

UNIVERSITÉ AMMAR THLEDJI LAGHOUAT

Faculté de Science
Département d'Informatique



MÉMOIRE DE MASTER

Pour obtenir diplôme Master Réseaux, Systèmes et Applications Réparties

Identification des facteurs les plus influents sur l'opération de routage dans les réseaux véhiculaires

Présenté par :

BENADJILA SOUAD

Promoteurs :

M. GUELLOUMA YOUNES

M. KERRACHE CHAKER

ABDELAZIZ

Soutenu devant le jury composé de :

M. ALLAoui Tahar

Président

M. BOUAKKAZ Mustapha

Examineur

2016/2017

ملخص

تشكل شبكات المركبات (*VANET*) حاليا موضوع اهتمام متزايد من جانب المصنعين والباحثين. فالهدف الرئيسي من هذه الشبكة هو تحسين السلامة على الطرق، ظروف حركة المرور، وتوفير تطبيقات الدعاية و الترفيه للسائقين والركاب. لهذا، من المهم تبادل المعلومات بالطريقة الأكثر فعالية بين مختلف المركبات. قد تتأثر هذه العملية بالبيئة المحيطة أو حتى سيناريو التنقل مثل كثافة الشبكة، نموذج الانتشار، خارطة طريق المحاكاة، الخ. في هذه الحالة، نحتاج إلى تحديد أهم العوامل ذات الصلة لتصميم بروتوكول توجيه موثوق وفعال في أي بيئة وسيناريو محتمل لتسهيل التصميم وتقليل وقت المحاكاة. لهذا، تم تصميم طريقة لتحليل العوامل 2^k التي تسمح لنا باستخراج تأثير كل عامل و تأثير مزيج كل منها. نتائج المحاكاة التي تم الحصول عليها باستخدام جهاز محاكاة $NS-2$ و تنفيذ تطبيق 2^k ، أظهرت أن بروتوكول التوجيه ثم كثافة الشبكة هي العوامل الأكثر تأثيرا على توجيه المعلومات في شبكة ذات كثافة 50 الى 300 مركبة، نموذج انتشار *FreeSpace* و *TwoRayGround* ، بروتوكول توجيه *AODV* و *GPSR* في خارطة المدينة وخارطة الطريق السريع.

Résumé

Les réseaux ad-hoc véhiculaires (VANET) font actuellement l'objet d'une attention accrue de la part des constructeurs et des chercheurs. Le principal objectif de ce type de réseau est d'améliorer la sécurité routière, les conditions de circulations, et d'apporter aux conducteurs et aux passagers des applications publicitaires ou de divertissement. Pour cela, il est important de faire circuler l'information de la manière la plus efficace possible entre les différents véhicules. Cette opération peut être affectée par l'environnement ou même par le scénario de mobilité comme la densité du réseau, le modèle de propagation, le roadmap simulé, etc. Dans ce cas, on a besoin d'identifier les facteurs les plus pertinents pour concevoir un protocole de routage fiable et efficace dans tout environnement et scénario éventuel afin de faciliter cette conception et réduire le temps de simulation. Pour cela, une méthode de l'analyse factorielle 2^k a été conçue qui nous permet d'extraire l'influence de chaque facteur et celle de leur combinaison. Les résultats de simulation obtenus à l'aide du simulateur NS-2 et de l'application 2^k , ont montré que le protocole de routage, puis la densité du réseau sont les facteurs le plus influent sur le routage dans des scénarios de densité de 50 à 300 véhicules, modèle de propagation Free Space et Two Ray Ground, protocole de routage AODV et GPSR et un roadmap de ville et autoroute.

Abstract

Vehicular ad hoc networks (VANETs) are currently receiving increased attention from manufacturers and researchers. The main objective of this type of network is to improve road safety, driving conditions, and to provide drivers and passengers with advertising and entertainment applications. To do this, it is important to circulate the information in the most efficient way possible between the different vehicles. This operation can be affected by the environment or even by the mobility scenario such as network density, propagation model, simulated roadmap, etc. In this case, it is necessary to identify the most relevant factors for designing a reliable and efficient routing protocol in any environment and possible scenario in order to facilitate this design and reduce the simulation time. For this purpose, a method of factor analysis 2^k has been devised which allows us to extract the influence of each factor and that of their combination. The simulation results obtained using the NS-2 simulator and the 2^k application showed that the routing protocol then network's density are the most influential factors on routing in density's scenarios of 50 to 300 vehicles, Free Space and Two Ray Ground propagation model, AODV and GPSR routing protocol, and city and highway roadmap.

Remerciement

D'abord, merci Allah de m'avoir donné la force pour finir ce modeste travail.

J'aimerais remercier tous ceux qui m'ont accompagné tout au long de la réalisation de ce mémoire. Dans un premier temps, mes encadrants, M. GUELLOUMA Younes et M. KERRACHE Chaker Abdelaziz d'avoir dirigé mon mémoire, qui ont su me guider, m'accompagner et être toujours disponible lorsque j'en avait besoin.

J'adresse mes remerciements aux membres du Jury d'avoir accepté d'examiner mon travail.

Je remercie aussi M. LAGRAA Nasreddine, pour leurs remarques et suggestions qui m'ont été très utiles et fructueuses, pour la finalisation de ce mémoire. Je profite de l'occasion pour leur adresser mes sincères respects pour leurs compétences, leurs conseils avisés, et leurs aides durant mes études.

Ensuite, j'aimerais remercier également tous les enseignants du département de mathématiques et informatique qui nous ont enseigné sans hésiter à nous faire passer leurs informations.

Il est bon et juste d'évoquer l'appui moral ainsi que la sollicitude trouvée auprès de toute ma famille. Je tiens à exprimer mes sentiments les plus respectueux et ma profonde reconnaissance à mes très chers parents, à mes sœurs, pour les encouragements constants qu'ils ont déployé tout au long des années de mes études.

Enfin, je remercie toutes les personnes qui, de près ou de loin, ont apporté leur contribution à ce travail. Je leur exprime ici toute ma reconnaissance et ma sympathie.

À mes très chers parents

Ma très chère mère

Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé d'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

Mon très cher père

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

À mes très chères sœurs Hadjer, Soumia et Houda en témoignage de l'attachement, de l'amour et de l'affection que je porte pour vous.

À mes chères amies Amel et Bouchra.

Et à tous ceux qui m'aiment.

Table des matières

Résumé	iii
Remerciement	v
1 Introduction générale	2
1.1 Contexte et problématique	2
1.2 Objectifs	3
1.3 Organisation du mémoire	3
2 Les réseaux véhiculaires : vue générale	4
2.1 Introduction	4
2.2 Architectures des réseaux VANETs	4
2.2.1 Communication véhicule à véhicule (V2V)	4
2.2.2 Communication véhicule à infrastructure (V2I)	5
2.2.3 Communication hybride	6
2.3 Caractéristiques des réseaux véhiculaires	6
2.4 Types d'applications	7
2.4.1 Applications de sécurité	7
2.4.2 Applications de gestion de trafic	7
2.4.3 Applications de confort	7
2.5 Les défis des réseaux véhiculaires	8
2.5.1 Défis de sécurité	8
2.5.2 Défis du réseau	8
2.5.3 Défis techniques	9
2.5.4 Défi commun : livraison de données	9
2.6 Les protocoles de routage dans les réseaux VANETs	10
2.7 Conclusion	13
3 L'analyse factorielle	14
3.1 Introduction	14
3.2 Définition	14
3.3 Caractéristiques de l'analyse factorielle	15

3.4	Les types de l'analyse factorielle	15
3.4.1	L'analyse factorielle exploratoire (AFE)	15
3.4.2	L'analyse factorielle confirmatoire (AFC)	16
3.5	Les méthodes de l'analyse factorielle	16
3.5.1	Analyse en composantes principales (ACP)	16
3.5.2	Analyse factorielle des correspondances (AFC)	16
3.5.3	Analyse factorielle multiple (AFM)	17
3.5.4	Analyse factorielle de données mixtes (AFDM)	18
3.5.5	Analyse factorielle multiple hiérarchique (AFMH)	18
3.6	L'analyse discriminante	18
3.6.1	Approche descriptive	19
3.6.2	Approche prédictive	19
3.6.2.1	Analyse discriminante linéaire	19
3.7	La régression	19
3.7.1	La régression linéaire	20
3.8	La méthode χ^2	20
3.9	Conclusion	22
4	Protocoles de routage et outil d'identification des paramètres les plus pertinents (basé sur 2^k)	23
4.1	Introduction	23
4.2	Ad hoc On-demand Distance Vector (AODV)	23
4.3	Greedy Perimeter Stateless Routing (GPSR)	26
4.3.1	mode glouton (greedy mode)	26
4.3.2	mode de périmètre (perimeter mode)	27
4.4	Intelligent Routing protocol using real time Traffic Information in urban Vehicular environment (IRTIV)	29
4.4.1	Estimation du trafic en temps réel	30
4.4.2	sélection du chemin	31
4.4.3	Mécanisme de livraison de données	32
4.5	Greedy Traffic-Aware Routing (GyTAR)	32
4.5.1	Sélection de jonction	33
4.5.2	Routage de données entre deux jonctions	33
4.5.3	Stratégie de rétablissement	34
4.6	Spatial and Traffic-Aware Routing (STAR)	34
4.6.1	Surveillance du trafic	35
4.6.2	Routage des paquets	36
4.7	Comparaison entre les protocoles étudiés	37

4.8	L'outil d'identification des facteurs (méthode 2^k)	38
4.8.1	Les facteurs définis dans la littérature du VANET	38
4.8.1.1	Densité du réseau	39
4.8.1.2	Vitesse des véhicules	39
4.8.1.3	Largeur du canal	39
4.8.1.4	Schéma de diffusion	39
4.8.1.5	Priorité des messages	39
4.8.1.6	Périodicité des messages	40
4.8.1.7	Modèle de mobilité	40
4.8.1.8	Modèle de propagation	40
4.8.1.9	Roadmap simulé	40
4.8.2	L'analyse factorielle 2^k	40
4.8.2.1	Exemple applicatif	41
4.9	Conclusion	45
5	Résultats et discussion	46
5.1	Introduction	46
5.2	L'outil de simulation	46
5.3	Modélisation de la simulation et scénarios	47
5.4	Résultats de simulation et analyse de performances	48
5.4.1	Délai moyen de bout en bout (End-to-End Delay)	48
5.4.2	Taux de délivrance des paquets (Packets Delivery Ratio, PDR)	49
5.5	Application 2^k	50
5.5.1	Délai moyen de bout en bout (End-to-End Delay)	52
5.5.2	Taux de délivrance des paquets (Packets Delivery Ratio, PDR)	52
5.6	Conclusion	53
6	Conclusion générale	55
	Bibliographie	57
A	annexe	59
A.1	Le simulateur NS-2	59
A.1.1	Définition	59
A.1.2	L'architecture de simulation	59
A.1.3	les étapes de simulation	60
A.1.4	Réalisation d'une simulation	60

