

GyTAR est un protocole de routage géographique basé sur la localisation (position) et adapté aux réseaux véhiculaires capable de trouver des chemins robustes dans un environnement urbain. L'objectif de ce protocole est de router les données de proche en proche [26], en adoptant une approche basée sur un choix adéquat des intersections, et utilise une approche gloutonne améliorée afin d'optimiser le choix des nœuds relais pour l'acheminement des données entre deux intersections. Ce protocole suppose que chaque véhicule connaît sa position courante et ceci grâce au GPS qui sont souvent équipés par une cartographie permettant aux véhicules candidats de se localiser dans une ville et de se

Chapitre II : Les protocoles de routage dans les réseaux VANETs

positionner par rapport aux différentes intersections. L'information de localisation est donnée par un service tel que GLS (Grid Location Service) et peut déterminer la position des intersections voisines. Gytar procède par :

- **La sélection des intersections (carrefours) :**

GyTAR sélectionne dynamiquement intersection par intersection qu'on appelle les points d'ancrage "anchor paths". Ce choix progressif des intersections intermédiaires permet de sélectionner des tronçons de routes à connectivité élevée. [15]

GyTAR attribue un score à chacune de ces intersections. Ce score prend en compte deux paramètres importants :

La distance qui sépare l'intersection candidate de la destination finale est calculée en prenant en compte la topologie des routes et la densité du trafic routier au niveau du tronçon de route. Ainsi, l'intersection qui aura le score le plus élevé et par conséquent sera choisie comme prochaine intersection destination est l'intersection qui se trouve la plus proche de la destination finale et pour laquelle la densité du trafic est suffisamment élevée pour l'atteindre.

$$\text{Score (J)} = \alpha \times f(D_j) + \beta \times g(T_j) \quad [26]$$

Tel que α et β sont des facteurs de poids.

J : Intersection candidate.

D_j : La distance entre J et la destination.

T_j: Densité du trafic entre J et la destination.

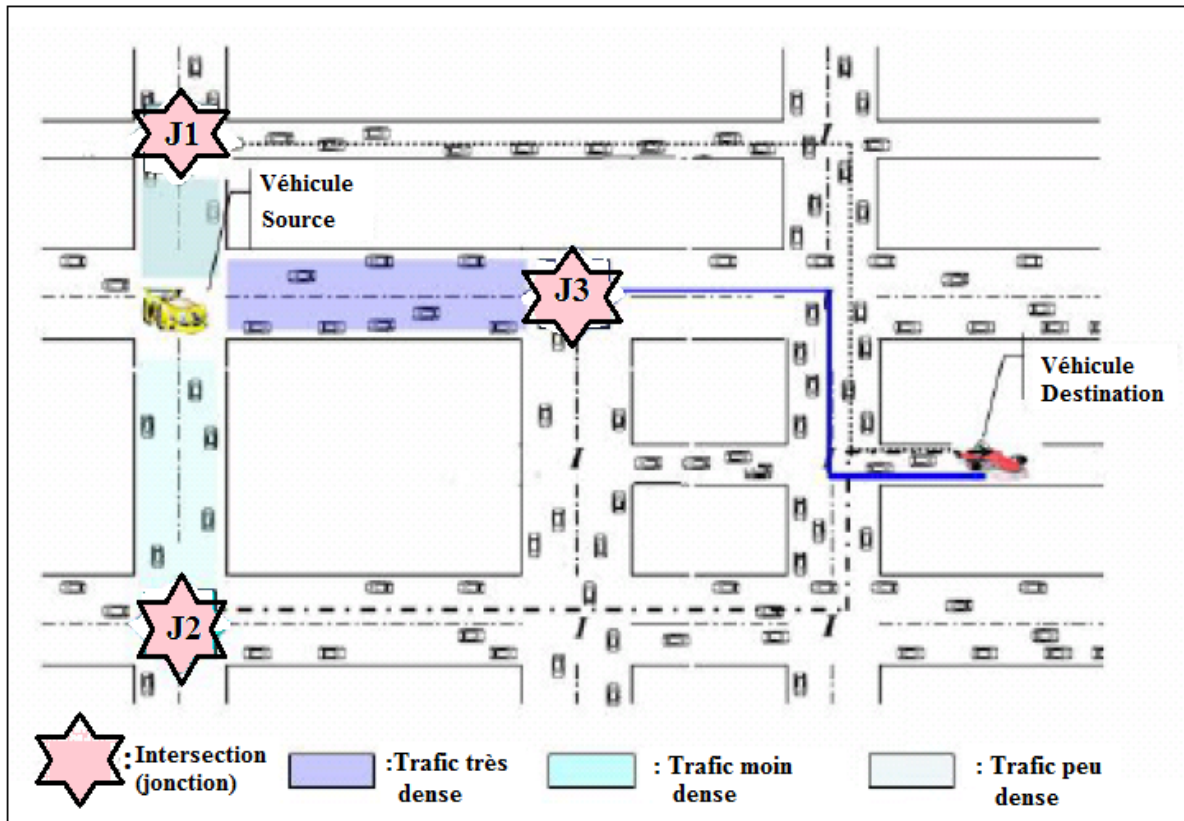


Figure II.7 : La sélection des intersections dans GyTAR.

- **L'envoi de données entre deux intersections**

GyTAR utilise une approche gloutonne améliorée pour router les paquets entre deux intersections. Chaque véhicule maintient une table dans laquelle la position, la vitesse et la direction de chaque véhicule voisin sont sauvegardées, ainsi lorsqu'un nœud désire envoyer des données il calcule la nouvelle position, pour chaque véhicule voisin, ensuite le prochain nœud voisin le plus proche à la jonction destinataire est choisi.

GyTAR utilise aussi une solution de recouvrement qui consiste à transporter les paquets par le véhicule lui-même en direction de cette intersection.

L'étude des performances de GyTAR montre qu'il minimise le délai de bout en bout tout en fournissant un faible taux de pertes des paquets de données. [30]

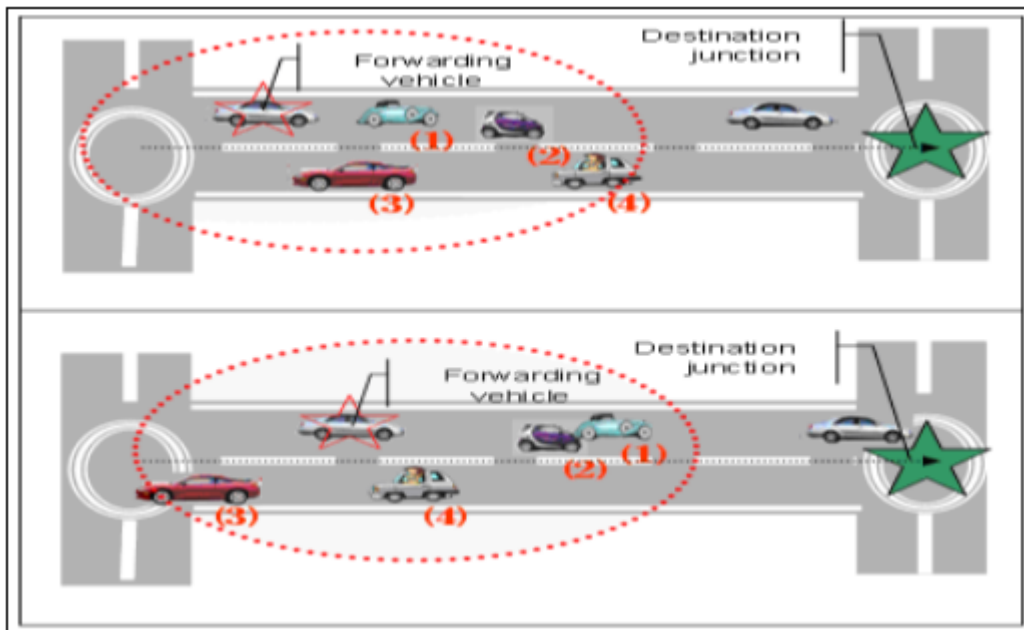


Figure II.8 : Echanges de données entre deux intersections.

II.4.2.3. Les protocoles géographiques Hybride :

Les protocoles de routage géographiques de type hybride combinent les protocoles de routage non-DTN et DTN pour exploiter la connectivité partielle du réseau.

II.4.2.3.1. Le protocole GéoDTN+NAV (Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments) :

Est un protocole qui inclut trois modes d'acheminement de données, le mode Greedy, le mode Perimeter et le mode DTN.

En mode Greedy (gourmand) un paquet est transféré vers la destination en choisissant un voisin le plus proche à la destination parmi tous les voisins. Cependant, en raison des obstacles, le paquet peut arriver à un maximum local où il n'y a pas de voisin plus près de la destination que lui-même. Dans ce cas, le mode perimeter est appliqué.

L'hypothèse selon laquelle le réseau est connecté peut ne pas toujours être vraie. En raison des caractéristiques mobiles de VANET, il est courant que le réseau soit déconnecté ou partitionné. Les deux modes Greedy et perimeter ne sont pas suffisants dans VANET. Par conséquent, ils présentent le troisième mode: DTN (Delay Tolerant Network) qui peut fournir des paquets même si le réseau est déconnecté ou partitionné (il peut passer du mode Non-DTN au mode DTN). [31] [32]

II.4.3. Les protocoles de routage basés sur les clusters (groupes) :

Dans ce type de protocole de routage, les véhicules ou les nœuds mobiles qui sont à proximité des autres forment un groupe ou (**cluster, grappes**), et dans chaque groupe un CH (**Cluster Head**) est désigné comme un chef de groupe. La formation de groupes et la sélection du chef de groupe sont des processus déterminants. [33] Chaque chef de groupe est responsable de la gestion des nœuds au sein d'un même groupe, mais également de la gestion entre les autres.

La communication diffère toute fois dans ces deux cas. La communication entre les nœuds d'un même groupe s'effectue par des liens directs entre eux, tandis que la communication entre les groupes s'effectue par le biais des chefs de groupes. [28] [34]

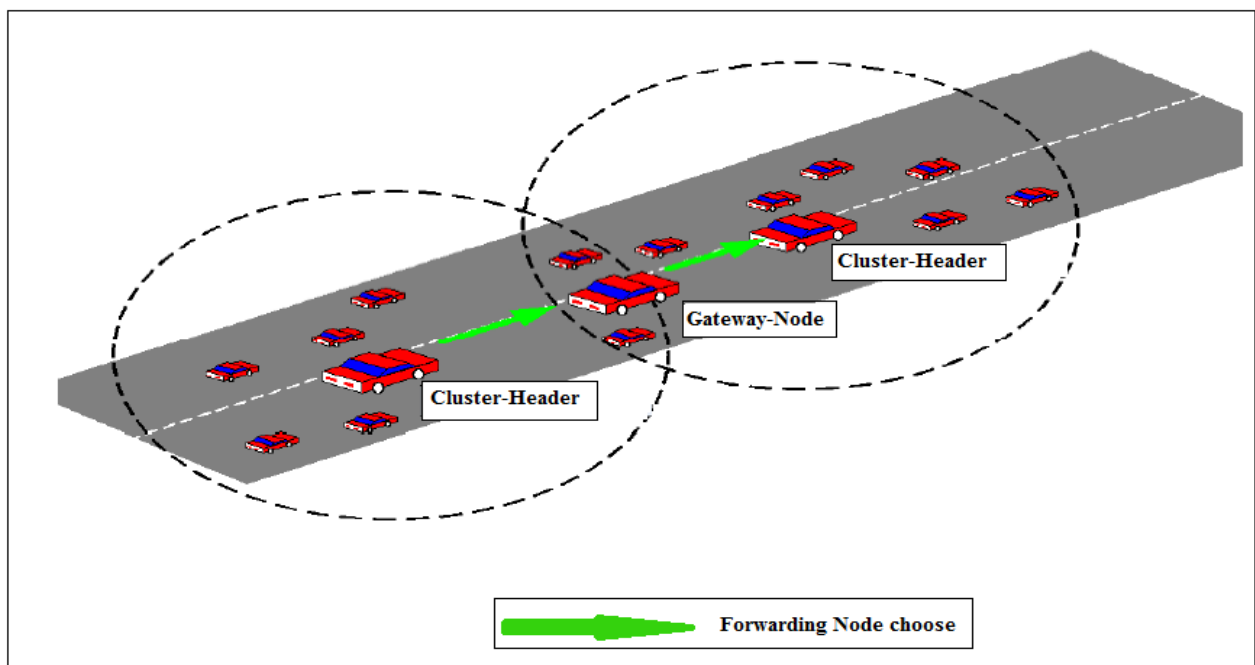


Figure II.9 : Réseaux véhiculaire basés sur les groupes.

II.4.3.1. Le protocole CBLR (Cluster Based Location Routing) :

Est un protocole de routage réactif basé sur les groupes. Chaque chef de groupe conserve une table de routage contenant les adresses et les localisations géographiques des nœuds de son propre groupe et des nœuds passerelles [34], en plus de maintenir une table de routage des groupes voisins. Lorsqu'une source veut envoyer des données à une destination, le chef de groupe vérifie d'abord si la destination est dans le même groupe ou non. Si celle-ci est dans le même groupe, il envoie le paquet à la plus proche voisine de la destination ; Sinon, la source stocke le paquet de données dans son tampon.

Le CBLR est adapté aux réseaux à haute mobilité, puisqu'il met à jour la localisation de la source et de la destination à chaque fois avant de commencer la transmission de données. [34]

II.4.3.2. Le protocole CBDRP (Cluster-Based Directional Routing Protocol) :

Cluster-Based Directional Routing Protocol est un protocole de routage directionnel basé sur les groupes son principe de fonctionnement est de diviser les véhicules en groupes et les véhicules qui se déplacent dans la même direction sous forme d'un cluster. La source envoie le paquet au chef de son groupe, puis ce dernier transmet le paquet au chef du groupe de destination, qui le transmet à son tour à la destination. Le CBDRP prend en considération la vitesse et la direction du véhicule. [28] [34]

II.4.4. Les protocoles Géo-caste :

La diffusion géographique est une technique de routage de paquets qui a pour but de transmettre des données à l'ensemble des nœuds situés dans une zone géographique donnée appelée zone d'intérêt (ZOR Zone Of Relevance)[20]. Par conséquent un nœud doit juste rentrer dans cette zone pour pouvoir recevoir des paquets.

II.4.4.1. Le protocole IVG (Inter-Vehicle Geocast) :

C'est un protocole qui optimise la dissémination d'information, limite le nombre de sauts qu'un message peut effectuer [8] en sélectionnant dynamiquement les nœuds relais pour rediffuser périodiquement les messages d'alerte. Ainsi l'objectif du protocole IVG est d'informer les véhicules (appelés groupe multicast) situés dans une région à risque à propos d'un danger survenu sur une autoroute.

Selon leur localisation par rapport à la zone à risque, les voisins recevant ce message calculent un intervalle de différence de temps (Differ Time Backoff) de réception d'une réponse de ses voisins à un saut qui permettra au voisin le plus loin d'être un relais pour la transmission selon le principe le plus loin est le plus favorable [20].

X est le voisin le plus loin de Z par rapport à Y, il permet d'atteindre W. La figure II.10 illustre la sélection de relais dans IVG.

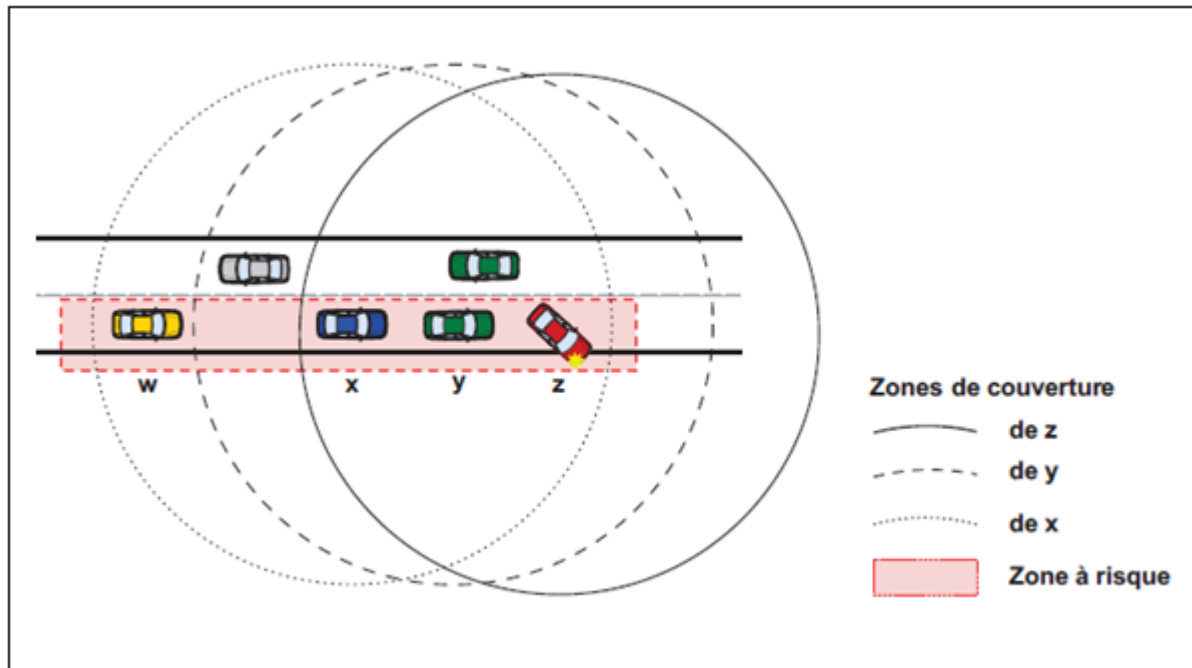


Figure II.10 : Sélection de relais dans IVG.

II.4.4.2. Le protocole DRG (Distributed Robust Geocast) :

Le protocole DRG se base sur la technique du Greedy-Forwarding utilisant le principe le plus loin des voisins d'un véhicule source et plus proche de la destination est le nœud relais le plus favorable.

Dans DRG , le message géocast est relayé de la source vers le destinataire à travers une zone appelée (Zone Of Relevance ZOR) qui représente l'ensembles des nœuds éligibles à relayer les messages géocasts puis délivrés à tous les nœuds appartenant à une zone d'intérêt ZOR définie comme étant l'ensembles des nœuds qui partagent les mêmes critères géographiques. [20]

II.4.5. Les protocoles Broad-caste :

La diffusion de données consiste à envoyer l'information à tous les nœuds qui sont à portée de transmission et utilise le plus souvent la méthode d'inondation («flooding» en anglais). Un message envoyé par un véhicule émetteur par diffusion est transmis à tous ces voisins directs puis est retransmis encore une fois par chacun de ces récepteurs jusqu'à atteindre le nœud destinataire.[1] L'inondation est efficace dans des réseaux de petite taille, mais mène rapidement à une surcharge du réseau lorsque celui-ci est grand.

II.4.5.1. Le protocole UMB (Urban Multihop Broadcast Protocol) :

C'est un protocole basé sur une diffusion multi saut pour les réseaux inter véhiculaires avec support d'infrastructure, dans le but de réduire les collisions et d'utiliser efficacement la bande passante. [35] [33]

UMB charge les nœuds les plus éloignés de l'envoi de paquets sans connaître à priori des informations sur la topologie du réseau par conséquent la sélection du nœud le plus éloigné est fondée sur la transmission du plus long signal. Cela implique une latence élevée et limite son usage dans les cas d'urgence.