Table des figures

2.1	Architectures des réseaux VANETs	5
2.2	Les protocoles de routage unicast dans les réseaux VANETs	12
3.1	Droite de régression	20
4.1	Les protocoles de routage unicast choisis pour l'étude	24
4.2	Routage glouton du protocole GPSR	26
4.3	Cas d'échec de routage glouton du protocole GPSR	27
4.4	La règle de droite (l'intérieur du triangle)	28
4.5	Sélection du chemin avec le protocole IRTIV	31
5.1	Délai moyen de bout en bout en fonction de la densité dans une zone urbaine	48
5.2	Délai moyen de bout en bout en fonction de la densité dans une autoroute	48
5.3	Taux de délivrance des paquets en fonction de la densité dans une zone urbaine en modèle Free Space	50
5.4	Taux de délivrance des paquets en fonction de la densité dans une zone urbaine en modèle Two Ray Ground	50
5.5	Taux de délivrance des paquets en fonction de la densité dans une autoroute en modèle Free Space	51
5.6	Taux de délivrance des paquets en fonction de la densité dans une autoroute en modèle Two Ray Ground	51
5.7	Résultats de l'application 2^k sur les observations de délai moyen de bout en bout	53
5.8	Résultats de l'application 2^k sur les observations de PDR	54
A.1	L'architecture de simulation avec NS-2	60
A.2	Les étapes de simulation	61

Liste des tableaux

4.1	Champs additionnels dans le paquet Hello modifié	30
4.2	La signification des flags Lj et Rj	30
4.3	Comparaison entre les protocoles de routage étudiés	38
4.4	Les expériences définies par la conception 2^2	41
4.5	Exemple de résultats obtenus en termes de temps de notification d'avertissement variant deux facteurs.	41
4.6	Tableau des signes pour calculer les effets des facteurs dans une conception 2^2	42
5.1	Paramètres de simulation	47
5.2	Les valeurs des facteurs considérés	50
5.3	Les observations	51

Liste d'abréviations

VANET	V ehicular A d hoc N ETwork
MANET	M obile A d hoc N ETwork
RSU	R oad S ide U nit
V2V	V ehicle to V ehicle
WiFi	W ireless F idelity
V2I	V ehicle to I nfrastructure
QOS	Q uality O f S ervice
GLS	G rid L ocation S ervice
DTN	D elay T olerent N etwork
AfE	A nalyse F actorielle E xploratoire
AfC	A nalyse F actorielle C onfirmatoire
ACP	A nalyse en C omposantes P incipales
AfC	A nalyse F actorielle des C orrespondances
AfCM	A nalyse F actorielle des C orrespondances M ultiples
AfM	A nalyse F actorielle M ultiple
AfDM	A nalyse F actorielle de D onnées M ixtes
AfMH	A nalyse F actorielle M ultiple H iéarchique
AODV	A d hoc O n-demand D istance V ector
RREQ	R oute RE quest
RREP	R oute RE ply
RERR	R oute ERR or
GPSR	G reedy P erimeter S tateless R outing
RNG	R elative N eighborhood G raph
GG	G abriel G raph
IRTIV	I ntelligent R outing protocol using real time T raffic I nformation in urban V ehicular environment
DLN	D irect L eft N eighbors
DRN	D irect R ight N eighbors
VL	V ehicles L eft
VR	V ehicles R ight
Lj	L eft j unction
Rj	R ight j unction

GyTAR	G reedy T raffic- A ware R outing
STAR	S patial and T raffic- A ware R outing
PA	P oint d'Ancrege
Tbit	T raffic b it
ASbit	A lready S ent b it
TTL	T ime T o L ive
PRV	P Resence V ector
PEV	P ERsistence V ector
AC	A ccess C lass
SST	S um of S quares T otal
NS-2	N etwork S imulator 2
OTCL	O bject T ools C ommand L anguage
TCL	T ools C ommand L anguage
C4R	C ity M ob f or R oadmaps
OSM	O pen S treet M ap
PDR	P ackets D elivery R atio

Chapitre 1

Introduction générale

1.1 Contexte et problématique

Le nombre de véhicules en circulation est de plus en plus important. Nous avons dépassé le milliard de voitures sur les routes, et l'augmentation est en moyenne de 35 millions par an [1]. Liés à cette croissance constante, les accidents de la route font partie des dix principales causes de mortalité dans le monde. Afin de réagir à cette situation, l'amélioration de la sécurité routière est devenue une préoccupation. C'est dans cet esprit que sont apparues un certain nombre de recherches qui visent non seulement à réduire le nombre de morts sur les routes et à améliorer les conditions de la circulation, mais aussi à diminuer les embouteillages et la pollution. Les réseaux mobiles véhiculaires, mieux connus sous le nom de Vehicular Ad-hoc NETwork (VANET), se sont fixés ces objectifs. Ce domaine de recherche intéresse aussi bien les universitaires que les industriels de l'automobile et les grandes entreprises de l'innovation (comme Google), qui travaillent actuellement sur trois projets majeurs [1] :

- Assistance à la conduite (self-parking, aide au dépassement, radar anti-collision, régulateur de vitesse, etc.) ;
- réseaux de communication entre les véhicules ;
- voitures autonomes.

Ce sont les réseaux de communications entre les véhicules et plus particulièrement le routage de l'information qui nous intéressent dans ce mémoire. Pour acheminer les informations d'un véhicule à un autre à travers un réseau composé de beaucoup de véhicules, se déplaçant à des vitesses atteignant 180 km/h, il est nécessaire d'effectuer un routage rapide et efficace de l'information.

Le routage de l'information dans les réseaux véhiculaires mobiles n'est pas un problème simple. En effet, les VANETs sont des réseaux spontanés qui doivent être en mesure de se mettre en place de façon autonome. De plus, les VANETs sont utilisés pour l'amélioration de la sécurité routière, c'est-à-dire pour la prévention et la diminution des accidents de la route. Dans ce contexte, les messages qui transitent sur le réseau se doivent d'être délivrés dans des délais courts et sans perte. Afin de répondre à ces problèmes, les chercheurs ont

développé un certain nombre de méthodes de routage de l'information. Toutefois, la plupart de ces protocoles ne prennent pas en compte les différentes spécificités qu'implique un réseau véhiculaire mobile et ne sont souvent efficaces que dans des situations particulières. En effet, un VANET doit être efficace en ville, en campagne et sur autoroute, dans des situations d'embouteillages, de circulation fluide, mais aussi dans les situations d'urgences.

1.2 Objectifs

Dans ce mémoire, nous nous sommes focalisés à l'étude du problème de routage dans les réseaux véhiculaires. Notre objectif est de proposer une solution adaptée à la conception d'un protocole de routage fiable et efficace pour délivrer au mieux les données. Cette solution est basée sur une méthode statistique de l'analyse factorielle, elle permet d'identifier les paramètres les plus influents pouvant exister dans un réseau de véhicules, ainsi réduire le temps de simulation requis.

1.3 Organisation du mémoire

Le mémoire est organisé comme suit :

Dans le chapitre 2, nous introduisons les concepts des réseaux VANETs. Nous décrivons les architectures et les caractéristiques. Nous décrivons aussi ses applications et les différents défis de ces réseaux, ainsi qu'une vue générale sur les protocoles de routage proposés.

Le chapitre 3 est consacré principalement à l'analyse factorielle. Tout d'abord, nous définissons l'analyse factorielle, ainsi nous présentons ses caractéristiques, ses types, et ses méthodes en décrivant chacune pour choisir la plus apte pour atteindre notre objectif.

Ensuite, le chapitre 4 présente les protocoles qui ont été choisis pour étudier leur performance à l'existence de certains paramètres, et l'outil permettant de déterminer quels sont les plus influents.

Dans le chapitre 5, on va discuter les résultats obtenus après la simulation des protocoles choisis, et appliquer sur lesquels la méthode 2^k .

Finalement, on termine par une conclusion résumant notre travail et résultats.

Chapitre 2

Les réseaux véhiculaires : vue générale

2.1 Introduction

L'évolution récente des systèmes de communications sans fil a conduit à la création d'un nouveau type de réseaux appelés réseaux mobiles véhiculaires (VANET) qui est une classe émergente de réseaux sans fil et surtout une classe spéciale des réseaux ad-hoc mobiles (MANET). L'objectif de ces réseaux est de réduire les délais en utilisant la communication de véhicule à véhicule et d'augmenter la couverture du réseau de l'opérateur tout en réduisant la charge du réseau. Les VANETs se constituent à partir d'un ensemble d'entités communicantes, composé de véhicules et d'unités de bords de route (RSU). Grâce aux différentes applications que supportent les VANETs, ces réseaux sont considérés comme étant le moyen le moins cher et le plus efficace pour éviter les embouteillages, minimiser la consommation de carburant et réduire le temps passé sur les routes.

Ce chapitre a pour objectif de donner une vue d'ensemble des VANETs. Nous présentons dans un premier temps leurs architectures, ainsi que ces caractéristiques. Nous évoquerons, par la suite, les différents types d'applications qu'ils peuvent proposer, les défis, et enfin les protocoles de routage utilisés dans ces réseaux.

2.2 Architectures des réseaux VANETs

Les réseaux véhiculaires combinent différents types de communication distingués par les services proposés dans ces réseaux, la figure 2.1 représente ces types :

2.2.1 Communication véhicule à véhicule (V2V)

Les véhicules constituent un réseau ad-hoc afin d'échanger des informations sur l'état de la route pour éviter des accidents ou réduire le chemin parcouru pour une meilleure utilisation des ressources comme le temps et le carburant.

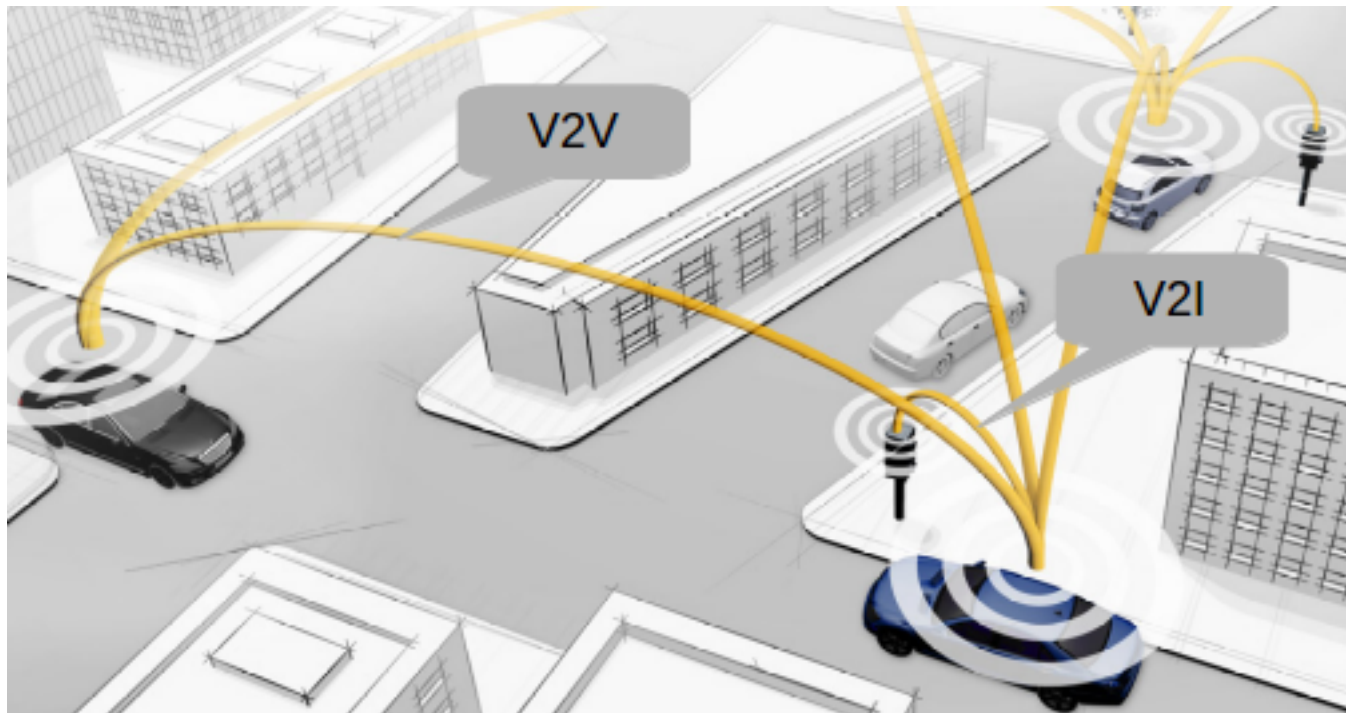


FIGURE 2.1 – Architectures des réseaux VANETs

Le réseau entre les véhicules est réalisé en utilisant la technologie sans fil WiFi. Cette architecture de communication peut être utilisée dans les scénarios de diffusion d’alerte (freinage d’urgence, collision, ralentissement, etc.) ou pour la conduite coopérative¹. Ces réseaux sont de nature totalement distribuée et dynamique, c’est-à-dire chaque nœud doit être capable de s’auto-configurer sans la nécessité d’aucune gestion centralisée, ni d’aucune infrastructure préalablement déployée[2]. Les VANETs n’ont aucune infrastructure fixe et ils s’appuient plutôt sur les nœuds ordinaires (les véhicules) pour effectuer le routage des messages et des fonctions de gestion de réseau. Les véhicules peuvent communiquer entre eux, en collectant les informations depuis leurs capteurs, qui peuvent être utiles pour notifier d’autres véhicules sur la condition de la route.

2.2.2 Communication véhicule à infrastructure (V2I)

Dans cette catégorie, on ne se concentre pas seulement sur de simples systèmes de communications inter-véhicules mais aussi sur ceux qui utilisent des stations de bases ou points d’infrastructure (RSU) d’où la tendance de relier les réseaux de véhicules à l’internet afin de fournir une grande gamme de services, pour le passager, le constructeur automobile ou l’exploitant d’infrastructure.

1. la conduite coopérative repose sur la possibilité des échanges d’informations entre le véhicule et son environnement de conduite.

Cette approche repose sur le modèle client/serveur où les véhicules sont les clients et les stations installées le long de la route sont les serveurs. Ces serveurs sont connectés entre eux via une interface filaire ou sans fil. Toute communication doit passer par eux. Ils peuvent aussi offrir aux utilisateurs plusieurs services : localisation des stations d'essence et emplacements de parking libre, le chat inter-véhicule, informations climatiques, informations culturelles, échange de données de voiture-à-domicile et même la communication de voiture à garage pour le diagnostic distant[2].

L'inconvénient majeur de cette approche est que l'installation des stations le long des routes est une tâche coûteuse et prend beaucoup de temps, sans oublier les coûts relatifs à la maintenance des stations.

2.2.3 Communication hybride

La combinaison des communications V2V avec les communications V2I permet d'accomplir une communication hybride très intéressante. Ce mode permet de répondre à la problématique de connectivité de longue distance dans le mode V2V. Les infrastructures, servent à leur tour de relais fixes afin d'étendre la distance de communication inter-véhicules. En revanche, les portées des infrastructures (RSU) étant limitées, l'utilisation des véhicules comme relais permet d'étendre cette distance. Dans un but économique et afin d'éviter la multiplication des stations de bases à chaque coin de rue, l'utilisation des sauts par véhicules intermédiaires prend tout son importance.

2.3 Caractéristiques des réseaux véhiculaires

Les réseaux VANETs ont des caractéristiques semblables à ceux des réseaux ad-hoc mobiles. Afin d'analyser ces réseaux il est important d'étudier leurs caractéristiques spécifiques, qui peuvent être résumées en :

- Mobilité rapide : les véhicules déplacent habituellement à grande vitesse, cela rend plus difficile de prédire la position d'un nœud et de protéger sa confidentialité ;
- topologie du réseau : due à la mobilité rapide et la vitesse aléatoire des véhicules, la position d'un nœud change tout le temps, ainsi la topologie du réseau change fréquemment. La topologie du réseau pourrait être affectée par la réaction des conducteurs après la réception de messages. Cela signifie que le contenu des messages peut changer la topologie du réseau ;
- taille du réseau illimitée : VANET peut être mis en œuvre pour une ville, plusieurs villes ou pour des pays. Cela signifie que la taille du réseau dans VANET est géographiquement non bornée[1] ;

- échange fréquent d'informations : la nature ad-hoc de VANET motive les nœuds à rassembler des informations auprès des autres véhicules et des unités routières. L'échange d'informations entre nœuds devient donc fréquent[3] ;
- temps critique : les informations doivent être transmises aux nœuds avec une limite de temps afin qu'une décision puisse être prise par le nœud et effectuer une action en conséquence ;
- énergie illimitée : les VANETs utilisent la puissance des voitures donc ils n'ont aucun problème de ressources d'énergie et de calcul. Cela permet à ce type de réseau d'utiliser des techniques exigeantes et fournit également une puissance de transmission illimitée.

2.4 Types d'applications

Diverses applications sont offertes dans les réseaux véhiculaires, celles-ci peuvent être classées en trois (3) catégories :

2.4.1 Applications de sécurité

Ces applications augmentent la sécurité des véhicules et des passagers sur les routes en améliorant les conditions de conduite et réduisant les risques d'accidents. Elles fournissent divers services comme la violation de signal de circulation, vidéo d'urgence en streaming, les feux de freinage d'urgence, pré-détection de collision, la prudence de collision, la prudence de la rue, etc. Ces applications doivent être rapides et fiables. Elles doivent être en mesure de faire suivre l'information à tous les véhicules dans une certaine zone.

2.4.2 Applications de gestion de trafic

Les applications de gestion du trafic routier ont pour objectif d'apporter au conducteur des informations concernant le trafic routier autour de lui ou dans un endroit précis. Ce sont des applications de planification permettant d'emprunter le trajet le plus court vers la destination du véhicule, en prenant en compte le trafic, les différentes routes ainsi que les obstacles ou accidents potentiellement présents sur le trajet. Ainsi pour apporter une assistance au niveau des intersections[1].

2.4.3 Applications de confort

Les applications de confort sont développées principalement pour divertir les passagers, elles permettent un accès à Internet, le partage et la messagerie inter-véhicule ou encore les jeux en réseau ou hors ligne. Mais aussi à des fins publicitaires destinées aux commerciaux

qui pourraient lorsque nécessaires, consulter le type de commerce présent autour d'eux et sélectionner celui correspondant à leurs besoins. Et sont développées à des fins informatives comme des applications de gestion du stationnement, afin d'informer le conducteur des places disponibles prêt de sa position via les données fournies par un RSU.

2.5 Les défis des réseaux véhiculaires

Dans VANET, il y a beaucoup de défis en termes de conception de protocoles pour échanger des informations entre les véhicules, la confidentialité des données et la plateforme de communication. Peu de défis sont classés ci-dessous qui impose de déployer VANET :

2.5.1 Défis de sécurité

Les problèmes de sécurité sont une préoccupation majeure dans l'environnement VANET lorsque les nœuds communiquent entre eux par la communication sans fil. Toute modification des informations réseau par un nœud de fraude peut causer un préjudice important aux conducteurs et passagers du véhicule, ainsi que la partition du réseau et la diminution des performances de l'ensemble du réseau. Ces réseaux font face à de nombreuses attaques de sécurité qui peuvent être classées sous ces types :

- Attaques du réseau : l'attaquant peut causer un danger pour le conducteur ou/et les passagers s'il dépend des informations de l'application, en les redirigeant vers d'autres routes ou les bloquant dans la circulation ;
- attaques d'application : l'attaquant peut modifier, supprimer ou fabriquer un message vers les véhicules, pour servir ses intérêts ou bien pour l'amusement, ce qui peut être un danger pour les autres ;
- attaques sociales : l'objectif de ces attaques est de créer un problème du réseau en influant négativement sur les conducteurs en envoyant des messages émotionnels.