UMB est scindé en deux phases :

La première appelée diffusion directionnelle où le véhicule source sélectionne un nœud dans la direction de diffusion pour faire un relayage de données sans aucune information sur la topologie.

La deuxième diffusion aux intersections pour disséminer les paquets dans toutes les directions, pour cela UMB utilise des répéteurs installés dans les intersections pour l'envoi des paquets vers tous les segments [16] [36]

II.4.5.2. Le protocole DVCAST (Distributed vehicular broadcast protocol):

Le protocole de diffusion distribué pour les véhicules (Distributed Vehicular Broadcast Protocol), utilise le principe de diffusion. Chaque véhicule utilise un drapeau (flag) [32] pour vérifier si le paquet est redondant ou non et il utilise des informations locales de topologie en envoyant des messages périodiques «hello» pour diffuser les informations.

DVCAST est un protocole qui divise les véhicules en trois catégories en fonction de leur connectivité locale : connecté, peu connecté et totalement déconnecté. [33][35][36]

II.5. Discussion :

Dans ce chapitre, nous avons présenté quelques protocoles de routage qui ont été proposés pour assurer le service de routage dans les réseaux mobiles VANETs et nous avons décrit leurs principales caractéristiques et fonctionnalités afin de comprendre les stratégies utilisées dans l'acheminement des données entre les différents nœuds. Le protocole AODV fera le sujet principal dans le prochain chapitre Dans ce cadre, nous allons étudier le protocole AODV et les outils de simulation pour VANET.

III.1. Préambule :

Pour mesurer les performances d'un réseau Ad Hoc de véhicule, et plus particulièrement les performances d'un protocole de routage nous avons recours souvent à la simulation, Elle est nécessaire quand l'expérimentation est trop coûteuse et l'étude théorique trop complexe ou quand on souhaite valider des hypothèses et anticiper des problèmes qui pourront se poser dans le futur. Son objectif est de reproduire les mêmes résultats que ceux que l'on obtiendrait en conditions réelles.

Dans ce chapitre nous allons présenter une étude détaillée du protocole AODV et du simulateur NS2.

III.2. Etude détaillé du protocole de routage AODV :

Le protocole « Routage avec Vecteur de Distance à la Demande » (AODV : Ad hoc On-demand Distance Vector), est destiné aux réseaux ad hoc à nœuds mobiles. C'est un protocole multi hop, adaptatif et dynamique par rapport aux conditions de la topologie, réduit le nombre de diffusion de messages et cela en créant les routes lors du besoin.

Ce protocole utilise trois types de paquets, RREQ (Route Request), RREP (Route Reponse), et RERR (Route Error). [37]

III.2.1. Le mécanisme de fonctionnement du protocole :

A cause de la mobilité des nœuds dans les réseaux ad hoc, les routes changent fréquemment ce qui fait que les routes maintenues par certains nœuds, deviennent invalides. De ce fait, AODV utilise le principe de numéro de séquence à fin de maintenir les routes les plus nouvelles ou autrement dit les plus fraîches. [37]

AODV maintient les chemins d'une façon distribuée en gardant une table de routage, au niveau de chaque nœud de transit appartenant au chemin cherché. Une entrée de la table de routage contient essentiellement :

- L'adresse de la destination.
- Le nœud suivant.
- La distance en nombre de nœud (le nombre de nœud nécessaire pour atteindre la destination).
- Liste des voisins actifs.

- Le numéro de séquence de la destination.
- Un tampon de requête afin qu'une seule réponse soit envoyée par requête [38]
- Le temps d'expiration de l'entrée de la table (temps au bout duquel l'entrée est invalidée)

III.2.2. Génération et acheminement des requêtes RREQ :

Quand une source cherche une route vers une destination, une requête RREQ est envoyée en broadcast à tout le voisinage. Chaque nœud recevant cette requête sauvegarde l'identificateur du nœud à partir duquel la première copie de la requête est reçue et envoie de nouveau cette requête jusqu'à ce que celle-ci atteigne sa destination. [37]

Le champ numéro de séquence destination du paquet RREQ, contient la dernière valeur connue du numéro de séquence, associé au nœud destination. Cette valeur est recopiée de la table de routage.

Si le numéro de séquence n'est pas connu, la valeur nulle sera prise par défaut. Le numéro de séquence source du paquet **RREQ** contient la valeur du numéro de séquence du nœud source.

Comme nous avons déjà dit, après la diffusion du **RREQ**, la source attend le paquet réponse de route **RREP**. [38]

A chaque nouvelle diffusion, le champ *Broadcast ID* du paquet RREQ est incrémenté pour identifier une requête de route particulière associée à une adresse source. Si la requête RREQ est rediffusée un certain nombre de fois (**RREQ_RETRIES**) sans la réception de réponse, un message d'erreur est délivré à l'application.

Les paquets DATA en attente d'une route (en attente d'une RREP après la génération d'une requête RREQ) doivent être sauvegardés dans une file d'attente. Si le **RREQ_RATELIMIT** pour une certaine destination est atteint, tous les paquets de DATA qui lui sont destinés doivent être rejetés.

@ SOURCE	Numéro Seq Source	Broadcast ID	@ Destination	Numéro Seq Destination	Nombre de Sauts
----------	----------------------	--------------	---------------	------------------------------	--------------------

Figure III.1 : Format d'un Message RREQ.

@Source : L'adresse du nœud qui a initié la découverte de route

Numéro de séquence Source : Le numéro de séquence courant associé au nœud source

Broadcast ID : Un Numéro Identifiant une découverte de Route

@ Destination : L'adresse du nœud pour lequel la route est recherchée

Nombre de Sauts : Le Nombre de sauts séparant la source du Nœud traitant le paquet RREQ

III.2.3. Génération et acheminement des réponses RREP :

Un nœud génère une RREP s'il est la destination, ou dispose d'une route active vers la destination de la requête ; l'envoi de cette réponse se fait en mode unicast vers la source en suivant le chemin inverse de la requête RREQ.[38]

Les nœuds appartenant au chemin de retour vont modifier leurs tables de routage suivant le chemin contenu dans le paquet de réponse (temps d'expiration, numéro de séquence et prochain saut). [3] une fois que la source a reçu RREP, des paquets de données peuvent être transmis à la destination.

@ Source	@ Destination	Numéro de séquence Destination	Nombre de sauts	Life time
----------	---------------	--------------------------------------	-----------------	-----------

Figure III.2 : Format d'un message RREP.

@Source : L'adresse du nœud qui a généré RREP

@Destination : L'adresse du nœud qui a initié la découverte de route

Numéro de séquence Destination : Le numéro de séquence du nœud destination

Nombre de Sauts : Le nombre séparant la destination du nœud traitant RREQ

Life Time : Un temps en Milliseconde pour lequel le Nœud recevant le RREQ considère la route comme étant valide.

III.2.4. Génération et acheminement des erreurs RERR :

Les messages d'erreurs sont générés si un nœud détecte un lien défaillant sur une route, il informe les nœuds de la route via un message RERR (Route Error).

Ainsi, quand un nœud détecte une erreur de transmission, un paquet route-error (erreur de route) contenant l'adresse du nœud qui a détecté l'erreur et celle du nœud qui le suit dans le chemin, est envoyé à l'émetteur original du paquet. Lors de la réception de ce paquet par la source, le nœud concerné par l'erreur est supprimé du chemin sauvegardé, et tous les chemins qui contiennent ce nœud sont tronqués à ce point-là. [39]

III.2.5. Maintenance des routes :

Une mise à jour de route est effectuée, notamment pour détecter la rupture de liens. Un mécanisme simple de messages HELLO périodiques est mis en œuvre.

Afin de maintenir des routes consistantes, Ces messages sont envoyés à tous les voisins par un nœud pour signaler son existence. Si trois messages « **HELLO** » ne sont pas reçus consécutivement, le lien en question est considéré défaillant.

Quand un lien, reliant un nœud **X** avec le nœud qui le suit dans le chemin de routage, devient défaillant, le nœud **X** diffuse un paquet **UNSOLICITED RREP**, avec une valeur de numéro de séquence égale à l'ancienne valeur du paquet RREP incrémentée, et une valeur infinie pour la distance. Le paquet UNSOLICITED RREP est diffusé aux voisins actifs, jusqu'à ce qu'il arrive à la source. Une fois le paquet est reçu, la source peut initier le processus de la découverte de routes. [38]

III.2.6 Gestion des numéros de séquence :

Chaque nœud possède donc son propre numéro de séquence permettant de dater les informations. Un numéro de séquence est incrémenté dans les cas suivants : [40]

- Avant de commencer une découverte de route, un nœud incrémente son numéro de séquence
- Avant d'envoyer une réponse RREP, le nœud met à jour son numéro de séquence.

III.2.7. Avantages d'AODV :

- ✓ Avec la mobilité, l'AODV entreprend une nouvelle découverte de route à chaque fois qu'il la perd.
- ✓ Le protocole AODV ne se sature pas lorsque le nombre de sauts dans une route dépasse une certaine limite.
- ✓ AODV propose des routes constamment à jour.
- ✓ Un faible délai de transmission des paquets de données.
- ✓ Le taux de réussite des paquets envoyés est élevé.

III.2.8. Inconvénients d'AODV :

- ✓ Une lenteur à cause de la recherche des chemins..
- ✓ L'impossibilité de connaître au préalable la qualité du chemin (en termes de bande passante, délais,... etc.). [38]

III.2.9. Métriques de performances :

Les métriques sont des paramètres de test du protocole de routage qui permettent de mesurer les performances de celui-ci.

❖ Délai de sélection de route :

Il permet de renseigner sur le temps mis par le protocole pour trouver une route. C'est la différence entre l'instant où l'initiateur d'une connexion reçoit la confirmation de route et l'instant où il avait émis la requête de route correspondante. [39]

❖ Trafic de contrôle :

Ce paramètre nous informe sur la quantité des messages (paquets) de contrôle générés par le protocole AODV pour la recherche, l'établissement et le maintien des routes.

❖ Délai de bout en bout :

C'est le temps nécessaire pour acheminer un paquet de donnée depuis le nœud source jusqu'au nœud destination. Il inclut le temps de latence pour la découverte des routes, le temps de passage dans les files d'attente des nœuds intermédiaires et le temps de transmission

d'un nœud vers un autre. C'est le paramètre principal que l'on vise à améliorer lors de l'évaluation d'un protocole de routage. Un bon protocole a des délais moyens de bout en bout qui sont les plus bas possible.

❖ **Débit de transmission :**

Le débit représente le taux de transfert de paquets avec succès sur un canal de communication à un moment donné, il est exprimé en bits par seconde.

❖ **Taux de perte :**

C'est le nombre de paquets perdu par rapport au nombre de paquets émis.

❖ **La gigue :**

C'est la variation de l'intervalle de temps entre deux paquets durant leur acheminement entre la source et la destination.

III.3. Les outils de simulations :

Lors du développement d'un réseau Ad Hoc de véhicules pour une application réelle, la simulation par logiciel de son comportement et de ses performances est essentielle ; elle a plus de crédibilité que les modèles analytiques car elle est plus proche du système réel.

La simulation des VANETs fait appel à deux types de simulateurs à savoir le simulateur de trafic et le simulateur de réseaux. [13]

Il existe plusieurs simulateurs pour les VANETs dont les principaux sont :

➤ **Le simulateur réseaux NS2 :**

NS-2 est un outil logiciel de simulation libre à code source ouvert et à événements discrets permettant l'étude, la conception et la gestion des protocoles pour les réseaux informatiques, très largement utilisé dans la recherche académique et dans l'industrie. Il a été développé à partir de méthodes de conception orientées objets dans le projet VINT (Virtual Inter Network Testbed).

C'est un simulateur réseaux qui permet de gérer les communications entre les nœuds, il permet également d'évaluer les protocoles de communication dans un scénario donné selon les paramètres définis par l'utilisateur.

L'ouverture du code source à la communauté a contribué à l'enrichir de nouveaux protocoles et de nouvelles fonctions au fil du temps. Il permet de définir un réseau et de simuler des communications entre les nœuds de ce réseau.

➤ **Le simulateur de trafic routier SUMO :**

SUMO (Simulator of Urban Mobility) est un logiciel de simulations de trafic routier open source. Il permet de traiter tout ce qui est en rapport avec le mouvement des véhicules sur un réseau routier donné, il offre souvent tous les outils nécessaires pour la réalisation des cartes routières (spécification des lignes d'une voie de circulation et sa direction, ajout d'un rond-point ...etc.) ; et il permet de gérer la mobilité des véhicules (spécification des caractéristiques des véhicules, générer des accidents...etc.).[20]

➤ **OMNet++:**

OMNET++ (Objectif Modular Network Test-Bed) [12] est un simulateur open source, orienté objet et à événements discrets écrit en C++ qui peut simuler n'importe quel type de réseau de grande topologie. son domaine d'application principal est celui des réseaux de communication.

Il peut être utilisé dans divers domaines tel que : modélisation des réseaux de communication, modélisation des protocoles, Réseaux de files d'attente, multiprocesseurs et autres matériels distribués. [41]

➤ **Veines :**

(Vehicles in Network Simulator) est un simulateur open source Inter-Véhicules communication (IVC), composé d'un simulateur de réseau basé sur les événements et un modèle de micro-traffic de la route.

Veines est composé de deux simulateurs distincts, OMNeT++ pour la simulation de réseau et SUMO pour la simulation du trafic routier. [41]

➤ **Le simulateur NCTUns :**

NCTUns (National Chiao Tung University Network Simulator) est un simulateur et émulateur réseau extensible à haute fidélité capable de simuler différents protocoles utilisés

dans les réseaux IP filaires et sans fil. Initialement, NCTUns a été développé comme un simulateur réseau, mais la version la plus récente intègre des fonctionnalités pour la modélisation du mouvement des véhicules, le changement de voies, le dépassement et la gestion des intersections. [41]

III.4. Le choix de simulateur:

Le choix de simulateur pour notre projet s'est porté sur le NS2 car c'est le simulateur le plus documenté et le plus polyvalent et son utilisation est gratuite. Son langage de programmation est facile pour décrire les scénarios. Le protocole AODV dont on va évaluer les performances est implémenté dans ce simulateur.

Pour la plateforme, notre choix s'est porté sur la plateforme de Windows via l'émulateur Cygwin. Nous avons travaillé avec ns-allinone-2.35RC7 sous Windows7.

III.4.1. Le simulateur du réseau NS2

III.4.1.1. Présentation :

Le simulateur du réseau NS2 est un outil logiciel de simulation de réseaux informatiques. Il est principalement bâti avec les idées de la conception par objets, de réutilisation du code et de modularité. Ce logiciel est exécutable sur les deux plateformes Unix et Windows.

Sous Windows, on doit faire appel à l'émulateur Cygwin pour que les logiciels fonctionnant sous linux puissent fonctionner sous Windows.

Il est bâti autour d'un langage de programmation appelé Tcl dont il est une extension. il permet de gérer les communications entre les nœuds et évaluer les protocoles de communication dans un scénario donné. Du point de vue de l'utilisateur, la mise en œuvre de ce simulateur se fait via une étape de programmation qui décrit la topologie du réseau et le comportement de ses composants, puis vient l'étape de simulation proprement dite et enfin l'interprétation des résultats. Cette dernière étape peut être prise en charge par un autre outil appelé Nam qui permet une visualisation et une analyse des éléments simulés.

NS2 est en réalité un programme relativement complexe écrit en C++ avec une interface qui utilise le langage OTcl (Object Tool Command Langage). Pour modifier le comportement d'objets existants, il est donc nécessaire de modifier le code C++ qui en réalise l'implantation. [42]

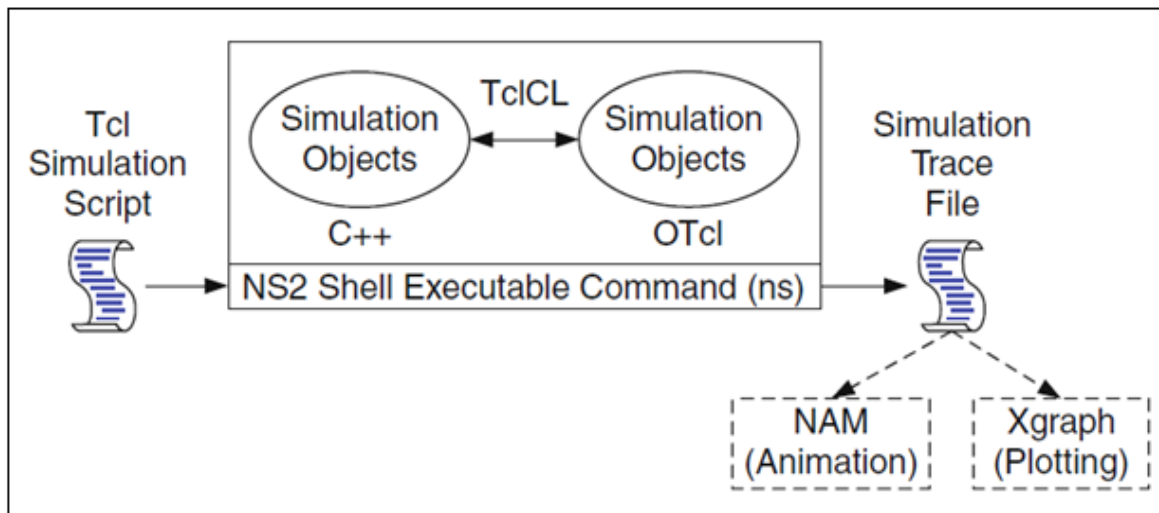


Figure III.3: Architecture basic de NS2.

❖ **Le langage C++ :**

Le langage C++ dans NS2 sert à décrire le fonctionnement interne des composants de la simulation, un grand nombre de classes de base sont prédéfinies et mettent en œuvre plusieurs types de protocoles, de files d'attentes, de sources et algorithmes de routage.

❖ **Le langage TCL (Tool Command Language) :**

TCL (Tool Command Language) est un langage de commande qui sert à contrôler les applications. C'est essentiellement un langage de scripts offrant des structures de programmation telles que les boucles, les procédures ou les notions de variables.

❖ **Le OTcl (Object Tool Command Language) :**

OTcl est une extension orientée objet de Tcl. Les commandes Tcl sont appelées pour un objet. En OTcl, les classes sont également des objets avec des possibilités d'héritage. La définition d'une classe commence par la directive **Class**. Les fonctions et les attributs d'une classe s'installent par la suite par les commandes **instvar** et **insproc**. L'utilisation **instproc** définit les méthodes de la classe de manière assez semblable à C++.

A travers OTCL, l'utilisateur décrit les conditions de la simulation à savoir la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés, le type de trafic généré par les sources et les communications qui ont eu lieu.