2.5.2 Défis du réseau

Aucun coordinateur de communication ne peut être assumé, bien que certaines applications impliquent vraisemblablement une infrastructure (par exemple, avertissement de violation de signal de trafic, collecte de péage, etc.), plusieurs applications seront censées fonctionner de manière fiable en utilisant des communications décentralisées. Plusieurs défis sont au courant :

- La mobilité : chaque nœud du réseau est mobile, et peut se déplacer d'un endroit à un autre, mais la mobilité est limitée, les véhicules font connexion avec un autre véhicule que peut être jamais rencontré auparavant, et cette connexion ne dure que quelques

secondes que chaque véhicule va dans sa direction, et ces deux véhicules ne peuvent jamais se réunir à nouveau. En outre, puisque deux véhicules restent dans leur portée de communication pour une question de secondes, il est un problème de recherche ouvert pour développer le modèle topologique riche pour VANET qui différera des topologies de réseau traditionnelles qui exigent une interaction significative entre l'expéditeur et le récepteur.

- la connectivité : la connectivité entre les nœuds peut être très éphémère, et peut être ne se reproduira pas, les véhicules déplacent et font des connexions avec d'autres véhicules, ces connexions seront perdues car chaque véhicule a une mobilité élevée, et peut être déplacé dans la direction opposée ;
- scalabilité du réseau : l'échelle de ce réseau dans le monde dépasse approximativement les 750 millions de nœuds[4], et ce nombre est en croissance, VANET travaille dans des zones de très faible densité telles que les autoroutes, les zones à forte densité de trafic, comme les villes, les zones urbaines où les encombrements sont élevés et les intersections importantes sur la route. Le nombre de véhicules et la conception de protocoles évolutifs peuvent constituer un gros problème pour les chercheurs et les développeurs.

2.5.3 Défis techniques

Les défis techniques portent sur les obstacles techniques à résoudre avant le déploiement de VANET. Quelques défis sont donnés ci-dessous :

- Gestion du réseau : en raison de la mobilité élevée, la topologie du réseau et l'état du canal changent rapidement. Donc, nous ne pouvons pas utiliser des structures comme l'arbre parce que ces structures ne peuvent pas être mises en place et maintenues aussi rapidement que la topologie a changé[4] ;
- contrôle de congestion et de collision : la taille du réseau illimitée crée également un défi. Les partitions réseau se produisent fréquemment pendant les heures de pointe, la charge de trafic est très élevée et donc le réseau est congestionné et la collision se produit dans le réseau ;
- impact environnemental : les VANETs utilisent les ondes électromagnétiques pour la communication. Ces vagues sont affectées par l'environnement qui peut présenter un obstacle pour ces ondes, tels que les bâtiments, les immeubles ou même les montagnes.

2.5.4 Défi commun : livraison de données

La livraison de données ou le routage joue un rôle important dans les applications de VANET, mais les protocoles de routage des réseaux traditionnels sont inappropriés avec

cet environnement de véhicule car les véhicules se déplacent en vitesse rapide résultant des changements rapides dans la topologie du réseau et ne pas établir la connectivité de bout en bout entre les nœuds de source et de destination[5]. La livraison de données dans un réseau ad-hoc s'appuie fortement sur le protocole de routage, qui a fait l'objet d'études approfondies depuis de nombreuses années. Cependant, la plupart des protocoles supposent que des nœuds intermédiaires peuvent être trouvés pour configurer une connexion de bout en bout, sinon, le paquet sera abandonné. Étant donné que le diamètre du réseau dans les VANETs est relativement faible, il devrait y avoir d'autres stratégies pour la transmission des données dans les réseaux de véhicules et les algorithmes traditionnels ne sont pas applicables. Ainsi, il peut être difficile d'utiliser la bande passante disponible allouée à VANET pour améliorer la livraison des messages ainsi que pour développer des protocoles adaptatifs de routage QoS qui établiront de nouvelles routes rapides et efficaces. Cela a incité les chercheurs et les développeurs à établir des algorithmes de routage riches et robustes pour faire face à l'environnement VANET qui sera en mesure de fournir un débit élevé et un meilleur ratio de livraison de paquets.

2.6 Les protocoles de routage dans les réseaux VANETs

Le routage est le processus de transmission des données entre deux nœuds considérés comme source et destination d'information. Une communication multi-sauts est 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.

Les chemins de routage multi-sauts peuvent être construits par des protocoles de routage de types variés, qui permettent d'assurer la transmission des paquets vers un (ou des) nœud(s) destinataire(s). En d'autres termes, les protocoles de routage sont chargés de déterminer la façon de relayer le paquet à sa destination, la façon de calculer la route et aussi d'ajuster la trajectoire en cas d'échec et comment améliorer la connectivité de données.

Les protocoles de routage dans les réseaux VANETs sont classés en trois (3) types : routage unicast, multicast/geocast et broadcast. Dans les applications de transfert de données volumineuses (par exp : applications multimédias), le routage unicast est utilisé pour le transfert des données. L'objectif des protocoles unicast est de transmettre des paquets depuis un nœud source vers un seul nœud destinataire (donc à travers un seul chemin multi-sauts). En outre, le multicast est essentiel pour la communication de plusieurs véhicules dans plusieurs situations, comme pour les intersections immédiates, l'incidence de la circulation, les incidents et les conditions dangereuses de la route. Le

routage geocast est une technique d'acheminement de paquets, incluse dans le multicast, pour transmettre des données à l'ensemble des nœuds situés dans une certaine zone géographique. Par ailleurs, le broadcast est nécessaire pour faire circuler des données vers des emplacements inconnus/non spécifiés. Dans ce mémoire on s'intéresse par le routage unicast.

La plupart des protocoles de routage de réseaux VANETs utilisent l'information de position afin d'améliorer leurs performances d'une façon compatible avec leurs besoins, tels que le changement de topologie dynamique et la fragmentation du réseau fréquente. Les techniques de routage sont utilisées depuis longtemps dans les réseaux MANETs et la plupart de celles-ci peuvent également être appliqué dans les VANETs. La figure 2.2 montre la classification des protocoles de routage unicast dans les réseaux VANETs :

Les protocoles de routage proposés pour les communications point à point dans les VANETs peuvent être classés en protocoles basés sur la topologie et protocoles basés sur la position (géographiques). Les protocoles basés sur la topologie prennent les décisions de routage en s'appuyant principalement sur des informations sur la topologie du réseau. C'est-à-dire qu'ils utilisent les informations sur l'existence de liaisons de communication entre les différents nœuds pour déterminer à quel chemin un paquet doit être acheminé. Cette classe peut être divisée en protocoles proactifs et réactifs. Dans les protocoles proactifs, le saut suivant vers une certaine destination peut être directement trouvé à partir d'une table de routage stockée à chaque nœud, tandis que dans les protocoles réactifs, une procédure de découverte d'itinéraire est initiée avant le début de transmission de paquets de données entre les nœuds source et destination.

La deuxième catégorie des protocoles de routage, connue sous le nom de protocoles géographiques (basés sur la position), prend des décisions de routage basées uniquement sur des informations de position et sans dépendre d'aucune information de topologie. Chaque nœud de relais connaît les positions de ses voisins à un seul saut ainsi que la position de la destination. La position de la destination est déterminée par un service de localisation (tel que GLS "Grid Location Service"), tandis que les positions des voisins à un seul saut sont déterminées en utilisant un algorithme de découverte de voisin, tel que l'échange de messages Hello. Sur la base de cette information de position, chaque nœud de relais transmet un paquet à un candidat parmi ses voisins à un saut, qui est prévu pour délivrer le paquet plus près ou plus rapidement à la destination finale.

Ces protocoles sont également classés en réseau tolérant au délai (DTN "Delay Tolerant Network"), non-DTN et hybride. le DTN utilise la stratégie carry & forward pour surmonter la déconnexion fréquente des nœuds dans le réseau. Dans la stratégie carry & forward lorsqu'un nœud ne peut pas contacter d'autres nœuds, il stocke le paquet et il l'envoie au prochain nœud rencontré (le plus près à la destination ou allant dans sa direction, selon le protocole de routage). Par ailleurs, les protocoles non-DTN sont classés en

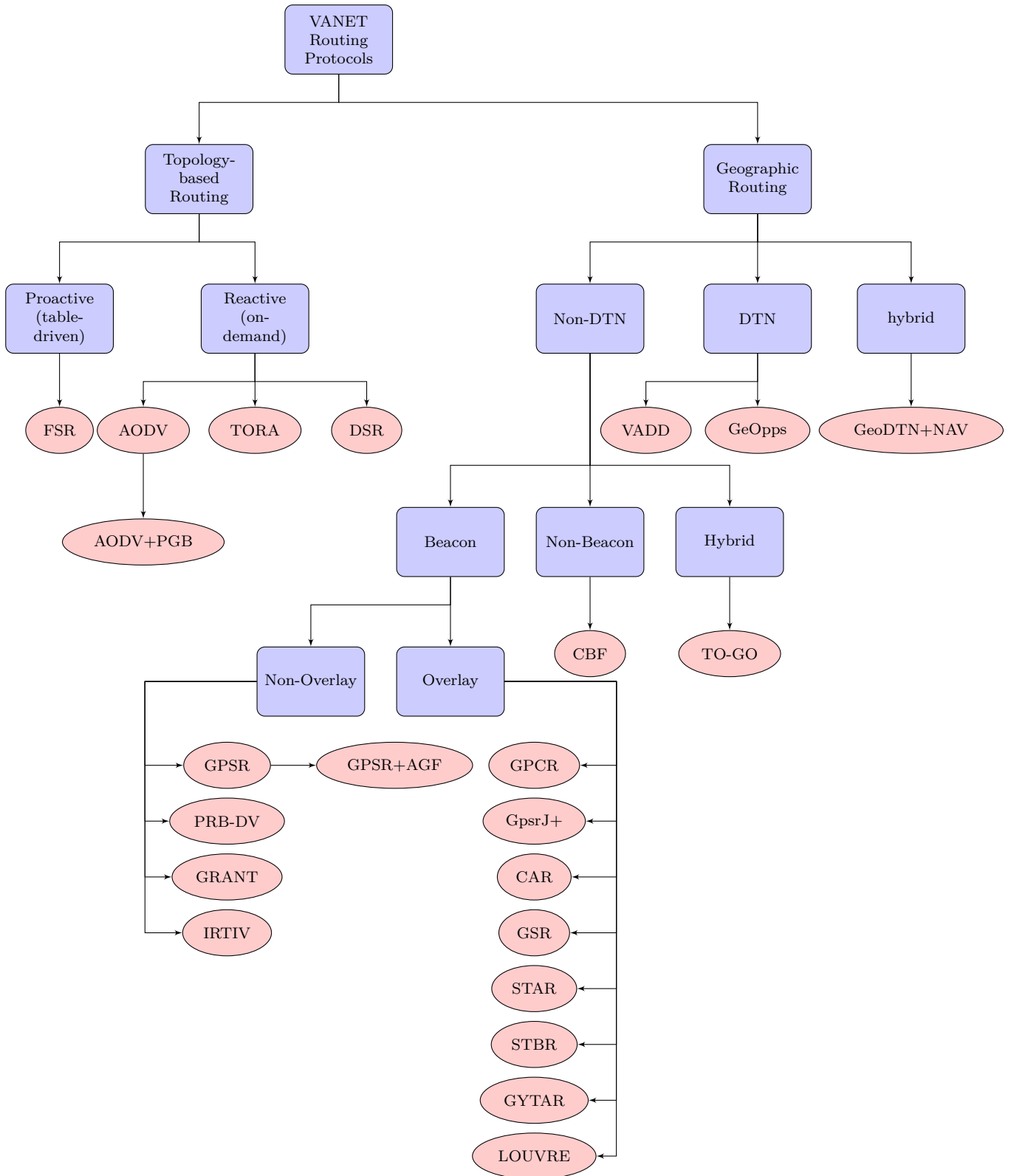


FIGURE 2.2 – Les protocoles de routage unicast dans les réseaux VANETs

trois (3) types : Beacon, non-Beacon et hybride. Beacon ou une balise sert à transmettre périodiquement un message Hello. Il expose la présence et la position d'un nœud. Une

entrée sera supprimée de la table de voisins d'un nœud récepteur s'il échoue à recevoir un beacon après une certaine période de temps à partir du nœud correspondant. Et enfin, on peut classer les protocoles de routage qui utilisent les beacons en non-Overlay et Overlay qui représente un réseau où chaque nœud est connecté par des liens virtuels ou logiques qui sont construits sur un réseau existant, c'est une couverture virtuelle.

La sélection des nœuds de relais dans le routage est gouvernée par un ensemble de facteurs défini dans la littérature des réseaux véhiculaires, ceux-ci influent sur la performance du système, et qui peuvent être présentés dans : le nombre de véhicules, vitesse de véhicules, largeur du canal (bande passante), schéma de diffusion (protocole de routage), la priorité des messages, la périodicité des messages, modèle de mobilité, modèle de propagation et le roadmap simulé (cf. chapitre 4 sous-section 5.2).

2.7 Conclusion

Dans ce chapitre, nous avons présenté la motivation des réseaux VANETs ainsi ses architectures, les applications et les caractéristiques de ce type de réseaux, ainsi les défis qui rendent le développement des protocoles plus difficile, et enfin on a présenté les protocoles de routage proposés dans VANET. Dans le chapitre suivant, on va présenter à quoi sert l'analyse factorielle, sa classification et ses méthodes.

Chapitre 3

L'analyse factorielle

3.1 Introduction

L'analyse factorielle est une méthode scientifique générale pour l'analyse de données, elle a un certain nombre d'objectifs de recherche. Les interdépendances entre les variables peuvent être délimitées, les masses de données peuvent être réduites à un sous-ensemble parcimonieux, les données peuvent être mises à l'échelle ou transformées, des hypothèses et théories peuvent être testées, etc.

Dans ce chapitre, nous présentons l'analyse factorielle, ces types, et ses différentes méthodes qui servent à donner les liens pouvant exister entre les différentes données, ainsi qu'à en tirer une information statistique qui sert à décrire de façon plus succincte les principales informations contenues dans ces données.

3.2 Définition

L'analyse factorielle est une méthode de la famille de la statistique multivariée, créée par **Charles Spearman**, utilisée pour décrire la variabilité entre des nuages observés, au moyen de nuages latents (non observés) dits facteurs. Pour diminuer le nombre de variables, la méthode calcule ces variables latentes comme combinaisons linéaires des variables observées[6]. Cette méthode est utilisée en informatique dans l'imagerie médicale, satellitaire, compression irréversible, dans le data mining pour l'extraction de connaissances, utilisée aussi dans le web sémantique (ontologies), et même dans l'analyse statistique (approximation / régression), etc. Au delà de l'informatique, elle peut être utilisée en psychologie, en sciences humaines et sociales, dans les tests théoriques, et d'une façon plus générale dans toute discipline faisant face à de grandes quantités de données.

L'analyse factorielle désigne un outil de réduction de données, elle permet d'éliminer la redondance dans l'ensemble de variables corrélées, et de les représenter dans un ensemble plus petit de variables dérivées. Dans le but de simplifier les données, vérifier la construction de l'échelle, construire des indices, etc.

Les facteurs (les variables non observées) découverts par une analyse factorielle peuvent être traités comme des concepts formels exprimant des relations mathématiques, comme des concepts théoriques ou comme des concepts empiriques. Ces facteurs peuvent être définis par une application inductive de l'analyse ou par déduction d'une théorie formelle. Ils peuvent être utilisés pour décrire des régularités de données réelles ou pour estimer des modèles universels à partir d'un échantillon. Les facteurs peuvent également être utilisés pour découvrir l'ordre causal, expliquer les uniformités ou classer les corrélations.

3.3 Caractéristiques de l'analyse factorielle

Le rôle de l'analyse factorielle devient important dans les recherches, en raison de ces caractéristiques :

- L'analyse factorielle peut analyser un grand nombre de phénomènes avec l'aide d'un ordinateur puissant que l'analyse de 100 variables devient de routine ;
- elle démêle les interrelations entre les phénomènes en unités fonctionnelles ou des modèles distincts ou indépendants de comportement et identifie les influences indépendantes ;
- elle gère les phénomènes sociaux dans la situation. i.e. Les interrelations entre comportement et environnement peuvent être analysées telles qu'elles existent dans la vie réelle ;
- l'analyse factorielle est un instrument flexible applicable à un large éventail de modèles de recherche et à une variété de données ;
- elle fournit un ensemble d'équations qui peuvent être utilisées pour décrire et prédire le comportement. Le modèle d'analyse factorielle peut ainsi être utilisé comme théorie mathématique du comportement, en s'appuyant sur un champ familier de mathématiques pour faire des déductions et des prédictions comportementales ;
- l'analyse factorielle a une représentation géométrique qui permet la représentation visuelle des relations comportementales.

3.4 Les types de l'analyse factorielle

On peut distinguer deux (2) types de l'analyse factorielle :

3.4.1 L'analyse factorielle exploratoire (AFE)

L'analyse factorielle exploratoire est utilisée pour identifier les interrelations complexes entre les éléments et les éléments du groupe qui font partie des concepts unifiés. En d'autres termes, utilise les données pour déterminer combien de facteurs il devrait y avoir

et quels éléments à associer à ces facteurs. Peut être réalisée en utilisant la méthode ACP et une variété d'autres méthodes.

3.4.2 L'analyse factorielle confirmatoire (AFC)

L'analyse factorielle confirmatoire est une approche plus complexe qui teste l'hypothèse selon laquelle les éléments sont associés à des facteurs spécifiques. Elle utilise la modélisation par équation structurelle pour tester un modèle de mesure selon laquelle la charge sur les facteurs permet une évaluation des relations entre les variables observées et les variables non-observées. On dit qu'elle utilise la théorie pour préciser une allocation de facteur d'élément et d'évaluer s'il s'agit d'un choix raisonnable.

3.5 Les méthodes de l'analyse factorielle

Plusieurs méthodes de l'analyse factorielle ont été conçues selon le besoin et l'objectif. Dans ce qui suit on présente les principales méthodes de l'analyse factorielle :

3.5.1 Analyse en composantes principales (ACP)

L'analyse en composantes principales est une méthode qui consiste à transformer des variables quantitatives corrélées en nouvelles variables non-corrélées les unes des autres. Ces nouvelles variables sont nommées « composantes principales » ou « axes principaux ». Elle permet d'obtenir des facteurs non-corrélés qui sont des combinaisons linéaires des variables de départ, afin d'utiliser ces facteurs dans des méthodes de modélisation telles que la régression linéaire, la régression logistique ou l'analyse discriminante.

On applique habituellement une ACP sur un ensemble de N variables aléatoires X_1, \dots, X_N connues à partir d'un échantillon de K réalisations conjointes de ces variables. Cet échantillon de ces N variables aléatoires peut être structuré dans une matrice M à K lignes et N colonnes. Chaque variable aléatoire $X_n = (X_{1,n}, \dots, X_{K,n})$ a une moyenne \bar{X}_n et un écart type σ_{X_n} [7]. On cherche une nouvelle matrice $\Pi(M)$ avec N nouvelles colonnes, chaque colonne représente une variable synthétique. Ce processus assure la minimisation de la corrélation entre les données.

3.5.2 Analyse factorielle des correspondances (AFC)

L'analyse factorielle des correspondances est une méthode qui permet d'étudier l'association entre deux variables qualitatives. Et est principalement utilisée pour de grands tableaux de données toutes identiques entre elles, et surtout permet d'étudier des tableaux de contingence (ou tableau croisé de co-occurrence). Cette méthode est basée sur l'inertie.

Elle vise à rassembler en un nombre réduit de dimensions la plus grande partie de l'information initiale en s'attachant non pas aux valeurs absolues mais aux correspondances entre les variables. L'AFC permet de déterminer ainsi qu'à hiérarchiser l'ensemble des dépendances entre les lignes et les colonnes du tableau.

La technique de l'AFC est parfois utilisée pour des tableaux d'un autre type que tableau de contingence. Par contre, son utilisation pour ces tableaux oblige toujours à des adaptations spécifiques au type de tableau (logiques, de notes d'intensité, de rang, de mesure, qualitatifs)[8].

Pour croiser plusieurs variables, on utilise l'analyse factorielle des correspondances multiples (AFCM).

3.5.3 Analyse factorielle multiple (AFM)

L'analyse factorielle multiple permet d'analyser simultanément plusieurs tableaux de variables, et d'obtenir des résultats, notamment des représentations graphiques, qui permettent d'étudier la relation entre les observations, les variables et les tableaux. À l'intérieur d'un tableau, les variables doivent être de même nature (quantitative ou qualitative), mais les tableaux peuvent être de différents types.