NS2 interprète le script OTCL et exécute la simulation. Les résultats obtenues (Fichier trace et fichier NAM) peuvent être visualisés avec l'outil NAM (Network Animator) et analysés à partir des courbes tracées par l'outil Xgraph (Un traceur de graphes). [40]
La figure suivante illustre les résultats de simulation avec NS2. [40]

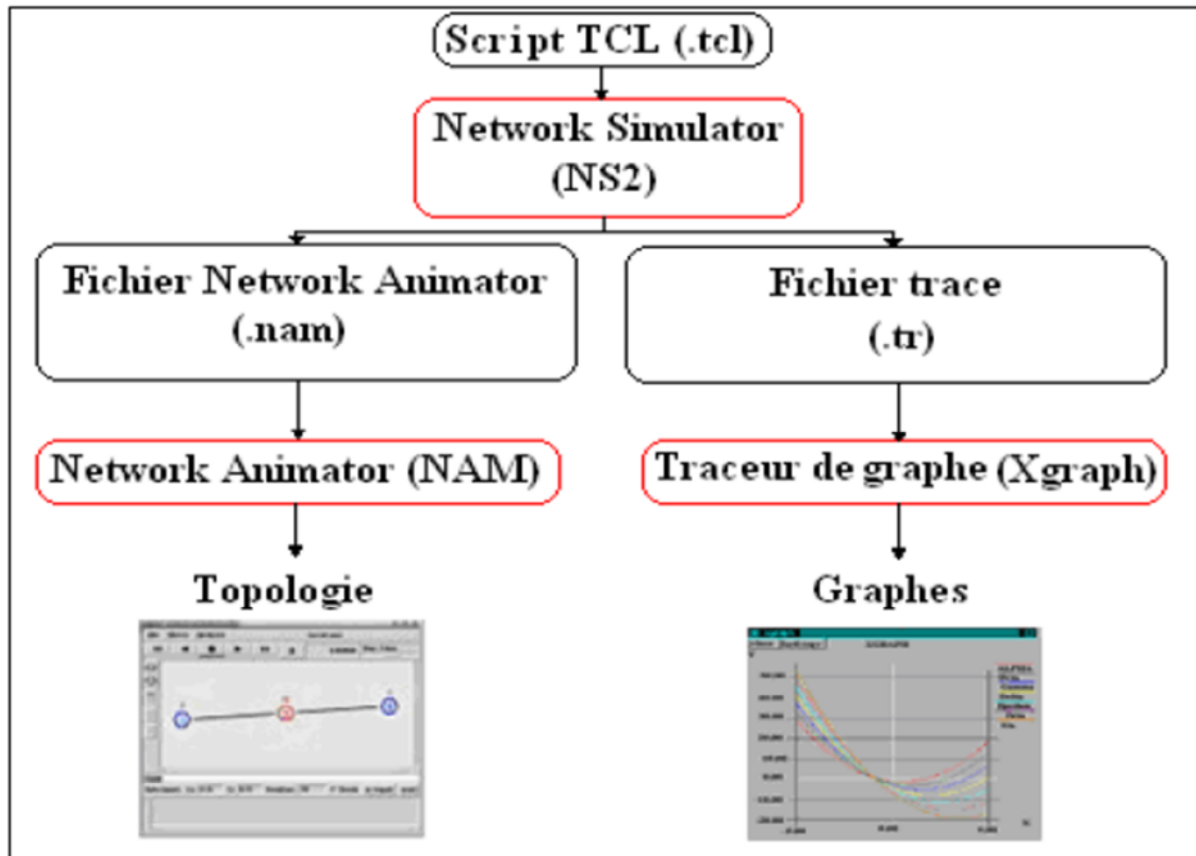


Figure III.4 : Flot de simulation avec NS2.

III.4.1.2.L'outil de visualisation (NAM) :

NS2 offre un outil de visualisation NAM (Network Animator) ; qui permet le suivi des états d'un grand nombre de nœuds, une analyse de l'échange de messages et des interactions dynamiques pour des trafics concurrents.

NAM (Network Animator) est un outil d'animation basé sur Tcl/TK permettant de visualiser le mouvement des paquets. Il est utilisé dans NS pour interpréter et visualiser le tracé de simulation des réseaux, le tracé de données.

La fenêtre d'animation NAM démarre avec la commande « **nam "nom du fichier nam" (.nam)** ».

Deux éléments intéressants sont proposés à la visualisation : un dessin de la topologie du réseau étudié, et une visualisation dynamique du déroulement du programme dans le temps.

- Les grands cercles représentent les paquets diffusés afin de prendre connaissance du réseau.
- Les cercles numérotés représentent les nœuds mobiles.
- Les petits points représentent les paquets de données de transmission.

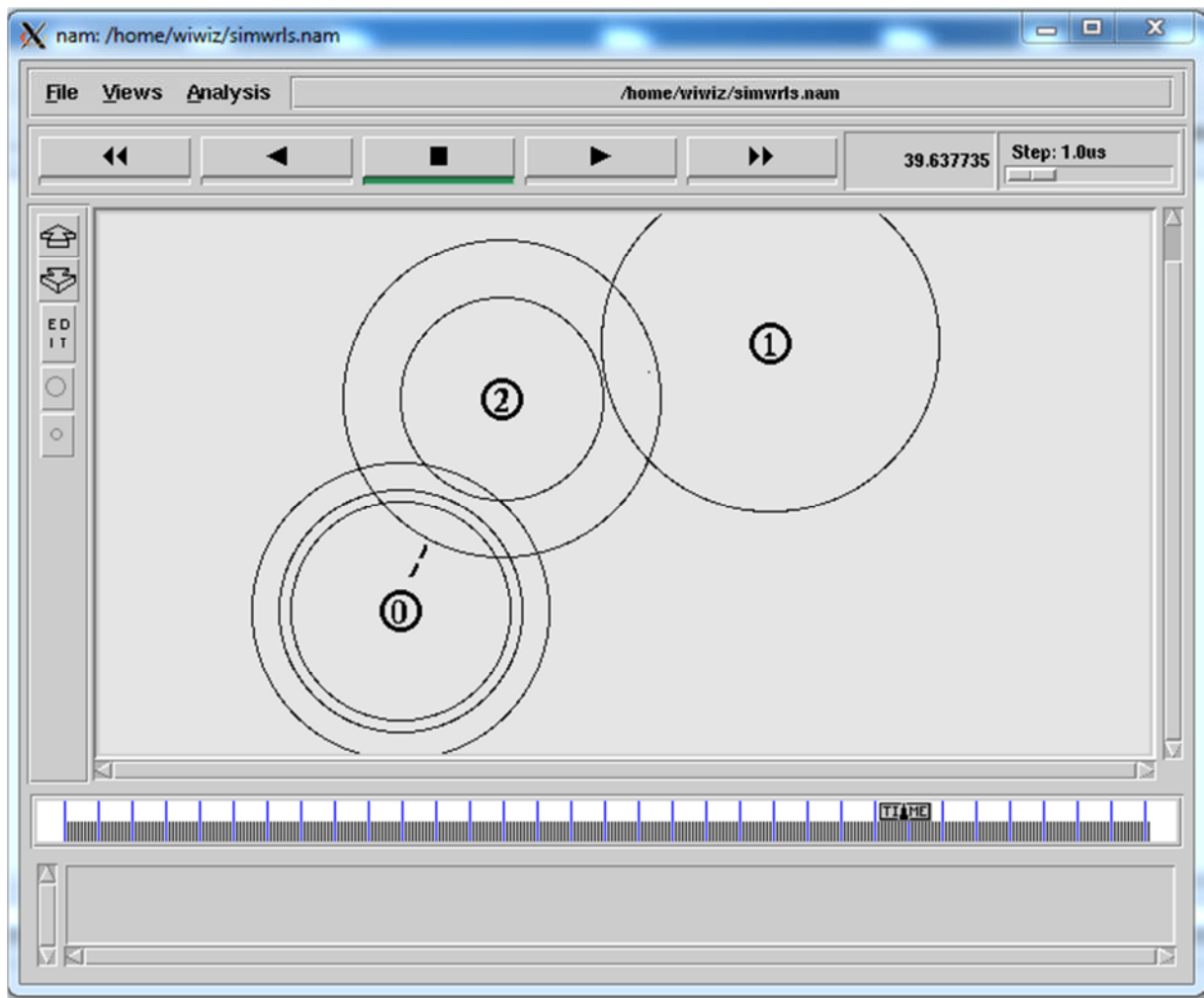


Figure III.5 : Visualisation de la topologie et le déroulement de trafic avec Nam.

III.4.1.3. Le Fichier trace (résultat de la simulation : out.tr) :

Le fichier trace est un fichier qui retrace tous les évènements qui se sont produits pendant la simulation à savoir émission, réception, perte de paquets. La sortie du traceur est indispensable pour l'étude des statistiques, afin de dessiner les courbes en utilisant Xgraph.

Ils existent deux formats de fichier dans les réseaux Ad hoc, l'ancien format et le nouveau format.

Dans l'ancien format, le fichier est vu comme étant un ensemble de colonne dont le sens de chacun est prédéfini. Le tableau III.1 illustre un exemple détaillé d'une ligne d'un fichier trace ancien format sous NS-2.

1	2	3	4	5	6	7	8	9	10	11	12
Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	@ Src	@ Dst	Seq num	Pkt ID
r	0.018	2	0	cbr	1000	-----	0	2.0	1.2	0	3
+	0.018	0	1	cbr	1000	-----	0	2.0	1.2	0	3
-	0.018	0	1	cbr	1000	-----	0	2.0	1.2	0	3
s	0.015	2	1	ftp	2000	-----	1	2.1	2.0	1	2

Tableau III.1 : Le format du fichier trace ancien format.

1 : Action effectuée sur le paquet.

Un "+" signifie que le paquet est reçu dans une file.

Un "-" signifie que le paquet quitte la file.

Un "s" signifie que le paquet est envoyé.

Un "d" signifie que le paquet est jeté (dropé).

Un "r" signifie que le paquet est réceptionné par un agent.

2 : Instant où l'action est effectuée.

3 : Nœud de départ du lien concerné.

4 : Nœud d'arrivée du lien concerné.

5 : Type de paquet.

6 : Taille du paquet en bytes.

7 : Flags.

8 : Identificateur de flux.

9 : Agent de départ.

10 : Agent d'arrivée.

11 : Numéro de séquence.

12 : Identificateur unique pour chaque paquet.

Le nouveau format est constitué d'un ensemble de tag pour chaque événement, chaque tag porte le sens de l'information qui le suit, nous pouvons voir ce format comme étant un ensemble de champs. [42]

La figure III.6 montre un exemple de deux lignes de fichier trace de protocole AODV.

```
r 1.115349444 _21_ RTR -- 0 AODV 48 [0 fffffff f 800] ----- [15:255 -1:255 26 0] [0x2 5 1 [21 0] [24 4]] (REQUEST)
s 1.115594280 _14_ MAC -- 0 AODV 102 [13a d e 800] ----- [21:255 24:255 29 13] [0x4 2 [21 4] 10.000000] (REPLY)
```

Figure III.6 : Exemple de deux lignes de fichier trace du protocole AODV.

Champ [1] : le premier champ, est une lettre, permet d'identifier le type d'événement qui s'est produit. Ce champ peut être :

r (pour *receive*) indiquant une réception de paquet.

s (pour *send*) indiquant un envoie de paquet.

d (pour *drop*) indiquant une suppression de paquet.

f (*forware*) pour un acheminement.

Champ [2] : « 1.115349444 » indique le temps de simulation, le temps de production de l'évènement.

Champ [3] : « 21 » indique le numéro de nœud.

Champ [4]: « RTR »'il s'agit du paquet routé.

Champ [5] : « 0 » le numéro de séquence globale du paquet.

Champ [6] : « AODV» information sur le type de paquet.

Champ [7] : « 48 » la taille du paquet en octet.

Champ [8] = [0 fffffff f 800] : « 0 » La durée du paquet dans l'en-tête de la couche Mac
« fffffff » l'adresse mac de destination.

« f » l'adresse MAC de la source.

« 800 » le type mac du corps du paquet.

Champ [9]= [15 :255 -1 :255 26 0] : Informations de trace IP

L'adresse IP source et destination de « 15 » et « 1 », respectivement.

Les ports pour la source et la destination sont « 255 ».

Le ttl (Time To Live) du paquet et l'adresse du prochain nœud de saut sont respectivement «26» et «0».

Champ [10] = [0x2 5 1 [21 0] [24 4]] : Il indique les informations de trace AODV

Il s'agit d'un paquet RREQ marqué avec l'ID « 0x2 ».

Le nombre de dénombrements de sauts est « 1 » et l'ID de diffusion est « 5 ».

L'adresse IP de destination et le numéro de séquence sont respectivement « 21 » et « 0 ».

L'adresse IP source et le numéro de séquence sont respectivement de « 24 » et « 4 ».

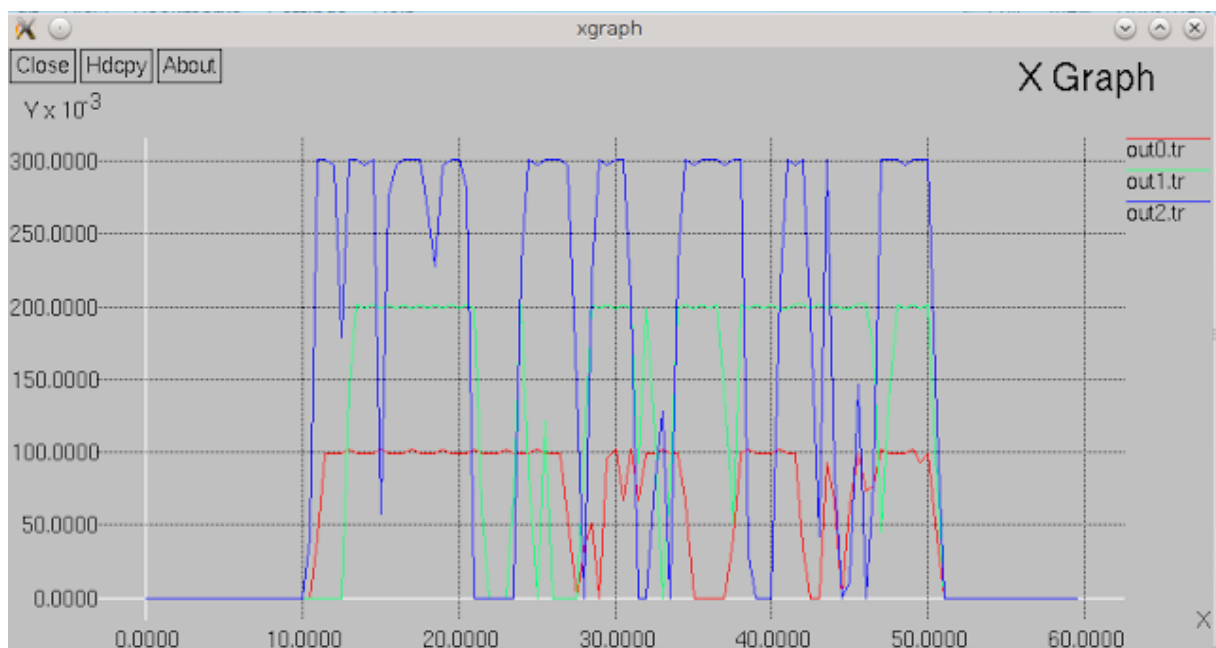
La chaîne "(REQUEST)" confirme qu'il s'agit du paquet RREQ.

Le format que nous venons de présenter varie réellement en fonction des protocoles utilisés pour la simulation.

III.4.1.4. Les outils d'analyse de fichier trace :

❖ Xgraph :

Xgraph est un programme de traçage qui peut être utilisé pour créer des représentations graphiques des résultats de simulation.



FigureIII.7 : Exemple d'un graph sous NS2.

❖ NS2 Visual Trace Analyzer :

Le simulateur de réseau NS2 est un logiciel de simulation le plus utilisé dans le monde, mais il manque d'outils pour analyser ses fichiers de trace.

NS2 Visual Trace Analyzer fournit un moyen simple de remplir cette tâche exhaustive permettant de tracer des graphiques, des paquets de filtrage, visualiser la position des nœuds, calculer les statistiques de nœud et de trafic, etc. Cette application autonome, avec une

interface conviviale, sans nécessité d'installation et sans exigences de bibliothèques externes, répond à la plupart des besoins des utilisateurs.

❖ AWK :

AWK est un langage pour le traitement des fichiers de texte, inspirée du langage C. Chaque fichier est traité comme une séquence d'enregistrements, et par défaut, chaque ligne est un enregistrement. Il décrit comment filtrer le fichier de sortie « .tr » de NS2 pour extraire des données significatives de la simulation.

III.5. Les composants d'un réseau sous NS2 : Un modèle de réseau sous NS est constitué de :

- **Les nœuds :**

Un nœud est une entité adressable. Il existe deux types de nœuds dans NS, des nœuds unicast et des nœuds multicast. La fonction d'un nœud est de recevoir des paquets, les examiner et les mapper à ses interfaces sortantes appropriées.

- **Les liens de communication entre les nœuds :**

Ils servent à raccorder les nœuds entre eux, il modélise le système de transmission. Le lien est principalement caractérisé par un délai de propagation et une bande passante. Dans la simulation de réseau Ad hoc, les liens ne sont pas définis explicitement comme reliant deux nœuds, ce qui est tout naturellement dû à la nature, sans fil, du réseau. De ce fait, les informations concernant les spécificités du canal sont fournies en guise de configuration du canal avant la création des nœuds.

- **Agents de communication :**

Les agents sont les points terminaux du réseau qui reçoivent ou délivrent les paquets à des applications, ils sont attachés aux nœuds et connectés les uns aux autres, afin d'échanger des données entre les nœuds.

A chaque agent est attribué un port. L'adresse d'un agent se compose du numéro de son nœud et son port. Actuellement il existe de nombreux agents dans NS2 chacun à un rôle spécifique :

- ✓ Agent TCP : pour émettre un trafic TCP.
- ✓ Agent UDP : pour émettre un trafic UDP.

- ✓ Agent TCPSink : pour la réception du trafic TCP.
- ✓ Agent NULL : pour la réception des paquets UDP.

- **La gestion de la file d'attente :**

La gestion de la file d'attente sur les liens est implémentée dans la classe Queue ; les files d'attente actuellement disponibles dans NS sont [42] :

- Droptail qui utilise la technique FIFO c'est la file d'attente par défaut.
- RED (Random Early Detection).
- FQ (Fair Queuing)
- SFQ (Stochastic FQ)
- CBQ (Class Based Queuing)

- **Application :**

Génère le trafic de données selon certaines applications (CBR, FTP), et se sert des agents de transport. Le tableau suivant englobe les principaux composants de NS2.

Application	Web, FTP, Telnet, générateur de trafic(CBR, ..)
Transport	TCP, UDP, RTP, SRM
Routage	Statique, dynamique (vecteur distance) et routage multipoint (DVMRP, PIM)
Gestion des files d'attente	RED, Drop Tail, Token bucket
Discipline de service	CBQ, SFQ, DRR, Fair queueing
Système de transmission	CSMA/CD, CSMA/CA, lien point à point

Tableau III.2 : Liste des composants disponibles dans NS.

III.6. Les différents modèles de propagation radio sous NS2 :

NS2 permet également de choisir parmi les modèles de propagation suivants, qui influenceront en particulier sur la manière dont seront atténués les signaux en fonction de la distance et les différents obstacles que ce soit les bâtiments qui annulent ou reflètent le signal,

ou les arbres qui se contentent de le réduire et aussi le sol qui réfléchit les ondes électromagnétiques.