L'analyse factorielle multiple est une synthèse de l'ACP et l'AFCM qu'elle permet de généraliser à des variables hétérogènes. La méthodologie de l'AFM se décompose en deux étapes :

1. On réalise successivement pour chacun des tableaux une ACP ou une AFCM en fonction de la nature des variables. On conserve la valeur de la première valeur propre de chacune des analyses pour pondérer ensuite les différents tableaux dans la seconde partie de l'analyse ;
2. on réalise ensuite une ACP pondérée sur les colonnes de l'ensemble des tableaux, sachant que les tableaux de variables qualitatives sont transformés en tableaux disjonctifs complets, chacune des indicatrices des tableaux disjonctifs ayant un poids en fonction de la fréquence de la modalité concernée. La pondération des tableaux permet d'éviter que les tableaux comprenant plus de variables ne pèsent trop dans l'analyse.

L'analyse factorielle multiple s'avère très utile pour analyser des enquêtes lorsque les questions peuvent être regroupées par thèmes, ou lorsque les mêmes questions sont posées à plusieurs intervalles de temps[9].

3.5.4 Analyse factorielle de données mixtes (AFDM)

L'analyse factorielle des données mixtes consiste à remplacer les variables qualitatives/quantitatives par une série d'indicatrices via un codage disjonctif complet. L'information n'est pas dénaturée car il y a une bijection exacte entre les variables initiales et les variables recodées. Mais il n'est pas très judicieux de traiter directement avec une ACP des données mélangeant des colonnes de valeurs numériques (continues) et des indicatrices[10].

L'AFDM est ensuite comparée à la pratique empirique qui consiste à introduire directement des indicatrices dans une ACP. On montre enfin son équivalence avec une AFM dans laquelle chaque groupe est réduit à une variable, quantitative ou qualitative.

3.5.5 Analyse factorielle multiple hiérarchique (AFMH)

Comme en AFM, les variables dans l'analyse factorielle multiple hiérarchique sont structurées en groupes, mais selon une hiérarchie.

On réalise une analyse factorielle (ACP par exp.) de chacun des groupes de variables associés au premier niveau de partition ; on stocke les facteurs obtenus. À chaque nœud de l'indice de niveau 2, on réalise une AFM dans laquelle les groupes de variables sont ceux qui constituent le nœud. (les variables d'un groupe sont remplacées par les facteurs issus de son ACP). À leur tour, les facteurs de ces AFM sont stockés : ce sont eux qui représenteront les groupes de variables associés aux nœuds dans les AFM de l'étape suivante[11].

Lors de la dernière étape (i. e. l'étape P), les principaux facteurs de variabilité de l'AFMH sont obtenus à l'issue de l'AFM des groupes de facteurs principaux associés aux nœuds de l'indice de niveau P.

3.6 L'analyse discriminante

L'analyse factorielle discriminante est une technique statistique qui vise à décrire, expliquer et prédire l'appartenance à des groupes prédéfinis (classes, modalités de la variable à prédire, etc.) d'un ensemble d'observations (individus, exemples, etc.) à partir d'une série de variables prédictives (descripteurs, variables exogènes, etc.)[12]. Elle est utilisée dans nombreux domaines ; en médecine (par exp. pour détecter les groupes à hauts risques cardiaques), en biologie (quand on veut affecter un objet à sa famille d'appartenance à partir de ses caractéristiques physiques), dans le domaine financière, et même pour la prévision météorologique.

On peut distinguer deux grandes approches selon l'objectif :

3.6.1 Approche descriptive

L'approche descriptive est une technique de statistique exploratoire qui travaille sur un ensemble de n observations décrites par J variables, réparties en K groupes. Elle vise à produire un nouveau dispositif de représentation, constitué de combinaisons linéaires des variables initiales, qui sert à séparer au mieux les K catégories. Elle propose une représentation graphique qui sert à visualiser les proximités entre les observations, appartenant au même groupe ou non. Il s'agit principalement d'une méthode géométrique.

3.6.2 Approche prédictive

Il s'agit de construire une fonction de classement (règle d'affectation, etc.) qui sert à prédire le groupe d'appartenance d'un individu à partir des valeurs prises par les variables prédictives. En ce sens, cette technique se rapproche des techniques supervisées en apprentissage automatique telles que les arbres de décision, les réseaux de neurones, etc. Elle repose sur un cadre probabiliste[12].

3.6.2.1 Analyse discriminante linéaire

L'analyse discriminante linéaire fait partie des techniques d'analyse discriminante prédictive. Il s'agit d'expliquer et de prédire l'appartenance d'un individu à une classe prédéfinie à partir de ses caractéristiques mesurées avec variables prédictives. Elle peut être comparée aux méthodes supervisées développées en apprentissage automatique ainsi qu'à la régression logistique développée en statistique.

3.7 La régression

La régression est une méthode statistique particulièrement utilisée pour analyser la relation d'une variable comparée à une ou plusieurs autres. En d'autres termes, elle est utilisée pour estimer une fonction f décrivant une relation entre une variable expliquée (dépendante) continue Y et une ou plusieurs variables explicatives (indépendantes) X_i . $Y = f(X_1, X_2, \dots, X_n) + \epsilon$ Dont f décrit la variation systématique de la relation, et ϵ représente la variation non systématique (aléatoire) de la relation. On parle de régression bivariée lorsqu'il y a une seule variable indépendante.[13]

C'est une technique de prévision qui permet de trouver une corrélation entre deux séries de données. La pente de la droite qui en résulte est obtenue lorsque la somme des carrés des distances à cette droite passe par un minimum (méthode des moindres carrés) (cf. le graphe 3.1).

Plusieurs modèles ont été déployés dans la régression :

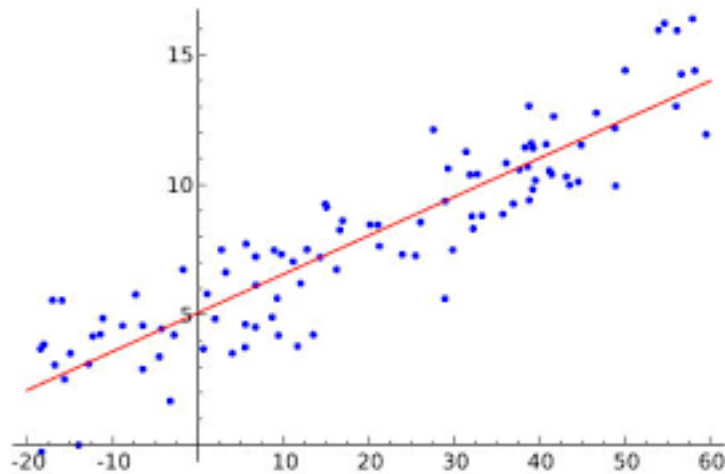


FIGURE 3.1 – Droite de régression

3.7.1 La régression linéaire

Un modèle de régression linéaire décrit la relation entre la variable y et la ou les variables indépendante(s) x comme ayant la forme d'une équation linéaire.

Étant donné un échantillon aléatoire (Y_i, X_i) , $i = 1, \dots, n$ un modèle de régression simple suppose la relation affine suivante entre Y_i et X_i : $Y_i = aX_i + b$, $i = 1, \dots, n$

Empiriquement, à partir d'observations (y_i, x_i) , $i = 1, \dots, n$, on représente dans un graphe la totalité de ces points représentant des mesures d'une grandeur y_i en fonction d'une autre x_i . Les points paraissent alignés. On peut alors proposer un modèle linéaire, c'est-à-dire chercher la droite dont l'équation est $Y_i = aX_i + b$ et qui passe au plus près des points du graphe, selon la méthode des moindres carrés, en rendant minimale la somme des carrés des écarts des points à la droite $\sum_{i=1}^n (y_i - ax_i - b)^2$ où $(y_i - ax_i - b)^2$ représente le carré de la distance verticale du point expérimental (y_i, x_i) à la droite reconnue comme la meilleure.

3.8 La méthode χ^2

Dans la théorie des probabilités et la statistique, la distribution du χ^2 avec k degrés de liberté est la distribution d'une somme des carrés de k variables aléatoires normales standard indépendantes.

Le nombre de degrés de liberté est le nombre de valeurs dans le calcul final d'une statistique qui sont libres de varier. En d'autres termes, le nombre de degrés de liberté peut être défini comme le nombre minimum de coordonnées indépendantes qui peuvent spécifier la position du système complètement.

La méthode χ^2 s'agit d'une des distributions de probabilité les plus largement utilisées dans la statistique inférentielle, dans l'essai d'hypothèse ou dans la construction d'intervalles de confiance. La distribution du χ^2 est utilisée dans les tests χ^2 communs pour la qualité d'ajustement d'une distribution observée à une autre théorique, l'indépendance de deux critères de classification des données qualitatives et dans l'estimation de l'intervalle de confiance pour un écart type de population d'une distribution normale à partir d'un échantillon d'écart type.

La loi χ^2 est une loi à densité de probabilité. Cette loi est caractérisée par un paramètre dit degrés de liberté à valeur dans la totalité des entiers naturels (non nuls).

Soit X_1, \dots, X_k k variables aléatoires indépendantes de même loi normale centrée et réduite, alors par définition la variable X , telle que $X = \sum_{i=1}^k X_i^2$ suit une loi du χ^2 à k degrés de liberté.

Soit K une variable aléatoire suivant une loi du χ^2 à K degrés de liberté, on notera $\chi^2(k)$ la loi de X .

Quand k est grand ($k > 100$), la loi du χ^2 peut s'approximer par une loi normale d'espérance k et de variance $2k$.

La principale utilisation de cette loi consiste à apprécier la correction d'une loi de probabilité à une distribution empirique en utilisant le test du χ^2 basé sur la loi multinomiale. D'une façon plus générale elle s'applique dans le test d'hypothèses à certains seuils (indépendance surtout).

Un test χ^2 est un test d'hypothèse statistique dans lequel la distribution d'échantillonnage de la statistique de test est une distribution χ^2 lorsque l'hypothèse nulle est vraie (le terme « hypothèse nulle » est un énoncé général ou une position par défaut qu'il n'existe aucune relation entre deux phénomènes mesurés, ou aucune association entre les groupes. Si la différence entre les deux distributions est réduite, l'hypothèse nulle sera acceptée. Si la différence est grande, l'hypothèse nulle sera rejetée.).

Les tests χ^2 sont souvent construits à partir d'une somme d'erreurs carrées ou de la variance de l'échantillon. Les statistiques d'essai qui suivent une distribution de χ^2 découlent d'une hypothèse de données indépendantes distribuées normalement, ce qui est valable dans de nombreux cas en raison du théorème de limite centrale. Un test χ^2 peut être utilisé pour tenter de rejeter l'hypothèse nulle que les données sont indépendantes.

On considère également ce test, comme un test dans lequel il est asymptotiquement vrai, ce qui signifie que la distribution d'échantillonnage (si l'hypothèse nulle est vraie) peut être faite pour approcher une distribution χ^2 aussi près que souhaité en rendant la taille de l'échantillon assez grande. Le test du χ^2 est utilisé pour déterminer s'il existe une différence significative entre les fréquences attendues et les fréquences observées dans une ou plusieurs catégories.

3.9 Conclusion

L'analyse factorielle permet aux chercheurs d'étudier des phénomènes comportementaux de grande complexité et de diversité et de modeler leurs découvertes dans des théories scientifiques. Elle propose à la fois une analyse technique et une structure théorique. Elle permet la manipulation inductive et déductive de données qualitatives et quantitatives. Dans le chapitre suivant nous allons présenter les protocoles de routage choisis pour étudier leur performance en l'influence de certains facteurs, et l'outil qui permet de déterminer parmi ces derniers le plus pertinents.

Chapitre 4

Protocoles de routage et outil d'identification des paramètres les plus pertinents (basé sur 2^k)

4.1 Introduction

La mobilité des nœuds dans les réseaux ad-hoc fortement dynamiques empêche les protocoles unicast de fonctionner proprement. En effet, la forte mobilité des véhicules provoque un changement rapide de la topologie du réseau et engendre une rupture du chemin de routage. Pour remédier à ce problème, des protocoles de routage unicast adaptés aux VANETs ont été proposés.

Comme nous avons cité dans la section 2.6 du chapitre 2, le routage unicast est classé en topologique et géographique. Le routage unicast dans les réseaux VANETs se base principalement sur l'information de la position des véhicules et non pas sur la topologie du réseau en raison de la mobilité rapide et le changement fréquent de la topologie qui rend l'acheminement des informations plus difficile. Par ailleurs, il existe des paramètres, tels que la densité, la bande passante, la périodicité des messages, etc, qui influent sur la performance des protocoles de routage.

Dans ce chapitre, nous allons présenter les protocoles de routage choisis (cf.4.1) pour étudier leur fiabilité dans l'influence de certains facteurs. Ainsi, l'outil qui nous permet d'identifier les paramètres les plus influents.

4.2 Ad hoc On-demand Distance Vector (AODV)

Ad hoc On-demand Distance Vector est un protocole de routage réactif; ainsi, les chemins sont découverts et maintenus à la demande. Ce protocole a été développé principalement pour les MANETs, mais il peut être utilisé dans les VANETs. Lorsqu'un nœud

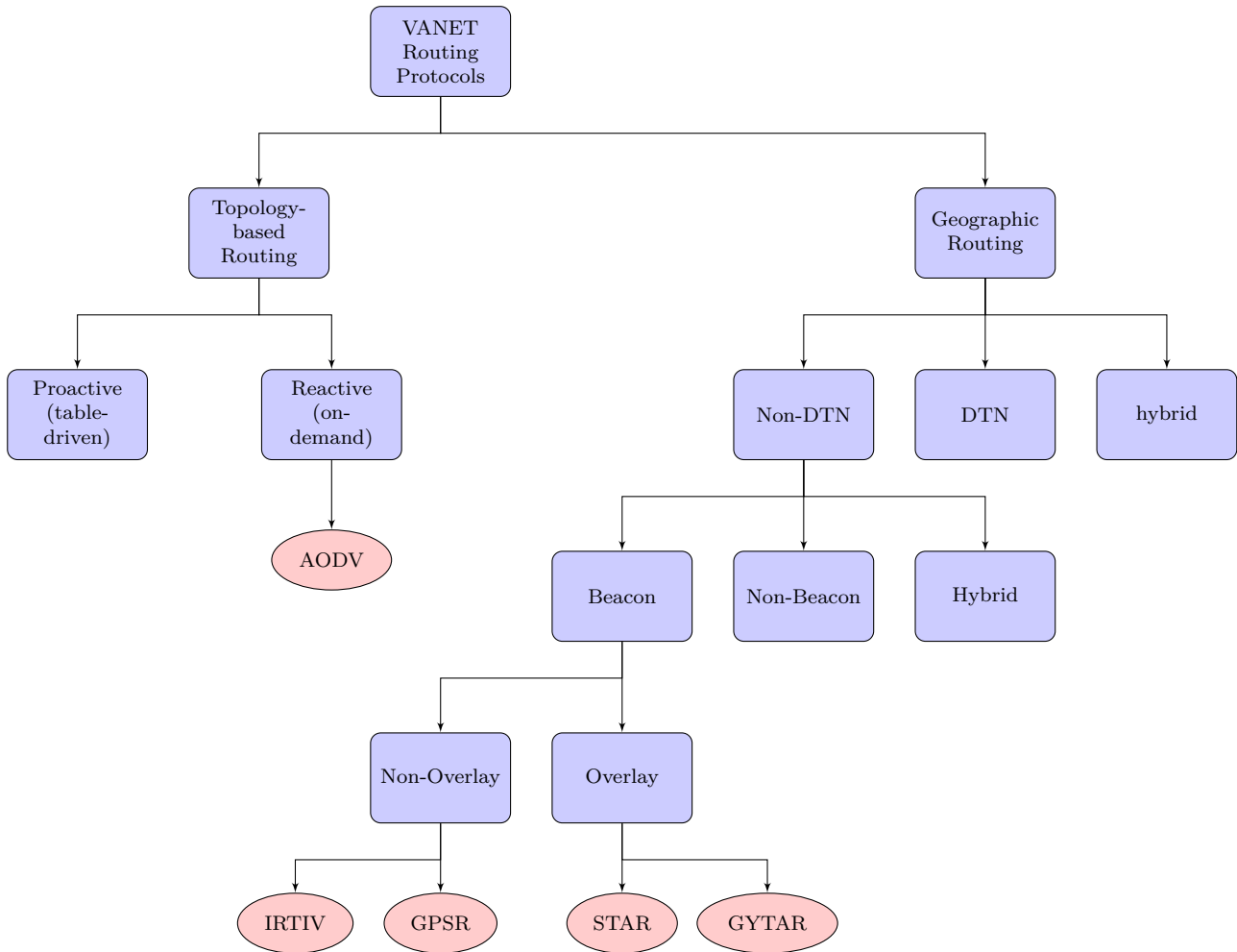


FIGURE 4.1 – Les protocoles de routage unicast choisis pour l'étude

doit commencer une transmission, AODV diffuse un paquet de découverte de façon broadcast. La destination va utiliser le chemin emprunté par le premier paquet de découverte qui l'a atteint. Il envoie alors un paquet réponse afin d'annoncer ce chemin à la source. L'envoi de données peut alors commencer. En cas de disparition d'un lien sur le chemin choisi, un paquet d'erreur est généré. On peut détailler le fonctionnement dans ce qui suit :

Le protocole AODV[14], utilise quatre (4) types de messages afin de construire et contrôler l'état des liens lors d'une communication : demande d'itinéraire RREQ (Route REQuest), réponse d'itinéraire RREP (Route REPLY), Hello et erreur de route RERR (Route ERRor). Le message de requête (RREQ) est diffusé par le nœud source afin de découvrir le réseau. Le message de réponse (RREP) est initié par le nœud destination ou par un nœud qui connaît le chemin jusqu'à la destination afin de rapporter à la source les informations à propos du chemin choisi. Ensuite, les messages Hello permettent de contrôler l'état de chaque lien de la route établie. Les messages Hello sont générés seulement par

les nœuds qui font déjà partie de cette route et ont une durée de vie d'un seul saut. Trois (3) messages Hello sont envoyés simultanément, de manière périodique entre chaque pair de nœuds de la route choisie. En cas de non-réception d'un de ces trois messages, le lien est considéré comme non-utilisable et un message d'erreur (RERR) est initié par le nœud qui l'a découvert.[1]

Un paquet, qu'il soit de données ou de contrôle (RREP / RREQ), possède un numéro d'identification unique. Ce numéro est dépendant du temps : le paquet avec le numéro le plus élevé est le paquet le plus récent. On appelle cela le numéro de séquence. Ce procédé est très utile pour le routage de l'information. Par exemple, si le nœud source reçoit plusieurs réponses, il choisira d'écouter celle qui possède la route la plus à jour (c'est-à-dire celle qui a le numéro de séquence le plus élevé)[1]. Le protocole AODV utilise le principe des numéros de séquences afin de contrôler si les liens utilisés pour atteindre la destination sont suffisamment à jour. Chaque nœud possède un numéro de séquence qu'il incrémente à chaque fois qu'il reçoit des informations utiles de la part des messages RREQ, RREP ou RERR. Cela permet à chaque nœud du réseau de valider si la route qu'il connaît vers la destination est bien la bonne. Lors de la réception d'un message RREQ ou RREP, les nœuds exécutent une série de vérification afin de contrôler que la table de routage qu'il possède est la meilleure. Ces décisions sont prises en fonction du numéro de séquence ainsi que du nombre de sauts entre le nœud et la destination. L'algorithme 1 [15] résume le fonctionnement du protocole AODV.