Trois modèles de propagation Free space, Two-ray ground reflection et Shadowing sont implémentés dans le simulateur. [8]

III.6.1. Le modèle de propagation en espace libre (Free space model):

Free space model (modèle de l'espace libre) est le modèle le plus simple. Il considère que la transmission se fait de façon directe sans aucun obstacle, réflexions ou effets perturbateurs. Seule la distance entre l'émetteur et le transmetteur est prise en compte [8]

III.6.2. Two-ray ground reflection model (modèle à deux rayons, directs et par le sol) :

Ce model considère deux rayons, le rayon de transmission direct ainsi que le rayon qui se réfléchi sur le sol comme le montre la Figure III.8 ; à condition que les antennes soient séparées d'une distance minimale appelée distance de croisement.

Ce modèle donne des résultats plus justes que le modèle de propagation en espace libre quand la distance est assez grande.

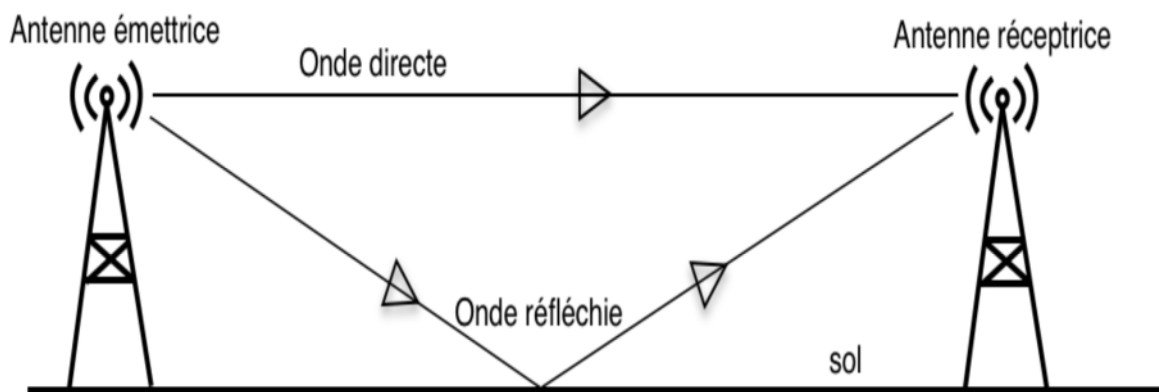


Figure III.8 :Modèle de propagation Two-ray ground.

III.6.3. Le modèle Shadowing :

Ce modèle est le plus proche de la réalité puisque plusieurs phénomènes de la propagation y sont pris en considération, à savoir, la réflexion, la diffusion et l'absorption. En plus le rayon de communication n'est plus concéderez comme un disque parfait.

III.7. Modèles de mobilité sous NS2 : NS2 implémente deux modèles de mobilités

III.7.1. Le Random Waypoint (RWP) :

Ce modèle génère un mouvement aléatoire du nœud de façon que le nœud choisie sa prochaine destination d'une manière aléatoire en se déplaçant avec une vitesse aléatoire constante. Le comportement est répété pour la durée de la simulation.

III.7.2. Le Trajectory Based Mobility Model:

C'est un mouvement généré par un scénario qui consiste à ce que l'utilisateur donne une destination bien précise et une vitesse de déplacement constante.

III.8. Le processus de simulation dans NS2 :

Le processus de simulation en utilisant NS-2 est composé de trois phases principales :

➤ Phase de préparation :

S'occupe de la génération des fichiers d'entrées. A cette étape, on introduit des fichiers de scripts Otcl qui décrivent l'environnement avec tous ses nœuds, leur déplacement et leur trafic de données. Ces fichiers sont classés en deux catégories : [43]

- Fichiers de scénario qui décrivent les nœuds, leurs positions ainsi que leurs mouvements.
- Fichiers de communication qui décrivent le trafic dans le réseau.

➤ Phase de simulation :

Pour lancer les simulations et générer les traces. Les deux fichiers obtenus de la phase de préparation sont introduits dans un script de lancement OTcl. Le script de simulation consiste à indiquer la topologie du réseau, à activer des traces aux endroits pertinents, à engendrer des événements particuliers à des instants donnés. A la fin de cette étape on obtient deux fichiers traces. Le premier fichier sera traité par l'outil de visualisation NAM. Et le deuxième doit être filtré afin d'afficher le résultat en utilisant l'outil Xgraph.

➤ Phase d'analyse :

Pour analyser les traces et générer les courbes. L'outil de visualisation NAM s'occupe du premier fichier trace. Deux éléments intéressants sont proposés à la visualisation : un dessin de la topologie du réseau étudié, et une visualisation dynamique du déroulement du programme dans le temps. Le deuxième fichier de trace sauvegarde tous les échanges de paquets effectués afin de dessiner les courbes en utilisant Xgraph. [43]

III.9. Création d'un scénario dans NS2 :

Pour décrire un réseau et son trafic, il faut définir :

1. la topologie du réseau : les nœuds.
2. la couche transport (UDP, TCP, ...) entre les nœuds
3. la couche application qui va fournir les données.
4. des temporisateurs précisant les instants auxquels les transferts vont démarrer.

- ✓ On déclare au début du script tous les paramètres de configuration des nœuds (type de canal, propagation, file d'attente, protocole de routage...) et aussi les paramètres de la topologie et de la scène (nombre de nœuds, dimension de la topographie...). On fera appel à ces paramètres plus tard pour la création des nœuds

```
set val(chan)           Channel/WirelessChannel      ;# channel type
set val(prop)           Propagation/TwoRayGround    ;# radio-propagation model
set val(netif)          Phy/WirelessPhy            ;# network interface type
set val(mac)            Mac/802_11                 ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue     ;# interface queue type
set val(ll)            LL                           ;# link layer type
set val(ant)            Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)         50                          ;# max packet in ifq
set val(nn)            9                           ;# number of mobilenodes
set val(rp)            AODV                         ;# routing protocol
set val(x)             1000                         ;# X dimension of the topography
set val(y)             800                         ;# Y dimension of the topography
set val(stop)          35                          ;# time of simulation end
```

- ✓ Dans un fichier otcl, on commence par créer un simulateur

```
set ns [new Simulator]
```

- ✓ Pour conserver l'historique de la simulation (notamment pour la visualiser avec NAM), on peut indiquer les deux fichiers vers lesquels seront dirigés les deux historiques.

```
# *** Initialize Trace file ***  
set tracefd [open trace1.tr w]  
set namtrace [open rp.nam w]
```

- ✓ Ensuite, on crée les nœuds du réseau, il y a deux façons de le faire : avec une boucle ou sans boucle.

```
# Creation des noeuds avec une boucle  
for {set i 0} {$i < $val(nn)} {incr i} {  
    set node_($i) [$ns node]  
    $node_($i) random-motion 0  
}
```

```
#Creation des noeuds sans boucle  
set n0 [$ns_ node]  
set n1 [$ns_ node]  
set n2 [$ns_ node]  
set n3 [$ns_ node]  
set n4 [$ns_ node]
```

- ✓ Configuration des nœuds

```
# configure nodes

$ns node-config -adhocRouting $val(rp) \
  -llType $val(ll) \
  -macType $val(mac) \
  -ifqType $val(ifq) \
  -ifqLen $val(ifqlen) \
  -antType $val(ant) \
  -propType $val(prop) \
  -phyType $val(netif) \
  -channelType $val(chan) \
  -topoInstance $topo \
  -agentTrace ON \
  -routerTrace ON \
  -macTrace ON \
  -movementTrace ON
```

- ✓ Définir la position initiale des nœuds et leurs mouvements.

```
# positions initiales des noeuds

$node_(0) set X_ 500.0
$node_(0) set Y_ 400.0
$node_(0) set Z_ 0.0

##donner la mobilité des noeuds
$ns at 0.0 "$node_(0) setdest 500.0 400.0 40.0"
```

Au moment de 0.0 secondes, le nœud (0) commence à se déplacer à partir de son emplacement actuel vers la position (X=500, Y=400,) à la vitesse de 40m / s.

- ✓ Il faut créer les agents de transport, les attacher aux nœuds et les connecter entre eux.
Le code est par exemple :

```
set agent1 [new Agent/UDP] ;# Creat UDP Agent
set sink [new Agent/LossMonitor] ;# Creat Loss Monitor Sink in order to be able
;#to trace the number obytes received
$ns attach-agent $node_(4) $agent1 ;# Attach Agent to source node
$ns attach-agent $node_(2) $sink ;# Attach Agent to sink node
$ns connect $agent1 $sink ;# Connect the nodes
set app1 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app1 set packetSize_ 512 ;# Set Packet Size to 512 bytes
$app1 set rate_ 200Kb ;# Set CBR rate to 200 Kbits/sec
$app1 attach-agent $agent1 ;# Attach Application to agent
```

- ✓ Il est possible de programmer des évènements qui seront exécutés à des dates précises de la simulation. Par exemple :

```
$ns at 1.70 "$app1 start" ;# Start transmission at time t = 1.7 Sec
$ns at 2.75 "$app1 stop" ;# Stop transmission at time t = 2.75 Sec
$ns at 10.0 "finish" ;# invoquer la procédure finish à t=10.0
```

- ✓ La fin de la simulation se fera par l'invocation d'une procédure chargée du traitement final des statistiques. Exemple :

```
proc finish {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd #fermeture du fichier de trace
    close $namtrace #fermeture du fichier de trace animation
exec nam nm.nam & # ouvrir la fenetre nam
exit 0
}
```

- ✓ En fin pour exécuter le programme il faut taper la commande

```
$ns run
```

III.10.Discussion :

Après avoir étudié le protocole AODV et le simulateur réseaux NS2, il est nécessaire de passer à l'expérimentation pour évaluer l'applicabilité et les performances de ce protocoles dans les réseaux VANETs, dans différents scénarios créés sous ns2 en terme de débit, taux de perte de paquets et le délai de transmission.

Nous présentons dans ce qui suit les résultats de simulations ainsi que leurs interprétations.

IV.1. Préambule :

L'objectif principal de notre application est d'analyser quelques performances du protocole de routage AODV dans les réseaux Ad Hoc de véhicule (VANET), à cet effet deux simulations ont été réalisées dans deux environnements urbains et autoroute. Ensuite, nous allons présenter les paramètres métriques sur lesquels on s'est basé afin d'évaluer les performances de ce protocole.

Nous avons jugé utile de faire une comparaison entre les résultats de simulation obtenus afin de savoir à quel environnement ce protocole est plus adapté. Les simulations ont été réalisées avec le simulateur réseaux NS2 présenté ci-avant.

IV.2. Scenarios et paramètres de simulation :

IV.2.1 Scénario 1 :

Cet environnement est modélisé par une intersection à quatre voies, ce réseau est de taille 1000m x 800m, composé de 8 nœuds mobiles (Véhicules), 1 nœud statique (positionné au milieu). On définit 4 nœuds sources qui vont s'échanger un flux CBR (*Constant Bit Rate*) de taille 512octets, la vitesse de transmission est de 200 Kbps avec 4 nœuds destinataires. La durée de la simulation est de 35 secondes.

Les véhicules se déplacent avec une vitesse qui varie de 30 à 40 km/h (vitesse de déplacement des voitures dans un environnement urbain).

Le tableau III.3 récapitule l'ensemble des paramètres de simulation utilisés dans le scénario 1.

Critère	Valeurs
Protocole MAC	IEEE 802.11
Modèle de propagation radio	Two-ray ground
Portée de communication	195m
Capacité du canal	2 Mbit/s
Taille max des files d'attente	50 paquets
Topologie réseau	1000m x 800m
Nombre de nœuds	9 nœuds (8 mobiles, 1 statique)
Vitesse des véhicules	30 à 40 km/h
Temps de simulation	35s
Couche transport	UDP
Type du trafic généré	CBR
Nombre de connexions	4 connexions
Taille du paquet de données	512 octet
Le paramètre simulé	Le délai de bout en bout le débit de transmission la perte des paquets
Transmissions effectuées	Transmission entre les nœuds : 4 et 2, 1 et 3, 5 et 7, 6 et 8.
Le protocole de routage	AODV
Antenne	Omni Antenna : Omnidirectionnel
Vitesse d'émission	200kbit/s

Tableau IV.1 : Paramètres de simulation dans le scenario 1.

Le protocole IEEE 802.11, est utilisé comme protocole d'accès au medium. Le type de gestion de file d'attente ou queue en chaque nœud est « Drop Tail ». Dans ce type de gestion de files, les paquets venant de différents flots sont traités comme étant de même ordre de priorité (selon le principe FIFO), si le tampon est plein, le dernier paquet qui arrive est supprimé. Le nombre maximum des paquets dans le tampon d'émission est de 50 paquets.

Chapitre IV : Simulations et interprétations des résultats

Chaque véhicule transmet grâce à un lien radio 802.11 de débit égal à 2 Mbit/s avec une portée radio de 195 mètres. Nous avons choisi comme modèle de propagation le "Two-Ray ground reflexion model" qui prend en compte l'onde directe et l'onde réfléchiée par le sol.

Le flux CBR est choisi vu qu'il est simple. Il est caractérisé par un débit constant. Ce type d'application exige des délais stricts. Un flux CBR est généré en fixant la taille du paquet (dans notre cas 512 octets).

Un exemple de disposition visualisé par l'outil d'animation nam est donné dans la figure IV.1.

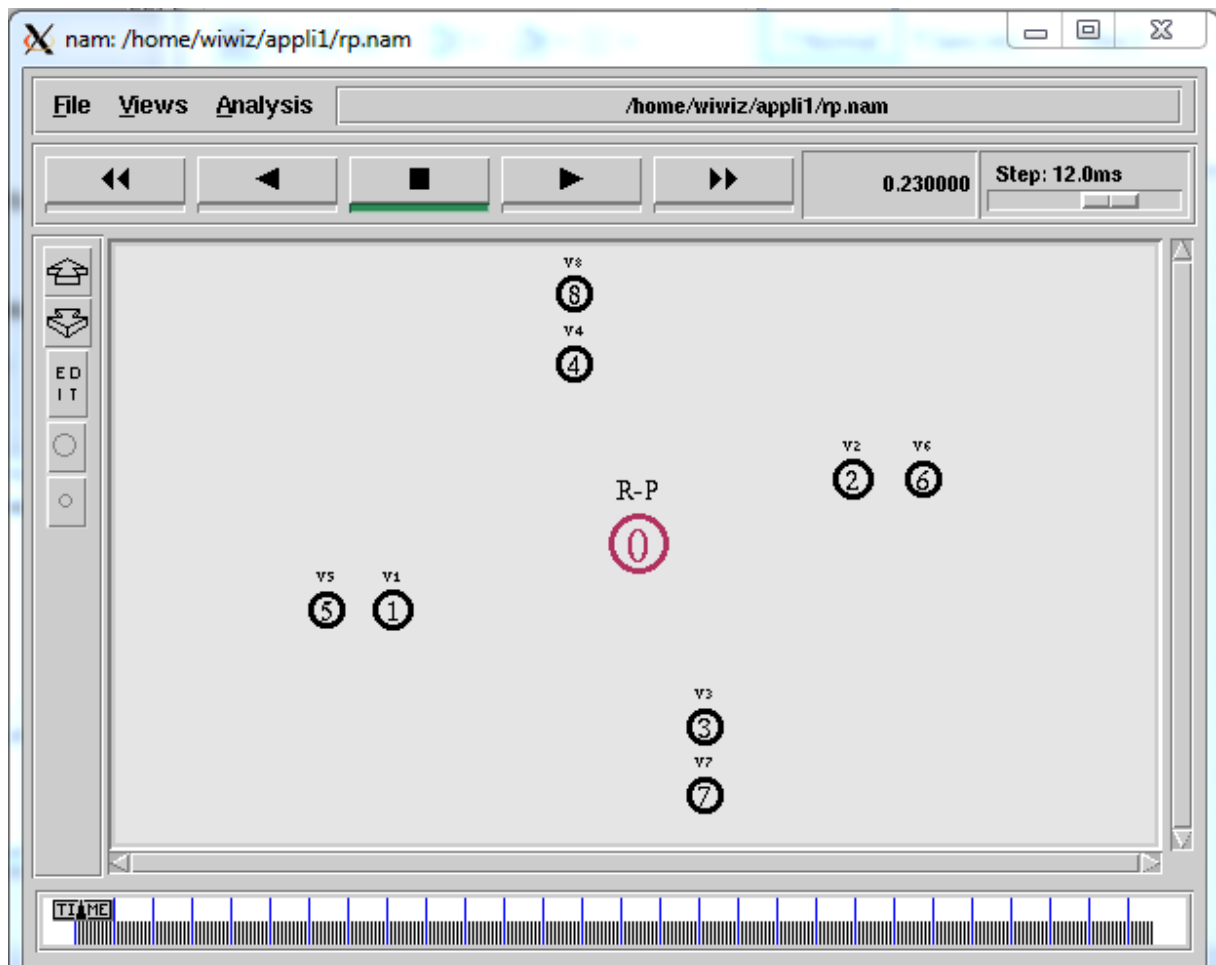


Figure IV.1 : Disposition des nœuds.

IV.2.1.1. Résultats et interprétations de scénario 1 :

Dans notre simulation nous avons pris en compte les trois métriques suivantes :

- ✓ Le débit
- ✓ Le taux de perte des paquets
- ✓ Le délai de bout en bout

Pour chacun de ces graphes, quatre courbes sont visualisées :

— : Correspondant au trafic de l'application1 attaché au nœud source 4 et le nœud destinataire 2.

— : Correspondant au trafic de l'application2 entre les nœuds 1 et 3.

— : Correspondant au trafic de l'application3 entre les nœuds 5 et 7.

— : Correspondant au trafic de l'application4 entre les nœuds 6 et 8.

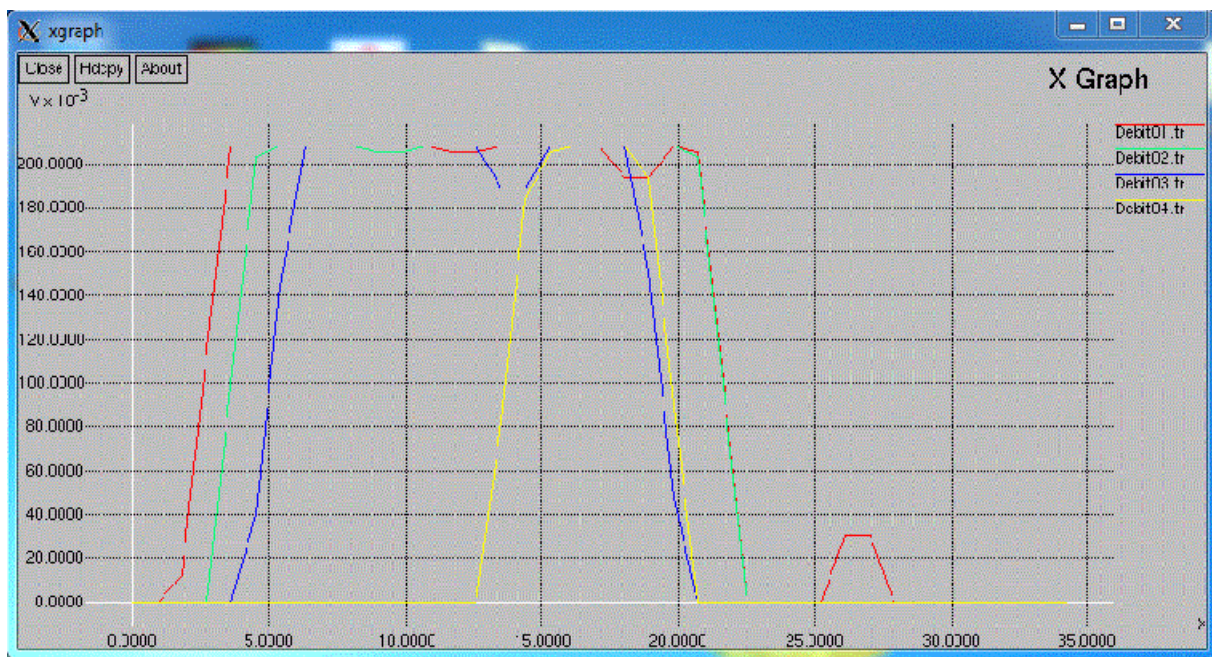


Figure IV.2 : L'évolution de débit de transmission de paquet des nœuds au cours du temps.

L'observation de ces courbes montre que le débit augmente progressivement au début de chaque transmission de paquet et prend une valeur maximale au bout d'un temps très court.

Ce débit est maintenu constant durant toute la période de communication et ceci peut être interprété par le fait que les nœuds sources et destinataires sont dans la même portée de transmission. Au bout d'un certain temps le débit commence à chuter car les nœuds ne sont plus en communication.

A 25s de temps de la simulation, le nœud 4 initie une nouvelle transmission de données. Les paquets de données issus de nœud 4 sont perdus car le protocole AODV ne trouve pas de nœud intermédiaire pour les acheminer à la destination, ainsi le débit est faible à cet instant.

La figure IV.3 montre la position des nœuds au moment où le débit est maximum pour les quatre applications (16.17s).

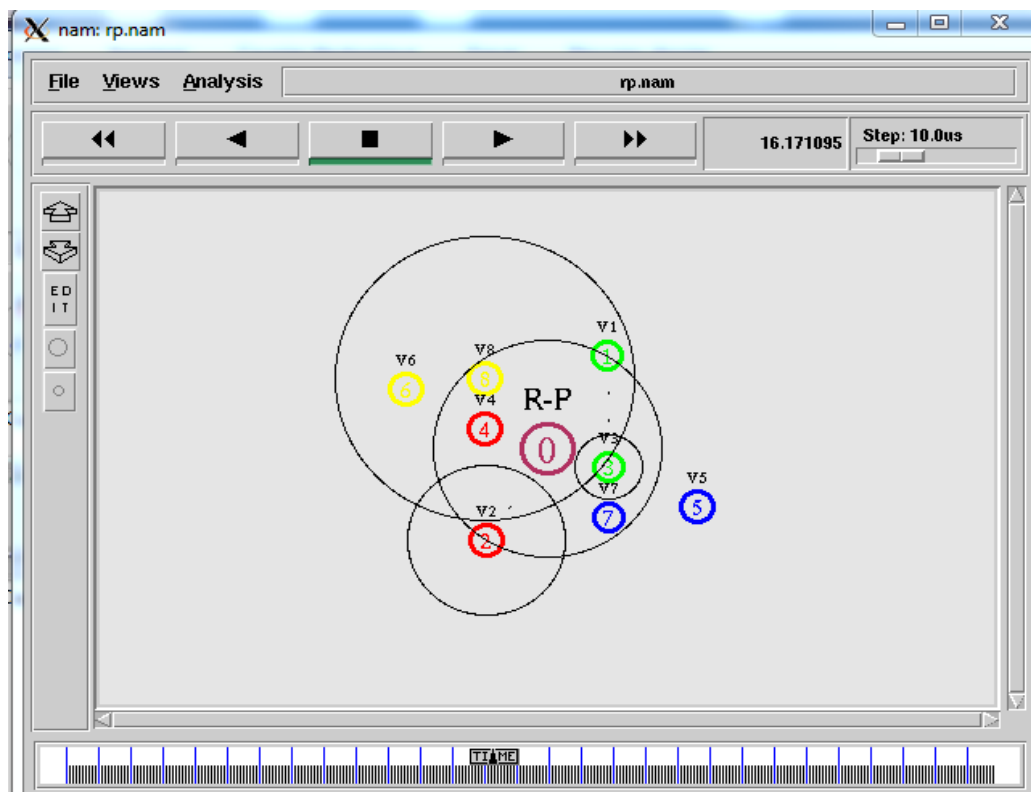


Figure IV.3 : La position des nœuds a 16.17s de la simulation.

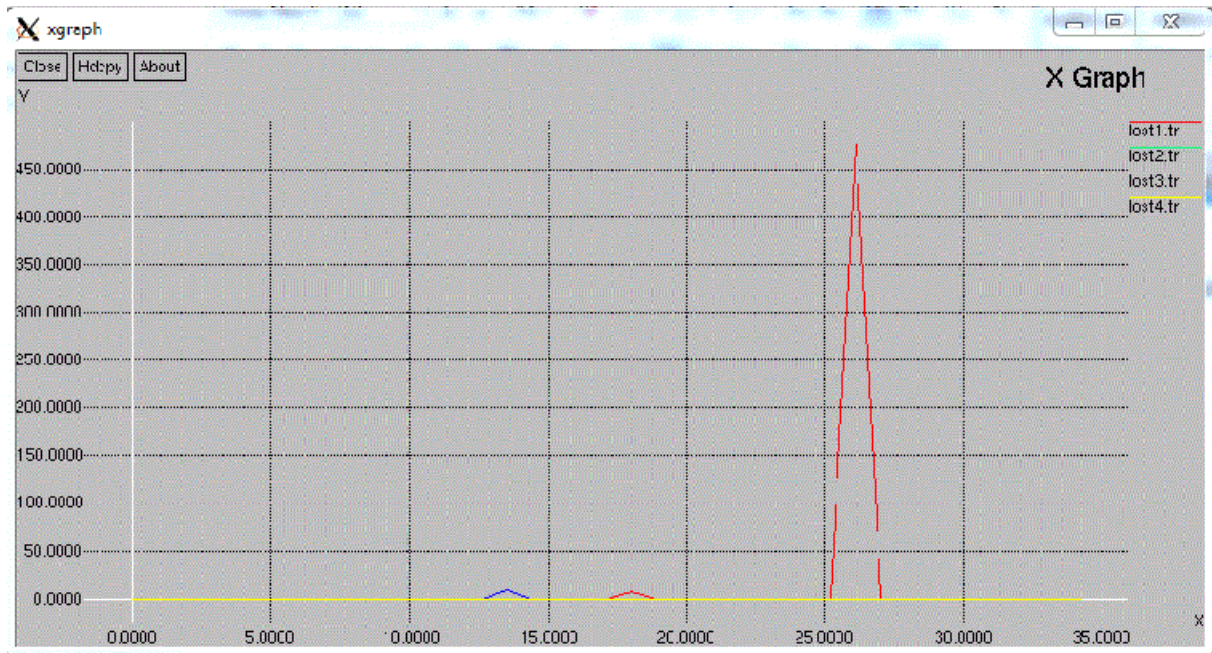


Figure IV.4 : L'évolution des pertes des paquets de transmission au cours du temps.

D'après ce graphe on constate que les pertes des paquets sont minimales pour deux applications et nulles pour deux autres. A 25s, il y a l'apparition d'un pic qui représente des paquets perdus relatifs au nœud 2.

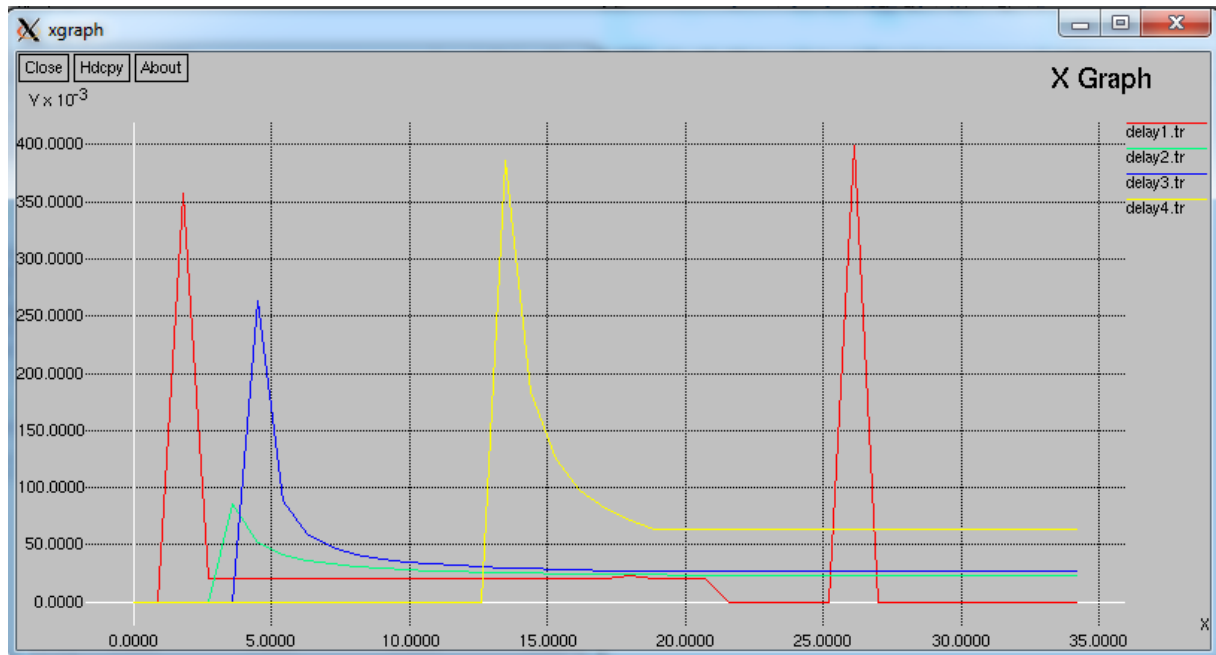


Figure IV.5 : L'évolution de délai de bout en bout au cours du temps.

Le délai de transit des paquets est important au début de chaque transmission de trafic qui dure quelques secondes et cela est dû au mécanisme propre de recherche des routes du protocole AODV. Une fois les routes sont maintenues le délai de bout en bout devient moins important et garde une valeur constante ainsi tous les paquets sont routés.

IV.2.2. Scénario 2 :

L'environnement simulé dans ce scénario est une autoroute bidirectionnelle comportant 4 voies composée de 24 nœuds mobiles, chaque voie comporte 6 nœuds mobiles (véhicules) plus 3 nœuds statiques définis comme des stations au bord de la route RSU. Les véhicules se déplacent avec une vitesse qui varie de 120 à 150 km/h, le temps de simulation est de 20s.

La position des nœuds et leur vitesse sont introduites dans le fichier tcl.

Le tableau ci-dessous illustre le contexte de notre simulation dans le scénario 2 :

Critère	Valeurs
Protocole MAC	IEEE 802.11
Modèle de propagation radio	Two-ray ground
Portée de communication	195m
Capacité du canal	2 Mbit/s
Taille max des files d'attente	50 paquets
Topologie réseau	1000m x 800m
Nombre de nœuds	27 nœuds (24 mobiles, 3 statique)
Vitesse des véhicules	120 à 150 km/h
Temps de simulation	20s
Couche transport	UDP
Type du trafic généré	CBR
Nombre de connexions	4 connexions
Taille du paquet de données	512 octet
Le paramètre simulé	Le délai de bout en bout le débit de transmission la perte des paquets
Transmissions effectuées	Transmission entre les nœuds : 26 et 11,24 et 21, 25 et 17, 5 et 16.
Le protocole de routage	AODV
Antenne	Omni Antenna : Omnidirectionnel
Vitesse d'émission	200kbit/s

Tableau IV.2 : Paramètres de simulation dans le scenario 2.

IV.2.2.1. Résultats et interprétations de scénario 2 :

Un exemple de disposition initiale des nœuds visualisé par l'outil d'animation nam est donné dans la figure IV.5.

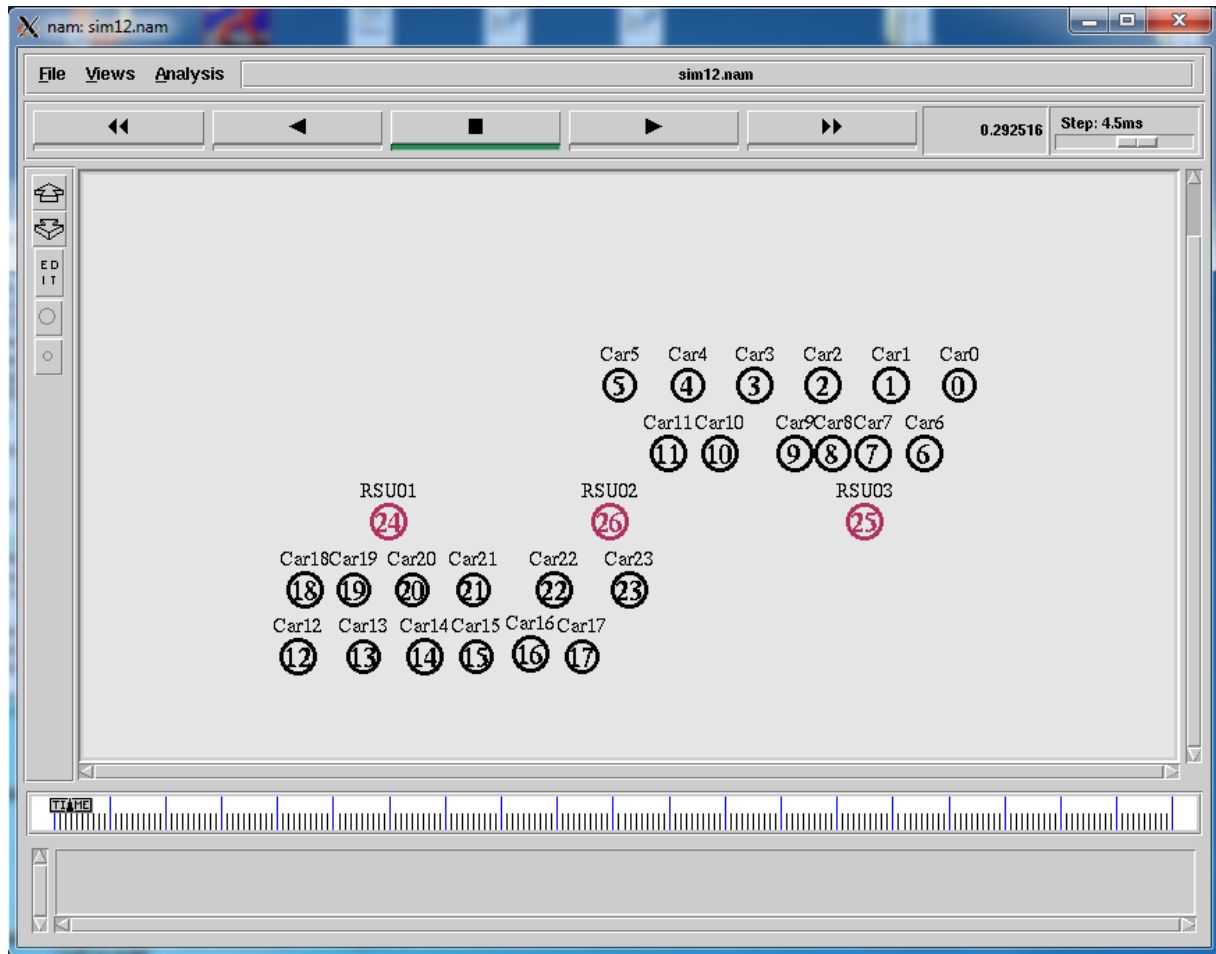


Figure IV.6 : Disposition initiale des nœuds.

Pour chacun de ces graphes, quatre courbes sont visualisées :

— : Correspondant au trafic de l'application1 attaché au nœud source (RSU) 26 et le nœud destinataire 11.

— : Correspondant au trafic de l'application2 entre la station au bord de la route RSU 24 et le nœud 21.

— : Correspondant au trafic de l'application3 entre la RSU25 et le nœud17.

— : Correspondant au trafic de l'application4 entre les nœuds 5 et 16.

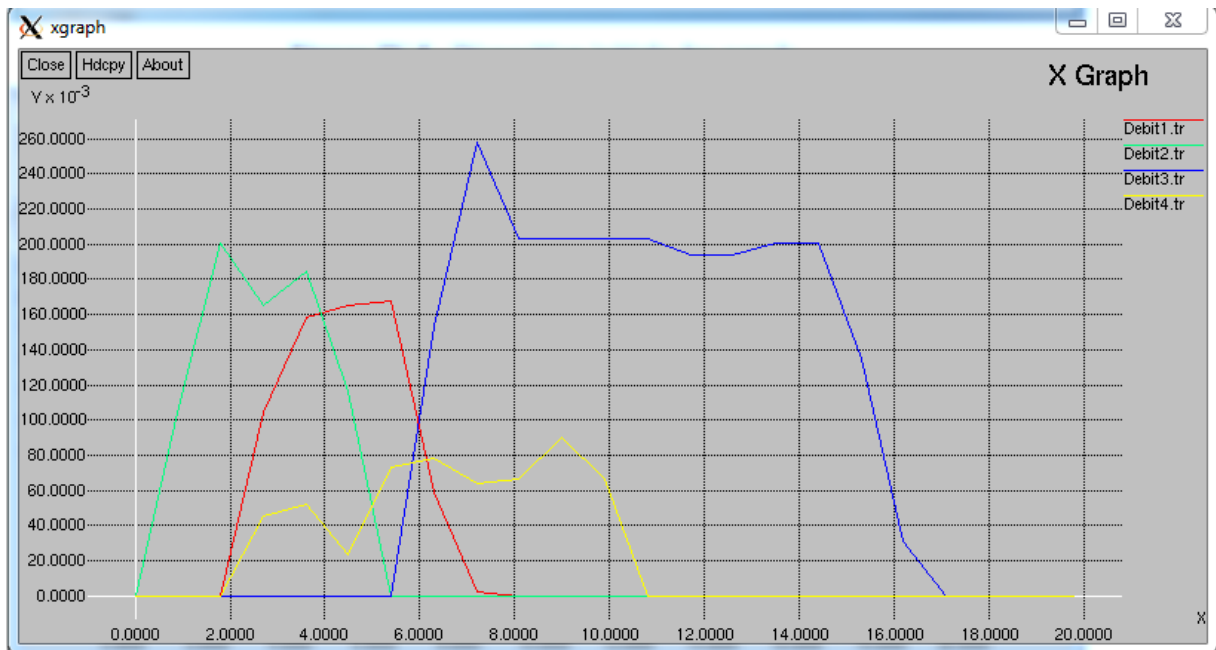


Figure IV.7 : L'évolution de débit de transmission de paquet des nœuds au cours du temps.

Le débit des quatre applications est différent car les nœuds se déplacent avec des vitesses différentes. Selon le graphe le débit est variable au cours du temps ceci est dû aux mouvements des nœuds qui changent la topologie du réseau et le protocole AODV initie à chaque fois une nouvelle demande de route.

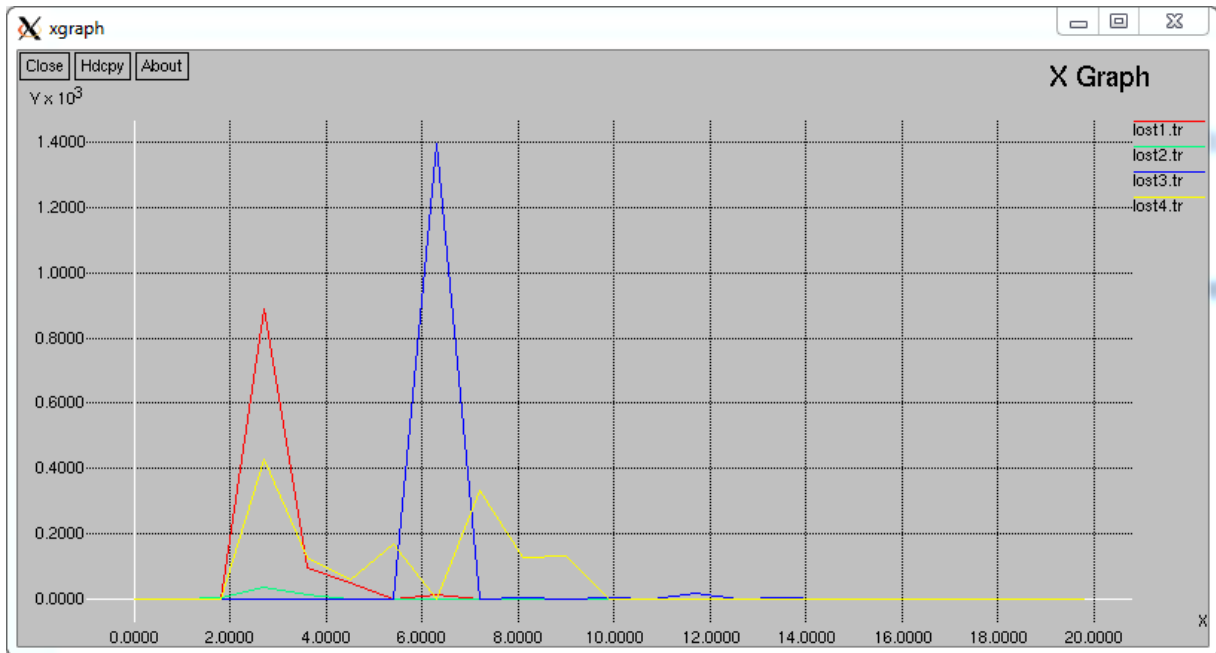


Figure IV.8 : L'évolution des pertes des paquets de transmission au cours du temps.

Nous remarquons d'après la figure IV.7 que le nombre de paquets perdus est important, ceci est dû à la forte mobilité des nœuds qui entraîne la reconfiguration du réseau.

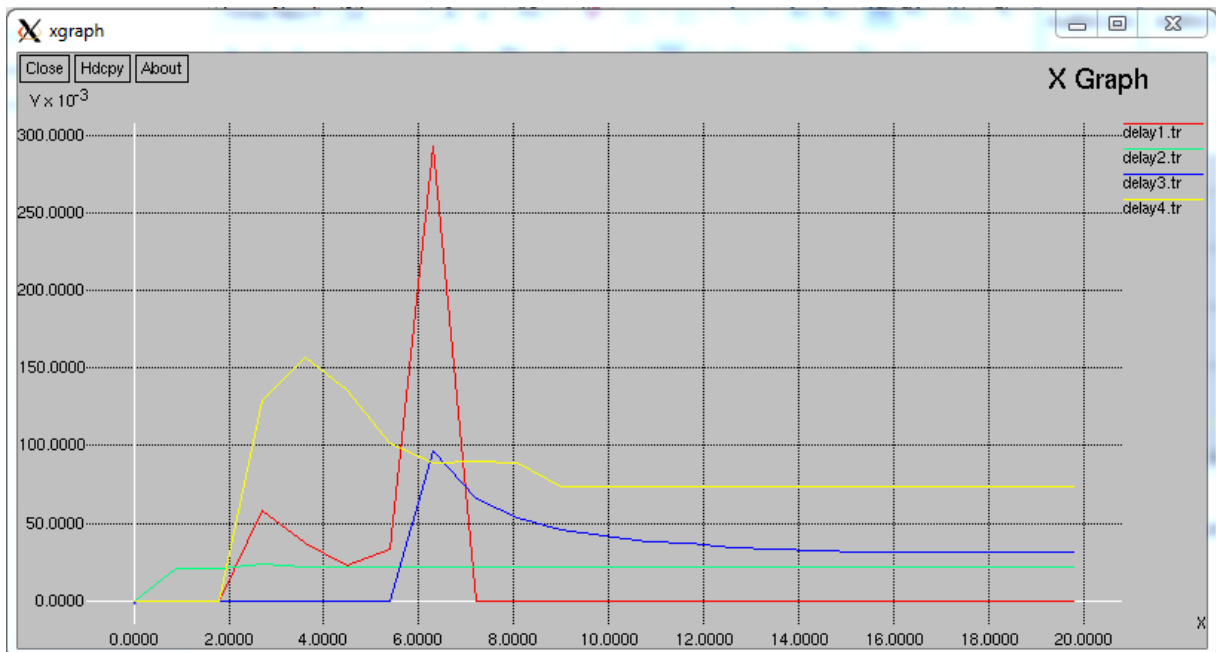


Figure IV.9 : L'évolution de délai de bout en bout au cours du temps.

Le délai de bout en bout dans la figure IV.8 prend des valeurs élevées au début des transmissions ceci s'explique par le fait que la communication passe par plusieurs nœuds

(multi saut), et se stabilise jusqu'à la fin de la communication. La figure IV.9 montre les paquets qui passent par plusieurs sauts.

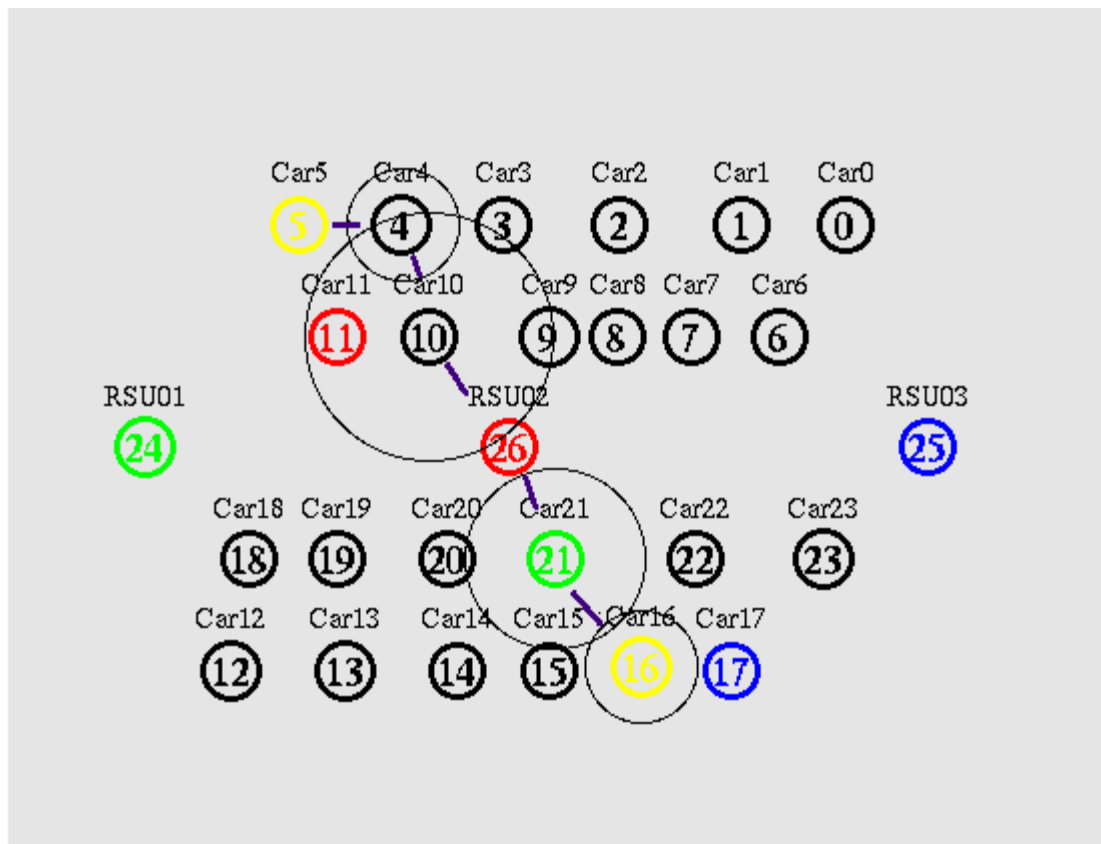


Figure IV.10 : Communication à multi saut entre les nœuds 5 et 16.

IV.3 : Discussion :

D'après les résultats de la simulation, nous avons constaté qu'il y a une différence de comportement du protocole AODV dans les deux scénarios.

Les résultats auxquels nous avons abouti montrent que les performances du protocole AODV dans le milieu autoroute se dégradent par rapport au milieu urbain en termes de débit, de taux de perte de paquets assez élevé et de délai de bout en bout assez important car la topologie du réseau change en permanence.

Le projet de notre mémoire s'est investi dans un domaine de recherche d'actualité, à savoir les réseaux véhiculaires et leur apport à la sécurité routière.

L'objectif principal de ce travail porte sur la simulation des réseaux véhiculaires sous le simulateur NS2, nous avons évalué les performances du protocole AODV de la classe des protocoles réactifs dans deux milieux différents urbain et autoroute.

Les métriques que nous avons évaluées sont celles qui renseignent sur le débit de transmission, le taux de pertes de paquets et le délai bout en bout. Les résultats des scénarios que nous avons considérés ont montré que dans le milieu urbain AODV a un débit élevé avec des pertes moins significatives et des délais de transmission de paquets courts.

Les résultats obtenus dans le deuxième scénario nous ont permis de conclure que le fait que la topologie du réseau soit de type très dynamique dégrade les performances du protocole AODV dans le milieu autoroute.

En effet, les résultats auxquels nous sommes parvenus ouvrent la voie à plusieurs perspectives qui peuvent faire objet de travaux futurs. Il serait également intéressant d'enrichir l'analyse des mécanismes de routage en considérant les nouveaux protocoles qui se basent sur les informations géographiques pour la détermination des routes comme GyTar et GPSR.

Annexe A : Le scripte de scénario 1

```

#[scenario 1]
#It consists of 8 mobiles nodes, 1 statics node (RP): 4 source nodes and 4 destination node.
Each source is a CBR source over UDP.
#The size of a transmitted packet is 512 bytes. Transmission rate of a node is 200 Kbps. We
assumed that the
#nodes are in transmission range at a constant distance of 195 m. The simulation time lasted
#for 35 sec.
#=====
#### Define options
#=====

set val(chan)      Channel/WirelessChannel  ;# channel type
set val(prop)      Propagation/TwoRayGround  ;# radio-propagation model
set val(netif)     Phy/WirelessPhy          ;# network interface type
set val(mac)       Mac/802_11               ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue  ;# interface queue type
set val(ll)        LL                       ;# link layer type
set val(ant)       Antenna/OmniAntenna      ;# antenna model
set val(ifqlen)    50                       ;# max packet in ifq
set val(nn)        9                        ;# number of mobilenodes
set val(rp)        AODV                     ;# routing protocol
set val(x)         1000                    ;# X dimension of the topography
set val(y)         800                     ;# Y dimension of the topography
set val(stop)      35                      ;# time of simulation end

Mac/802_11 set RTSThreshold_ 3000
Mac/802_11 set basicRate_ 1Mb
Mac/802_11 set dataRate_ 2Mb
#=====
# Initialize trace file descriptors
#=====

# *** Throughput Trace ***

set f0 [open Debit01.tr w]
set f1 [open Debit02.tr w]
set f2 [open Debit03.tr w]
set f3 [open Debit04.tr w]

# *** Packet Loss Trace ***

set f4 [open lost1.tr w]
set f5 [open lost2.tr w]
set f6 [open lost3.tr w]
set f7 [open lost4.tr w]

# *** Packet Delay Trace ***

```

```
set f8 [open delay1.tr w]
set f9 [open delay2.tr w]
set f10 [open delay3.tr w]
set f11 [open delay4.tr w]

# *** Initialize Simulator ***

set ns      [new Simulator]

# *** Initialize Trace file ***
set tracefd [open trace1.tr w]
$ns trace-all $tracefd
# *** Initialize Network Animator ***
set namtrace [open rp.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
# *** set up topography object ***
set topo     [new Topography]
$topo load_flatgrid 1000 800

# Create General Operations Director (GOD) object. It is used to store global information
about the state of the environment, network, or nodes that an

# omniscient observer would have, but that should not be made known to any participant in the
simulation.

create-god $val(nn)

# configure nodes

$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON

# Create Nodes

for {set i 0} {$i < $val(nn)} {incr i} {

    set node_($i) [$ns node]
```

```
$node_($i) random-motion 0      ;# disable random motion
}

# Initialize Node Coordinates

$node_(0) set X_ 500.0
$node_(0) set Y_ 400.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 100.0
$node_(1) set Y_ 300.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 850.0
$node_(2) set Y_ 500.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 600.0
$node_(3) set Y_ 100.0
$node_(3) set Z_ 0.0

$node_(4) set X_ 400.0
$node_(4) set Y_ 700.0
$node_(4) set Z_ 0.0
#####
$node_(5) set X_ 1.0
$node_(5) set Y_ 300.0
$node_(5) set Z_ 0.0

$node_(6) set X_ 950.0
$node_(6) set Y_ 500.0
$node_(6) set Z_ 0.0

$node_(7) set X_ 600 .0
$node_(7) set Y_ 1.0
$node_(7) set Z_ 0.0

$node_(8) set X_ 400.0
$node_(8) set Y_ 800.0
$node_(8) set Z_ 0.0

# Setup traffic flow between nodes
# UDP connections between node_(0) and node_(1)
# Create Constant four Bit Rate Traffic sources

#### APPLICATION (01) ####

set agent1 [new Agent/UDP]      ;# Create UDP Agent
```

```
$agent1 set prio_0           ;# Set Its priority to 0
set sink [new Agent/LossMonitor] ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(4) $agent1 ;# Attach Agent to source node
$ns attach-agent $node_(2) $sink ;# Attach Agent to sink node
$ns connect $agent1 $sink ;# Connect the nodes
set app1 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app1 set packetSize_ 512 ;# Set Packet Size to 512 bytes
$app1 set rate_ 200Kb ;# Set CBR rate to 200 Kbits/sec
$app1 attach-agent $agent1 ;# Attach Application to agent
```

APPLICATION (02)

```
set agent2 [new Agent/UDP] ;# Create UDP Agent
$agent2 set prio_ 1 ;# Set Its priority to 1
set sink2 [new Agent/LossMonitor] ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(1) $agent2 ;# Attach Agent to source node
$ns attach-agent $node_(3) $sink2 ;# Attach Agent to sink node
$ns connect $agent2 $sink2 ;# Connect the nodes
set app2 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app2 set packetSize_ 512 ;# Set Packet Size to 512 bytes
$app2 set rate_ 200Kb ;# Set CBR rate to 200 Kbits/sec
$app2 attach-agent $agent2 ;# Attach Application to agent
```

APPLICATION (03)

```
set agent3 [new Agent/UDP] ;# Create UDP Agent
$agent3 set prio_ 2 ;# Set Its priority to 2
set sink3 [new Agent/LossMonitor] ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(5) $agent3 ;# Attach Agent to source node
$ns attach-agent $node_(7) $sink3 ;# Attach Agent to sink node
$ns connect $agent3 $sink3 ;# Connect the nodes
set app3 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app3 set packetSize_ 512 ;# Set Packet Size to 512 bytes
$app3 set rate_ 200Kb ;# Set CBR rate to 200 Kbits/sec
$app3 attach-agent $agent3 ;# Attach Application to agent
```

APPLICATION (04)

```
set agent4 [new Agent/UDP] ;# Create UDP Agent
$agent4 set prio_ 3 ;# Set Its priority to 3
set sink4 [new Agent/LossMonitor] ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(6) $agent4 ;# Attach Agent to source node
$ns attach-agent $node_(8) $sink4 ;# Attach Agent to sink node
$ns connect $agent4 $sink4 ;# Connect the nodes
set app4 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app4 set packetSize_ 512 ;# Set Packet Size to 512 bytes
```

```
$app4 set rate_ 200Kb           ;# Set CBR rate to 200 Kbits/sec
$app4 attach-agent $agent4     ;# Attach Application to agent
```

```
# defines the node size in Network Animator
#for {set i 0} {$i < $val(nn)} {incr i} {
#  $ns initial_node_pos $node_($i) 50
#}
```

```
##### SIZE NODES #####
```

```
$ns initial_node_pos $node_(0) 85
$ns initial_node_pos $node_(1) 50
$ns initial_node_pos $node_(2) 50
$ns initial_node_pos $node_(3) 50
$ns initial_node_pos $node_(4) 50
$ns initial_node_pos $node_(5) 50
$ns initial_node_pos $node_(6) 50
$ns initial_node_pos $node_(7) 50
$ns initial_node_pos $node_(8) 50
```

```
##### Setting The Labels For Nodes
```

```
$ns at 0.0 "$node_(0) label R-P"
$ns at 0.0 "$node_(1) label V1"
$ns at 0.0 "$node_(2) label V2"
$ns at 0.0 "$node_(3) label V3"
$ns at 0.0 "$node_(4) label V4"
$ns at 0.0 "$node_(5) label V5"
$ns at 0.0 "$node_(6) label V6"
$ns at 0.0 "$node_(7) label V7"
$ns at 0.0 "$node_(8) label V8"
```

```
###Colors nodes#####
```

```
$node_(0) color Maroon
$ns at 0.01 "$node_(0) color Maroon "
```

```
$node_(4) color red
$ns at 1.70 "$node_(4) color red"
$node_(2) color red
$ns at 1.70 "$node_(2) color red"
```

```
$node_(1) color green
$ns at 2.75 "$node_(1) color green"
$node_(3) color green
$ns at 2.75 "$node_(3) color green"
```

```
$node_(5) color Blue
```

```
$ns at 4.15 "$node_(5) color Blue"
$node_(7) color Blue
$ns at 4.15 "$node_(7) color Blue"
```

```
$node_(6) color yellow
$ns at 12.80 "$node_(6) color yellow"
$node_(8) color yellow
$ns at 12.80 "$node_(8) color yellow"
```

```
#####
```

```
$ns at 0.0 "$node_(0) delete-mark R-P"
$ns at 0.01 "$node_(0) add-mark $node_(0) Orange Circle"
```

```
#####
```

##Giving Mobility to Nodes

```
$ns at 0.0 "$ns set-animation-rate 12.0ms"
  $ns at 0.0 "$node_(0) setdest 500.0 400.0 0.0"
#####
  $ns at 0.0 "$node_(1) setdest 400.0 300.0 100.0"
  $ns at 0.0 "$node_(5) setdest 340.0 300.0 80.0"
#####
  $ns at 0.0 "$node_(2) setdest 600.0 500.0 90.0"
  $ns at 0.0 "$node_(6) setdest 660.0 500.0 60.0"
#####
  $ns at 0.0 "$node_(3) setdest 600.0 260.0 85.0"
  $ns at 0.0 "$node_(7) setdest 600.0 200.0 65.0"
#####
  $ns at 0.0 "$node_(4) setdest 400.0 540.0 98.0"
  $ns at 0.0 "$node_(8) setdest 400.0 600.0 70.0"
#####
  $ns at 4.55 "$node_(1) setdest 600.0 300.0 40.0"
  $ns at 4.55 "$node_(2) setdest 400.0 500.0 40.0"
  $ns at 4.55 "$node_(5) setdest 550.0 300.0 30.0"
  $ns at 4.55 "$node_(6) setdest 450.0 500.0 30.0"
#####
  $ns at 9.45 "$node_(1) setdest 600.0 799.0 40.0"
  $ns at 9.45 "$node_(2) setdest 400.0 1.0 40.0"
  $ns at 9.45 "$node_(5) setdest 999.0 300.0 40.0"
  $ns at 9.45 "$node_(6) setdest 1.0 500.0 40.0"
#####
  $ns at 13.50 "$node_(3) setdest 600.0 500.0 40.0"
  $ns at 13.50 "$node_(4) setdest 400.0 300.0 40.0"
  $ns at 13.50 "$node_(7) setdest 600.0 400.0 30.0"
  $ns at 13.50 "$node_(8) setdest 400.0 240.0 30.0"
#####
  $ns at 20.0 "$node_(3) setdest 60.0 500.0 40.0"
  $ns at 20.0 "$node_(4) setdest 940.0 300.0 40.0"
  $ns at 20.0 "$node_(7) setdest 600.0 740.0 40.0"
```

```
$ns at 20.0 "$node_(8) setdest 400.0 60.0 40.0"
# Initialize Flags

set holdtime 0
set holdseq 0
set holdtime1 0
set holdseq1 0
set holdtime2 0
set holdseq2 0
set holdtime3 0
set holdseq3 0

set holdrate1 0
set holdrate2 0
set holdrate3 0
set holdrate4 0

# Function To record Statisticis (Bit Rate, Delay, Drop)

proc record { } {
    global sink sink2 sink3 sink4 f0 f1 f2 f3 f4 f5 f6 f7 holdtime holdseq holdtime1 holdseq1
    holdtime2 holdseq2 holdtime3 holdseq3 f8 f9 f10 f11 holdrate1 holdrate2 holdrate3 holdrate4

    set ns [Simulator instance]
    set time 0.9 ;#Set Sampling Time to 0.9 Sec

    set bw0 [$sink set bytes_]
    set bw1 [$sink2 set bytes_]
    set bw2 [$sink3 set bytes_]
    set bw3 [$sink4 set bytes_]
    set bw4 [$sink set nlost_]
    set bw5 [$sink2 set nlost_]
    set bw6 [$sink3 set nlost_]
    set bw7 [$sink4 set nlost_]
    set bw8 [$sink set lastPktTime_]
    set bw9 [$sink set npkts_]
    set bw10 [$sink2 set lastPktTime_]
    set bw11 [$sink2 set npkts_]
    set bw12 [$sink3 set lastPktTime_]
    set bw13 [$sink3 set npkts_]
    set bw14 [$sink4 set lastPktTime_]
    set bw15 [$sink4 set npkts_]

    set now [$ns now]

    # Record Bit Rate in Trace Files

    puts $f0 "$now [expr ((bw0+$holdrate1)*8)/(2*$time*1000000)]"
    puts $f1 "$now [expr ((bw1+$holdrate2)*8)/(2*$time*1000000)]"
    puts $f2 "$now [expr ((bw2+$holdrate3)*8)/(2*$time*1000000)]"
```

```
puts $f3 "$now [expr (($bw3+$holdrate4)*8)/(2*$time*1000000)]"

# Record Packet Loss Rate in File
puts $f4 "$now [expr $bw4/$time]"
puts $f5 "$now [expr $bw5/$time]"
puts $f6 "$now [expr $bw6/$time]"
puts $f7 "$now [expr $bw7/$time]"

# Record Packet Delay in File
if { $bw9 > $holdseq } {
    puts $f8 "$now [expr ($bw8 - $holdtime)/($bw9 - $holdseq)]"
} else {
    puts $f8 "$now [expr ($bw9 - $holdseq)]"
}
if { $bw11 > $holdseq1 } {
    puts $f9 "$now [expr ($bw10 - $holdtime1)/($bw11 - $holdseq1)]"
} else {
    puts $f9 "$now [expr ($bw11 - $holdseq1)]"
}

if { $bw13 > $holdseq2 } {
    puts $f10 "$now [expr ($bw12 - $holdtime2)/($bw13 - $holdseq2)]"
} else {
    puts $f10 "$now [expr ($bw13 - $holdseq2)]"
}
if { $bw15 > $holdseq3 } {
    puts $f11 "$now [expr ($bw14 - $holdtime3)/($bw15 - $holdseq3)]"
} else {
    puts $f11 "$now [expr ($bw15 - $holdseq3)]"
}

# Reset Variables
$sink set bytes_ 0
$sink2 set bytes_ 0
$sink3 set bytes_ 0
$sink4 set bytes_ 0

$sink set nlost_ 0
$sink2 set nlost_ 0
$sink3 set nlost_ 0
$sink4 set nlost_ 0

set holdtime $bw8
set holdseq $bw9
set holdrate1 $bw0
set holdrate2 $bw1
set holdrate3 $bw2
set holdrate4 $bw3
```

```

$ns at [expr $now+$time] "record" ;# Schedule Record after $time interval sec
}

# Start Recording at Time 0

$ns at 0.0 "record"
$ns at 1.70 "$app1 start" ;# Start transmission at time t = 1.4 Sec
$ns at 2.75 "$app2 start" ;# Start transmission at time t = 10 Sec
$ns at 4.15 "$app3 start" ;# Start transmission at time t = 20 Sec
$ns at 12.80 "$app4 start" ;# Start transmission at time t = 30 Sec

# Stop Simulation at Time 35.0 sec
$ns at 35.0 "stop"

# Reset Nodes at time 35.0 sec
for {set i 0} {$i < $val(nn) } {incr i} {

    $ns at 35.0 "$node_($i) reset";
}
# Exit Simulatoion at Time 35.01 sec
$ns at 35.01 "puts \"NS EXITING...\" ; $ns halt"
proc stop {} {
    global ns tracefd f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11
# Close Trace Files

    close $f0
    close $f1
    close $f2
    close $f3
    close $f4
    close $f5
    close $f6
    close $f7
    close $f8
    close $f9
    close $f10
    close $f11

# Plot Recorded Statistics

    exec xgraph Debit01.tr Debit02.tr Debit03.tr Debit04.tr -geometry 800x400 &
    exec xgraph lost1.tr lost2.tr lost3.tr lost4.tr -geometry 800x400 &
    exec xgraph delay1.tr delay2.tr delay3.tr delay4.tr -geometry 800x400 &

# Reset Trace File
$ns flush-trace
close $tracefd

```

```
    exec nam rp.nam &  
    exit 0  
}  
  
puts "Starting Simulation..."  
  
$ns run
```

Annexe B : Le script de scénario 2

```
#[scenario 2]
#It consists of 24 mobiles nodes, 03 statics nodes (RSU): 4 source nodes and 4 destination
#node. Each source is a CBR source over UDP.
#The size of a transmitted packet is 512 bytes. Transmission rate of a node is 200 Kbps. We
#assumed that the
#nodes are in transmission range at a constant distance of 195 m. The simulation time lasted
#for 20 sec.
```

```
#=====
# Define Node Configuration paramaters
#=====
```

```
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 27 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 2000 ;# X dimension of the topography
set val(y) 1500 ;# Y dimension of the topography
set val(stop) 20 ;# time of simulation end
```

```
Mac/802_11 set RTSThreshold_ 3000
Mac/802_11 set basicRate_ 1Mb
Mac/802_11 set dataRate_ 2Mb
```

```
#=====
# Initialize trace file descriptors
#=====
```

```
# *** Throughput Trace ***
```

```
set f0 [open Debit1.tr w]
set f1 [open Debit2.tr w]
set f2 [open Debit3.tr w]
set f3 [open Debit4.tr w]
```

```
# *** Packet Loss Trace ***
```

```
set f4 [open lost1.tr w]
set f5 [open lost2.tr w]
set f6 [open lost3.tr w]
set f7 [open lost4.tr w]
```

```
# *** Packet Delay Trace ***

set f8 [open delay1.tr w]
set f9 [open delay2.tr w]
set f10 [open delay3.tr w]
set f11 [open delay4.tr w]

# *** Initialize Simulator ***

set ns [new Simulator]

# *** Initialize Trace file ***

set tracefd [open trace2.tr w]
$ns trace-all $tracefd

# *** Initialize Network Animator ***

set namtrace [open sim12.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
$ns namtrace-all $namtrace

set topo [new Topography]
$topo load_flatgrid 2000 1500
create-god $val(nn)

# configure nodes

$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channelType $val(chan) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON

#####
#defining node size and motion
#####

for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns node]
```

}

Initialize Node Coordinates

#Voies aller (1)

\$node_(0) set X_ 2000.0

\$node_(0) set Y_ 800.0

\$node_(0) set Z_ 0.0

\$node_(1) set X_ 1800.0

\$node_(1) set Y_ 800.0

\$node_(1) set Z_ 0.0

\$node_(2) set X_ 1600.0

\$node_(2) set Y_ 800.0

\$node_(2) set Z_ 0.0

\$node_(3) set X_ 1400.0

\$node_(3) set Y_ 800.0

\$node_(3) set Z_ 0.0

\$node_(4) set X_ 1200.0

\$node_(4) set Y_ 800.0

\$node_(4) set Z_ 0.0

\$node_(5) set X_ 1000.0

\$node_(5) set Y_ 800.0

\$node_(5) set Z_ 0.0

#voies aller (2)

\$node_(6) set X_ 1900.0

\$node_(6) set Y_ 600.0

\$node_(6) set Z_ 0.0

\$node_(7) set X_ 1750.0

\$node_(7) set Y_ 600.0

\$node_(7) set Z_ 0.0

\$node_(8) set X_ 1630.0

\$node_(8) set Y_ 600.0

\$node_(8) set Z_ 0.0

\$node_(9) set X_ 1520.0

\$node_(9) set Y_ 600.0

\$node_(9) set Z_ 0.0

\$node_(10) set X_ 1300.0

\$node_(10) set Y_ 600.0

\$node_(10) set Z_ 0.0

\$node_(11) set X_ 1150.0
\$node_(11) set Y_ 600.0
\$node_(11) set Z_ 0.0

#voies retour (1)

\$node_(12) set X_ 10.0
\$node_(12) set Y_ 1.0
\$node_(12) set Z_ 0.0

\$node_(13) set X_ 200.0
\$node_(13) set Y_ 1.0
\$node_(13) set Z_ 0.0

\$node_(14) set X_ 380.0
\$node_(14) set Y_ 1.0
\$node_(14) set Z_ 0.0

\$node_(15) set X_ 530.0
\$node_(15) set Y_ 1.0
\$node_(15) set Z_ 0.0

\$node_(16) set X_ 690.0
\$node_(16) set Y_ 10.0
\$node_(16) set Z_ 0.0

\$node_(17) set X_ 840.0
\$node_(17) set Y_ 1.0
\$node_(17) set Z_ 0.0

#voies retour (2)

\$node_(18) set X_ 30.0
\$node_(18) set Y_ 200.0
\$node_(18) set Z_ 0.0

\$node_(19) set X_ 170.0
\$node_(19) set Y_ 200.0
\$node_(19) set Z_ 0.0

\$node_(20) set X_ 340.0
\$node_(20) set Y_ 200.0
\$node_(20) set Z_ 0.0

\$node_(21) set X_ 520.0
\$node_(21) set Y_ 200.0
\$node_(21) set Z_ 0.0

\$node_(22) set X_ 760.0

```
$node_(22) set Y_ 200.0  
$node_(22) set Z_ 0.0
```

```
$node_(23) set X_ 980.0  
$node_(23) set Y_ 200.0  
$node_(23) set Z_ 0.0
```

```
#stations RSU
```

```
$node_(24) set X_ 300.0  
$node_(24) set Y_ 400.0  
$node_(24) set Z_ 0.0
```

```
$node_(25) set X_ 1700.0  
$node_(25) set Y_ 400.0  
$node_(25) set Z_ 0.0
```

```
$node_(26) set X_ 950.0  
$node_(26) set Y_ 400.0  
$node_(26) set Z_ 0.0
```

```
#### Setting The Labels For Nodes
```

```
  $ns at 0.0 "$node_(0) label Car0"  
  $ns at 0.0 "$node_(1) label Car1"  
  $ns at 0.0 "$node_(2) label Car2"  
  $ns at 0.0 "$node_(3) label Car3"  
  $ns at 0.0 "$node_(4) label Car4"  
  $ns at 0.0 "$node_(5) label Car5"  
  $ns at 0.0 "$node_(6) label Car6"  
  $ns at 0.0 "$node_(7) label Car7"  
  $ns at 0.0 "$node_(8) label Car8"  
  $ns at 0.0 "$node_(9) label Car9"  
  $ns at 0.0 "$node_(10) label Car10"  
  $ns at 0.0 "$node_(11) label Car11"  
  $ns at 0.0 "$node_(12) label Car12"  
  $ns at 0.0 "$node_(13) label Car13"  
  $ns at 0.0 "$node_(14) label Car14"  
  $ns at 0.0 "$node_(15) label Car15"  
  $ns at 0.0 "$node_(16) label Car16"  
  $ns at 0.0 "$node_(17) label Car17"  
  $ns at 0.0 "$node_(18) label Car18"  
  $ns at 0.0 "$node_(19) label Car19"  
  $ns at 0.0 "$node_(20) label Car20"  
  $ns at 0.0 "$node_(21) label Car21"  
  $ns at 0.0 "$node_(22) label Car22"  
  $ns at 0.0 "$node_(23) label Car23"  
  $ns at 0.0 "$node_(24) label RSU01"  
  $ns at 0.0 "$node_(25) label RSU03"  
  $ns at 0.0 "$node_(26) label RSU02"
```

```
###Colors nodes#####
```

```
$node_(11) color red
$ns at 1.10 "$node_(11) color red"
$node_(26) color red
$ns at 1.10 "$node_(26) color red"
```

```
$node_(24) color green
$ns at 1.20 "$node_(24) color green"
$node_(21) color green
$ns at 1.20 "$node_(21) color green"
```

```
$node_(25) color Blue
$ns at 1.80 "$node_(25) color Blue"
$node_(17) color Blue
$ns at 1.80 "$node_(17) color Blue"
```

```
$node_(5) color yellow
$ns at 1.99 "$node_(5) color yellow"
$node_(16) color yellow
$ns at 1.99 "$node_(16) color yellow"
```

```
### RSU #####
```

```
$node_(24) color Maroon
$ns at 0.0 "$node_(24) color Maroon "
$node_(25) color Maroon
$ns at 0.0 "$node_(25) color Maroon "
$node_(26) color Maroon
$ns at 0.0 "$node_(26) color Maroon "
```

```
#Color change while moving from one group to another
```

```
$ns at 0.0 "$node_(24) delete-mark RSU01"
$ns at 0.01 "$node_(24) add-mark $node_(24) fuchsia Circle"
$ns at 0.0 "$node_(24) delete-mark RSU01"
$ns at 1.20 "$node_(24) add-mark $node_(24) green Circle"
```

```
$ns at 0.0 "$node_(25) delete-mark RSU03"
$ns at 0.02 "$node_(25) add-mark $node_(25) fuchsia Circle"
$ns at 0.0 "$node_(25) delete-mark RSU03"
$ns at 1.80 "$node_(25) add-mark $node_(25) blue Circle"
```

```
$ns at 0.0 "$node_(26) delete-mark RSU02"
$ns at 0.03 "$node_(26) add-mark $node_(26) fuchsia Circle"
$ns at 0.0 "$node_(26) delete-mark RSU02"
$ns at 1.10 "$node_(26) add-mark $node_(26) red Circle"
```

```
#Animate Few of node
```

```
$ns at 0.1 "$node_(0) setdest 0.10 800.0 123.0"
$ns at 0.1 "$node_(1) setdest 0.10 800.0 119.0"
$ns at 0.1 "$node_(2) setdest 1.0 800.0 124.0"
$ns at 0.1 "$node_(3) setdest 1.0 800.0 126.0"
$ns at 0.1 "$node_(4) setdest 1.0 800.0 121.0"
$ns at 0.1 "$node_(5) setdest 1.0 800.0 117.0"

$ns at 0.1 "$node_(6) setdest 1.0 600.0 128.0"
$ns at 0.1 "$node_(7) setdest 1.0 600.0 130.0"
$ns at 0.1 "$node_(8) setdest 1.0 600.0 133.0"
$ns at 0.1 "$node_(9) setdest 1.0 600.0 136.0"
$ns at 0.1 "$node_(10) setdest 1.0 600.0 135.0"
$ns at 0.1 "$node_(11) setdest 1.0 600.0 139.0"

$ns at 0.1 "$node_(12) setdest 1999.0 1.0 122.0"
$ns at 0.1 "$node_(13) setdest 1999.0 1.0 126.0"
$ns at 0.1 "$node_(14) setdest 1999.0 1.0 131.0"
$ns at 0.1 "$node_(15) setdest 1999.0 1.0 135.0"
$ns at 0.1 "$node_(16) setdest 1999.0 1.0 137.0"
$ns at 0.1 "$node_(17) setdest 1999.0 1.0 140.0"

$ns at 0.1 "$node_(18) setdest 1999.0 200.0 125.0"
$ns at 0.1 "$node_(19) setdest 1999.0 200.0 130.0"
$ns at 0.1 "$node_(20) setdest 1999.0 200.0 137.0"
$ns at 0.1 "$node_(21) setdest 1999.0 200.0 141.0"
$ns at 0.1 "$node_(22) setdest 1999.0 200.0 144.0"
$ns at 0.1 "$node_(23) setdest 1999.0 200.0 147.0"

##### RSU #####

$ns at 0.1 "$node_(24) setdest 300.0 750.0 0.0"
$ns at 0.1 "$node_(25) setdest 1700.0 750.0 0.0"
$ns at 0.1 "$node_(26) setdest 950.0 750.0 0.0"

# Setup traffic flow between nodes
# UDP connections between node_(26) and node_(11)

$ns at 0.0 "$ns set-animation-rate 4.5ms"

set agent1 [new Agent/UDP]           ;# Create UDP Agent
$agent1 set prio_ 0                   ;# Set Its priority to 0
set sink [new Agent/LossMonitor]      ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(26) $agent1   ;# Attach Agent to source node
$ns attach-agent $node_(11) $sink     ;# Attach Agent to sink node
$ns connect $agent1 $sink              ;# Connect the nodes
set app1 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app1 set packetSize_ 512             ;# Set Packet Size to 512 bytes
$app1 set rate_ 200Kb                 ;# Set CBR rate to 200 Kbits/sec
```

```

$app1 attach-agent $agent1          ;# Attach Application to agent

set agent2 [new Agent/UDP]          ;# Create UDP Agent
$agent2 set prio_ 1                 ;# Set Its priority to 1
set sink2 [new Agent/LossMonitor]   ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(24) $agent2 ;# Attach Agent to source node
$ns attach-agent $node_(21) $sink2  ;# Attach Agent to sink node
$ns connect $agent2 $sink2         ;# Connect the nodes
set app2 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app2 set packetSize_ 512          ;# Set Packet Size to 512 bytes
$app2 set rate_ 200Kb              ;# Set CBR rate to 200 Kbits/sec
$app2 attach-agent $agent2         ;# Attach Application to agent

set agent3 [new Agent/UDP]          ;# Create UDP Agent
$agent3 set prio_ 2                 ;# Set Its priority to 2
set sink3 [new Agent/LossMonitor]   ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(25) $agent3 ;# Attach Agent to source node
$ns attach-agent $node_(17) $sink3  ;# Attach Agent to sink node
$ns connect $agent3 $sink3         ;# Connect the nodes
set app3 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app3 set packetSize_ 512          ;# Set Packet Size to 512 bytes
$app3 set rate_ 200Kb              ;# Set CBR rate to 200 Kbits/sec
$app3 attach-agent $agent3         ;# Attach Application to agent

set agent4 [new Agent/UDP]          ;# Create UDP Agent
$agent4 set prio_ 3                 ;# Set Its priority to 3
set sink4 [new Agent/LossMonitor]   ;# Create Loss Monitor Sink in order to be able to
trace the number obytes received
$ns attach-agent $node_(5) $agent4  ;# Attach Agent to source node
$ns attach-agent $node_(16) $sink4  ;# Attach Agent to sink node
$ns connect $agent4 $sink4         ;# Connect the nodes
set app4 [new Application/Traffic/CBR] ;# Create Constant Bit Rate application
$app4 set packetSize_ 512          ;# Set Packet Size to 512 bytes
$app4 set rate_ 200Kb              ;# Set CBR rate to 200 Kbits/sec
$app4 attach-agent $agent4         ;# Attach Application to agent

# defines the node size in Network Animator
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns initial_node_pos $node_($i) 95
}
# Initialize Flags
set holdtime 0
set holdseq 0
set holdtime1 0
set holdseq1 0
set holdtime2 0
set holdseq2 0

```

```

set holdtime3 0
set holdseq3 0
set holdrate1 0
set holdrate2 0
set holdrate3 0
set holdrate4 0

```

```
# Function To record Statistcis (Bit Rate, Delay, Drop)
```

```
proc record { } {
```

```

    global sink sink2 sink3 sink4 f0 f1 f2 f3 f4 f5 f6 f7 holdtime holdseq holdtime1 holdseq1
    holdtime2 holdseq2 holdtime3 holdseq3 f8 f9 f10 f11 holdrate1 holdrate2 holdrate3 holdrate4

```

```

    set ns [Simulator instance]
    set time 0.9 ;#Set Sampling Time to 0.9 Sec

```

```

    set bw0 [$sink set bytes_]
    set bw1 [$sink2 set bytes_]
    set bw2 [$sink3 set bytes_]
    set bw3 [$sink4 set bytes_]

```

```

    set bw4 [$sink set nlost_]
    set bw5 [$sink2 set nlost_]
    set bw6 [$sink3 set nlost_]
    set bw7 [$sink4 set nlost_]

```

```

    set bw8 [$sink set lastPktTime_]
    set bw9 [$sink set npkts_]

```

```

    set bw10 [$sink2 set lastPktTime_]
    set bw11 [$sink2 set npkts_]

```

```

    set bw12 [$sink3 set lastPktTime_]
    set bw13 [$sink3 set npkts_]

```

```

    set bw14 [$sink4 set lastPktTime_]
    set bw15 [$sink4 set npkts_]

```

```
set now [$ns now]
```

```
# Record Bit Rate in Trace Files
```

```

puts $f0 "$now [expr (($bw0+$holdrate1)*8)/(2*$time*1000000)]"
puts $f1 "$now [expr (($bw1+$holdrate2)*8)/(2*$time*1000000)]"
puts $f2 "$now [expr (($bw2+$holdrate3)*8)/(2*$time*1000000)]"
puts $f3 "$now [expr (($bw3+$holdrate4)*8)/(2*$time*1000000)]"

```

```
# Record Packet Loss Rate in File
```

```

puts $f4 "$now [expr $bw4/$time]"
puts $f5 "$now [expr $bw5/$time]"
puts $f6 "$now [expr $bw6/$time]"

```

```

puts $f7 "$now [expr $bw7/$time]"

# Record Packet Delay in File

if { $bw9 > $holdseq } {
    puts $f8 "$now [expr ($bw8 - $holdtime)/($bw9 - $holdseq)]"
} else {

    puts $f8 "$now [expr ($bw9 - $holdseq)]"
}

if { $bw11 > $holdseq1 } {
    puts $f9 "$now [expr ($bw10 - $holdtime1)/($bw11 - $holdseq1)]"
} else {
    puts $f9 "$now [expr ($bw11 - $holdseq1)]"
}

if { $bw13 > $holdseq2 } {
    puts $f10 "$now [expr ($bw12 - $holdtime2)/($bw13 - $holdseq2)]"
} else {
    puts $f10 "$now [expr ($bw13 - $holdseq2)]"
}

if { $bw15 > $holdseq3 } {
    puts $f11 "$now [expr ($bw14 - $holdtime3)/($bw15 - $holdseq3)]"
} else {
    puts $f11 "$now [expr ($bw15 - $holdseq3)]"
}

# Reset Variables

$sink set bytes_ 0
$sink2 set bytes_ 0
$sink3 set bytes_ 0
$sink4 set bytes_ 0

$sink set nlost_ 0
$sink2 set nlost_ 0
$sink3 set nlost_ 0
$sink4 set nlost_ 0

set holdtime $bw8
set holdseq $bw9
set holdrate1 $bw0
set holdrate2 $bw1
set holdrate3 $bw2
set holdrate4 $bw3

$ns at [expr $now+$time] "record" ;# Schedule Record after $time interval sec
}
# Start Recording at Time 0

$ns at 0.0 "record"

```

```
$ns at 1.10 "$app1 start"           ;# Start transmission at time t = 0.10 Sec
$ns at 1.20 "$app2 start"           ;# Start transmission at time t = 8 Sec
$ns at 1.80 "$app3 start"           ;# Start transmission at time t = 10.20 Sec
$ns at 1.99 "$app4 start"           ;# Start transmission at time t = 29 Sec

# Stop Simulation at Time 20 sec
$ns at 20.0 "stop"
# Reset Nodes at time 20 sec
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns at 20.0 "$node_($i) reset";
}
# Exit Simulatoion at Time 20.01 sec
$ns at 20.01 "puts \"NS EXITING...\" ; $ns halt"
proc stop {} {

    global ns_ tracefd f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11

    # Close Trace Files
    close $f0
    close $f1
    close $f2
    close $f3
    close $f4
    close $f5
    close $f6
    close $f7
    close $f8
    close $f9
    close $f10
    close $f11
    exec nam sim12.nam &
    # Plot Recorded Statistics

    exec xgraph Debit1.tr Debit2.tr Debit3.tr Debit4.tr -geometry 800x400 &
    exec xgraph lost1.tr lost2.tr lost3.tr lost4.tr -geometry 800x400 &
    exec xgraph delay1.tr delay2.tr delay3.tr delay4.tr -geometry 800x400 &

    # Reset Trace File
    $ns_ flush-trace
    $ns_ flush-tracefile
    close $tracefd

    exit 0
}

puts "Starting Simulation..."
$ns run
```

Bibliographie

[1] : Jonathan Petit « *Surcoût de l'authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires* », Thèse de Doctorat de l'Université de Toulouse Discipline : Informatique, soutenue Le 13 Juillet 2011,167pages.

[2] : Boulkamh Chouaib, « *Prise en Compte de la QoS par les Protocoles de Routage dans les Réseaux Mobiles Ad Hoc* », Mémoire de Magistère de l'Université El Hadj Lakhdar de Batna, Discipline : Informatique Industrielle, Soutenue 2008, 112 pages.

[3] : Florent Kaisser, « *Communications dans les réseaux fortement dynamiques* », Thèse de Doctorat de l'Université de Paris Sud - Paris XI Ecole Doctorale Sciences et Technologies de l'Information des Télécommunications et des Systèmes, Spécialité : Physique, Soutenue le 21 juin 2010, 180 pages.

[4] : Boussad AIT-SALEM, « *Sécurisation des Réseaux Ad hoc : Systèmes de Confiance et de Détection de Répliques* », Thèse de Doctorat de l'Université de Limoges Spécialité : Informatique, Soutenue le 12/07/2011,181 pages.

[5] : Messaoud BELLOULA, « *La géolocalisation dans les réseaux de capteurs sans fils* », Mémoire de Magistère de l'Université El Hadj Lakhdar de Batna, Spécialité : Ingénierie des Systèmes Informatiques (ISI), 95pages.

[6] : James Bernsen, and D. Manivannan, « *Unicast Routing Protocols for Vehicular Ad Hoc Networks : A Critical Comparison and Classification*», Elsevier Journal of Pervasive and Mobile Computing, vol. 5, pp. 1-18, 2009.

[7] : Vehicle Infrastructure Integration – VII. <http://www.its.dot.gov/vii/>. Visité le 19 Mars 2017 à 10h.

[8] : Olivier Rivaton, « *Le routage de l'information dans les réseaux véhiculaires mobiles* », Mémoire de l'Université LAVAL Québec, Canada, 2016, 124pages.

[9] : Bouguer Safia, « *Etude et simulation comparative entre les réseaux de capteurs sans fils traditionnels et les réseaux de capteurs véhiculaires* » Mémoire d'Ingénieur del' Université Abou Bekr Belkaid Tlemcen, Options Système de Télécommunication, Soutenu en Juin 2012, 81pages.

- [10] : Farah El Ali, « *Communication unicast dans les réseaux mobiles dynamiques* », Thèse de Doctorat de l'Université de Technologie de Compiègne, Spécialité : Génie Informatique, Soutenu le 04 décembre 2012,137pages.
- [11] : Ayoub Benchabana et Ramla Bensaci, « *Analyse des protocoles de routage dans les réseaux VANET* », Mémoire Master Académique de l'Université Kasdi Merbah-Ouargla, Spécialité : Informatique Industrielle, Soutenu le 12/06/2014,107pages.
- [12] : GRICH Sofiane, « *Contribution à la Qualité de Service dans les réseaux VANET* », Mémoire de Magistère de l'Université d'Oran, Spécialité : Ingénierie des Systèmes Complexes et Multimédia, Soutenu le 04 Novembre 2015,80pages.
- [13] : Ait Ali Kahina, « *Modélisation Et Etude De Performances Dans Les Réseaux VANET* », Thèse de doctorat de l'Université de Technologie de Belfort-Montbéliard, Spécialité : Informatique, Soutenu le Mardi 16 Octobre 2012,198pages.
- [14] : Nadia Haddadou, « *Réseaux Ad Hoc véhiculaires : vers une dissémination de données efficace, coopérative et fiable* », Thèse de doctorat de l'Université Paris-Est, Discipline : Informatique, Soutenu le 16 Juin 2014,134pages.
- [15] : Talar Atéchian, « *Protocole de routage géo-multipoint hybride et mécanisme d'acheminement de données pour les réseaux ad hoc de véhicules (VANETs)* », Thèse de doctorat de Institut National des Sciences Appliquées de Lyon, Spécialité : Informatique et Mathématiques, Soutenu le vendredi 24 septembre 2010,166pages.
- [16] : Meraihi Yassine, « *Routage Dans Les Reseaux Vehiculaires (Vanet) Cas D'un Environnement Type Ville* », Mémoire de Magister En Genie Electrique Universite M'Hamed Bougara – Boumerdes, Soutenu 2011,122pages.
- [17] : Ba_Amadou Adama, « *Protocole de routage basé sur des passerelles mobiles pour un accès Internet dans les réseaux véhiculaires* », Thèse de doctorat de l'université de Montréal, Spécialité : Informatique, Avril, 2011,68pages.
- [18] : Sana Ajmal, Asim Rasheed and Aamir Hasan, « *Classification of VANET MAC, Routing and Approaches A Detailed Survey* », Journal of Universal Computer Science, vol. 20, no. 4 (2014), 462-487.

[19] : Mohammed ERRITALI, « *Contribution à la sécurisation des réseaux ad hoc véhiculaires* », Thèse de doctorat de l'université de Mohammed V –Agdal Faculté Des Sciences Rabat, Spécialité : Informatique, Soutenue le : 10 Octobre 2013,137pages.

[20] : Salim ALLAL, « *Optimisation des échanges dans le routage géocast pour les réseaux de Véhicules Ad Hoc VANETs* », Thèse de doctorat de l'Université de PARIS NORD 13, Spécialité : Réseaux et Techniques Informatiques, Soutenue le 10 Décembre 2014,145pages.

[21] : Ignacy GAWĘDZKI, « *Algorithmes distribués pour La sécurité et la qualité de service dans les réseaux ad hoc mobiles* », Thèse de doctorat de l'Université de Paris-Sud 11, Spécialité informatique, Soutenue le 29 septembre 2008,145pages.

[22] : Jean Carle, Olivier FLAUZAC, Bachar Salim HAGGAR, Florent, « *État de l'art sur les protocoles de routage dans les réseaux ad hoc* », Rapport RISC-SP3.1-#6.ed1, 31/07/2008.

[23] : ABDELLAOUI, Rachid, « *SU-OLSR une nouvelle solution pour la sécurité du protocole OLSR* », Thèse de Maîtrise ès sciences (M. Sc.) en génie concentration réseaux de télécommunications, Ecole de Technologie Supérieure Université du Québec Montréal, Soutenue le 05 MAI 2009,126pages.

[24] : Uma Nagaraj, Dr. M. U. Kharat and Poonam Dhamal, « *Study of Various Routing Protocols in VANET* », International Journal of Computer Science & Technology, IJCST Vol. 2, Issue 4, Oct. - Dec. 2011

[25] : Lionel Barrère, « *Étude et proposition de services dans les réseaux mobiles militaires de type MANet* », », Thèse de doctorat de l'Université de Bordeaux I, Spécialité : Informatique, Soutenue le 9 juillet 2009,1pages.42

[26] : Moez JERBI, « *Protocoles pour les communications dans les réseaux de véhicules en environnement urbain : Routage et GeoCast basés sur les intersections* », Thèse de doctorat de l'Université de D'evry Val D'essonne, Spécialité : Reseaux et Informatique, Soutenue le 06 Novembre 2008,150pages.

[27] : Lylia Alouache, Nga Nguyen, Makhlouf Aliouat, Rachid Chelouah, « *New robust protocol for IoV communications* », 2017 ISTE OpenScience – Published by ISTE Ltd, London.

- [28] : MOHAMED BOUARIR, « *Protocole de routage intelligent pour les réseaux ad hoc de véhicules* », Mémoire de l'Université de Québec En Abitibi-Témiscamingue, Soutenu Octobre 2012,138pages.
- [29] : KAHINA MOGHRAOUI, « *Gestion de l'anonymat des communications dans les réseaux véhiculaires ad hoc sans fil (vanets)* », Mémoire présenté à l'Université du Québec à Trois-Rivières, Soutenu Juillet 2015.
- [30] : Moez Jerbi, Sidi-Mohammed Senouci, Rabah Meraihi and Yacine Ghamri-Doudane, « *An Improved Vehicular Ad Hoc Routing Protocol for City Environments* », IEEE Communications Society subject matter, publication 07/ 2007.
- [31] : Bijan Paul Md, Md. Ibrahim and Abu Naser Bikas, « *VANET Routing Protocols : Pros and Cons* », International Journal of Computer Applications (0975 – 8887), Volume 20–No.3, April 2011.
- [32] : Pei-Chun Cheng, Kevin C, Lee Mario Gerla, Jérôme Härrri, « *GeoDTN+Nav : Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments* », Mobile Netw Appl DOI 10.1007/s11036-009-0181-6, Published 17 June 2009.
- [33] : POONAM DHAMAL et all, « *Broadcasting Routing Protocols in VANET* », Oriental Journal Of Computer Science & Technology, Vol. 4(2), 393-398 (2011).
- [34] : Suresh Kumar et all, « *Qualitative Based Comparison of Routing Protocols for VANET* », Journal of Information Engineering and Applications, Vol 1, No.4, 2011.
- [35] : Nouredine CHAIB, « *La sécurité des communications dans les réseaux VANET* » Mémoire Magister en Informatique, Option : Ingénierie des systèmes informatiques (ISI).
- [36] : Ahizoune Ahmed, « *Un protocole de diffusion des messages dans les réseaux véhiculaires* » Thèse de Maîtrise ès sciences (M. Sc.) de l'Université de Montréal,Soutenu Avril 2011,88pages.
- [37] : YUCEF ZIANI, « *Etude comparative de Méthodes de routage dans les réseaux de capteurs sans fil pour le domaine résidentiel* », Mémoire présenté à l'Université du Québec à Trois-Rivières, Juin 2013.

- [38] : Mariam Dawoud, « *Analyse du protocole AODV* », DEA d'Informatique de l'Université Paul Sabatier – I.R.I.T, 2006.
- [39] : DRICI Hanane et CHIBANI Chahinez, « *Impact des Générateur de Mobilité Sur Les Performances Des Réseaux Véhiculaire VANETs* », Mémoire de Master 2 de l'Université Dr. Tahar Moulay de Saïda, Soutenu le 15/ 06 / 2015,88pages.
- [40] : BOUZEBIBA Hadjer et BOUIZEM Yasmina, « *Impact des modèles de mobilités sur les performances des protocoles de routage en milieu urbain réaliste dans les réseaux VANET (V2V)* », Mémoire de Master 2 de l'Université Abou Bakr Belkaid–Tlemcen, Option : Réseaux et Systèmes Distribués (R.S.D), Soutenu Juin 2015,62pages.
- [41] : Nadeen SALAMEH, « *Conception d'un système d'alerte embarqué basé sur les communications entre véhicules* », Thèse de doctorat de l'Institut National des Sciences Appliquées de Rouen, Discipline : Informatique, soutenue le 4 Novembre 2011,143pages.
- [42] : Teerawat Issariyakul and Ekram Hossain « *Introduction to Network Simulator NS2* » Edition 2, édition Springer New York Dordrecht Heidelberg London, 2012,535pages.
- [43] : Mohamed djihad BEN SALEM et Oussama BOUGOFFA, « *Etude comparative de deux simulateurs pour les réseaux ad-hoc sans fil* », Mémoire de Master en informatique de l'Université Kasdi Merbah Ouargla, Spécialité : Informatique Industriel, Soutenu 14/06/2014,84pages.