Algorithme 1 : AODV pseudo code

```

Procedure broadcast RREQ
  Send RREQ (nodeX);
  SET  $seq - nbr - rq \leftarrow 1, Hop - count - rq = 0$ ;
  Broadcast RREQ to neighbors;

Procedure Handling RREQ
  Receive RREQ (RREQ, nodeX);
  if ( $nodeX = destination$ ) then
    Update Route, Send RREP (nodeX, RREQ);
  if ( $nodeX \neq destination$ ) then
    if ( $seq - nbr - rq > seq - nbr - tb$ ) OR ( $(seq - nbr - rq = seq - nbr - tb)$  AND ( $Hop - count - rq < Hop - count - tb$ )) then
      Update, Forward RREQ;
    Forward RREQ, Update RREQ :  $seq - nbr - tb = seq - nbr - rq$ ;
     $Hop - count - tb = Hop - count - rq + 1$ ;
  Update Route :  $seq - nbr - tb = seq - nbr - rq$ ;
   $Hop - count - tb = Hop - count - rq$ ;

Procedure broadcast RREP
  Send RREP (nodeX, RREP);
  SET  $seq - nbr - rp \leftarrow seq - nbr - rq, Hop - count - rp \leftarrow 0$ ;
  Broadcast RREP to neighbors;

Procedure Handling RREP
  Receive RREP (RREP, nodeX);
  if ( $nodeX = source$ ) then
    Update Route, data;
  if ( $nodeX \neq destination$ ) then
    if ( $seq - nbr - rp > seq - nbr - tb$ ) OR ( $(seq - nbr - rp = seq - nbr - tb)$  AND ( $Hop - count - rp < Hop - count - tb$ )) then
      Update, Forward RREP;
    Forward RREP, Update RREP :  $seq - nbr - tb \leftarrow seq - nbr - rp$ ;
     $Hop - count - tb \leftarrow Hop - count - rp + 1$ ;
  Update Route :  $seq - nbr - tb \leftarrow seq - nbr - rp$ ;
   $Hop - count - tb \leftarrow Hop - count - rp$ ;

```

4.3 Greedy Perimeter Stateless Routing (GPSR)

Le GPSR est un protocole de routage basé sur la position visant à gérer les environnements mobiles de véhicules. Il est le plus approprié pour les autoroutes, où les nœuds sont uniformément répartis. GPSR assume que chaque nœud connaît sa position géographique (par GPS), la position de ses voisins à un saut (par l'envoi périodique du beacon), et celle de la destination (en utilisant un service de localisation (GLS)). L'approche de routage du protocole dépend de deux modes :

4.3.1 mode glouton (greedy mode)

L'algorithme de routage glouton permet d'acheminer le paquet vers un nœud voisin géographiquement proche du nœud destinataire. GPSR est un exemple caractéristique des protocoles basés sur un algorithme de routage glouton. Chaque nœud dans le réseau maintient dans sa table de voisinage la localisation ainsi que l'identifiant des nœuds dans son voisinage direct. Mentionnons que ces informations géographiques sont collectées grâce aux messages de signalisation envoyés périodiquement dans le réseau. Lors du routage des paquets, le nœud retransmetteur consulte sa table de voisinage et choisit le nœud le plus proche de la destination.

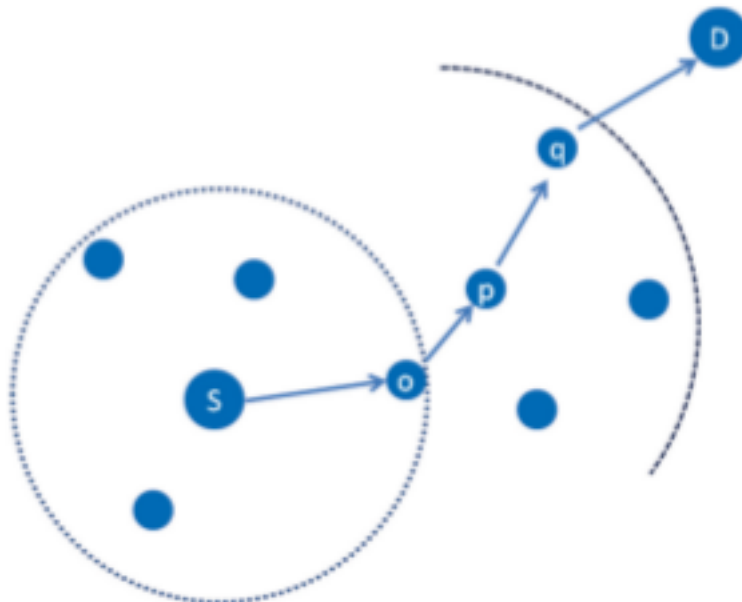


FIGURE 4.2 – Routage glouton du protocole GPSR

Dans la figure 4.2, les nœuds S et D sont respectivement les nœuds source et destinataire. S, d'après l'algorithme glouton, choisit dans sa zone de transmission radio, le nœud le plus proche de D (il choisit alors le nœud o). Le même algorithme se répète entre les

nœuds p et q . Lorsque le paquet atteint le nœud q , celui ci envoie directement au nœud destinataire D [16].

L'avantage génial du routage gourmand est sa dépendance uniquement de la connaissance des voisins immédiats du nœud expéditeur. De plus, les algorithmes traditionnels de routage de plus court chemin ne peuvent pas exploiter la structure des adresses IP pour prendre des décisions d'acheminement. Ils doivent traiter les adresses IP comme des identificateurs plats et recourir à une recherche de table entre toutes les destinations dans le domaine de routage. C'est la nature auto-descriptive des coordonnées géographiques qui permet de transférer les routeurs pour interpréter l'emplacement de destination dans un paquet pour prendre une décision d'acheminement purement locale[17].

Le pouvoir du routage gourmand pour l'itinéraire utilisant uniquement les positions des nœuds voisins comporte un inconvénient majeur : il existe des topologies dans lesquelles le seul itinéraire vers une destination requiert un déplacement de paquets temporairement plus loin géographiquement de la destination. La figure 4.3 illustre un exemple d'une telle topologie. Ici, x est plus proche de D que ses voisins w et y . Encore, l'arc en pointillé autour de D a un rayon égal à la distance entre x et D . Bien que deux chemins, $x \rightarrow y \rightarrow z \rightarrow D$ et $x \rightarrow w \rightarrow v \rightarrow D$, existent à D , x ne choisira pas de passer à w ou y en utilisant un routage gourmand. x est un maximum local à proximité de D . Un autre mécanisme doit être utilisé pour transférer des paquets dans ces situations.

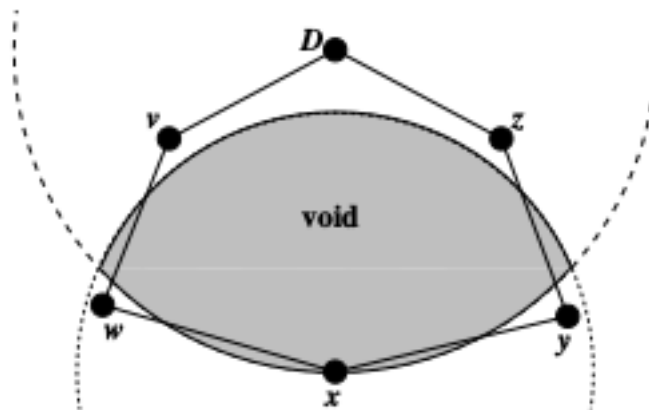


FIGURE 4.3 – Cas d'échec de routage glouton du protocole GPSR

4.3.2 mode de périmètre (perimeter mode)

Mode de périmètre est un mode de récupération utilisé lorsqu'un paquet atteint un maximum local¹. Ainsi, un mode de récupération est utilisé pour transférer un paquet vers

1. Un maximum local est un nœud expéditeur plus proche à la destination que ses voisins

un nœud plus proche de la destination que le nœud où le paquet fait face au maximum local.

Lorsqu'un paquet fait face à un maximum local, i.e. une région vide, la règle de droite (right-hand rule) est appliquée. La règle de droite connue pour parcourir un graphe est illustrée à la figure 4.4. Cette règle indique que, lorsqu'on arrive au nœud x du nœud y , le bord suivant traversé est le suivant séquentiellement dans le sens antihoraire sur x du bord $x y$. On sait que la règle de droite traverse l'intérieur d'une région polygonale fermée (une face) dans le sens des aiguilles d'une montre dans ce cas, le triangle délimité par les bords entre les nœuds x , y et z dans l'ordre $y \rightarrow x \rightarrow z \rightarrow y$. La règle traverse une région extérieure, dans ce cas, la région située à l'extérieur du même triangle, dans le sens des arêtes dans le sens inverse des aiguilles d'une montre[17]. On appelle les arêtes traversées par la règle de droite : un périmètre.

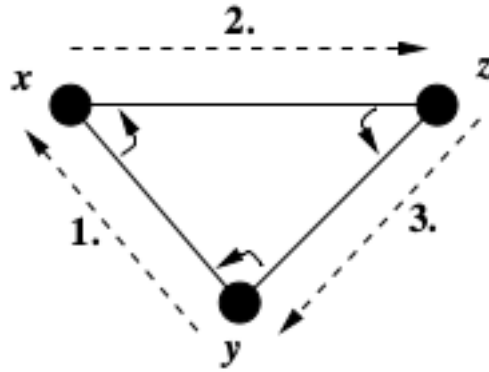


FIGURE 4.4 – La règle de droite (l'intérieur du triangle)

Malheureusement, la règle de droite ne permet pas de traverser le périmètre d'un polygone fermé sur tous les graphes de réseau sans fil. Sur les graphes avec des arêtes qui se croisent, la règle de droite peut plutôt prendre une tour dégénérée des arêtes qui ne trace pas la limite d'un polygone fermé. Les graphes des arêtes croisés sont connus sous le nom de graphes non planaires.

Pour bien appliquer la règle de droite pour un meilleur acheminement de données, il faut rendre les graphes non planaires des graphes planaires, en appliquant les algorithmes RNG (Relative Neighborhood Graph) et GG (Gabriel Graph). L'algorithme RNG annonce que pour u, v, w sont des nœuds : $\forall w \neq u, v : d(u, v) > \max[d(uw), d(v, w)] \Rightarrow$ éliminer l'arête entre u et v . En outre, l'algorithme de GG dit qu'il existe une arête entre u et v s'il n'existe pas un nœud dans le cercle dont le diamètre est $\overline{u, v}$, i.e. $\forall w \neq u, v : d^2(u, v) < [d^2(u, w) + d^2(v, w)]$. Donc pour m point milieu de $\overline{u, v} : d(m, w) < d(m, u) \Rightarrow$ éliminer l'arête entre u et v .

L'algorithme 2 suivant illustre les deux procédures[17] utilisées dans le protocole GPSR :

Algorithme 2 : GPSR pseudo code

```

Procédure GREEDY FORWARDING
     $n_{best} \leftarrow self.a;$ 
    // l'adresse du nœud courant
     $d_{best} \leftarrow DISTANCE(self.l, p.l);$ 
    // la distance entre le nœud courant et la destination
    // N la table de voisins du nœud
    foreach (a,l) in N do
         $d \leftarrow DISTANCE(l, p.l);$ 
        if (a = p.a) OR (d <  $d_{best}$ ) then
             $n_{best} \leftarrow a;$ 
             $d_{best} \leftarrow d;$ 
            if (a = p.a) then
                Break;
    if ( $n_{best} = self.a$ ) then
        Greedy Forwarding Failure;
        Apply Right Hand Rule;
    forward p to  $n_{best}$ ;
    Greedy Forwarding Success;

Function Right Hand Rule
     $b_{in} \leftarrow NORM(ATAN2(self.l.y - n_{in}.y, self.l.x - n_{in}.x));$ 
    // NORM : normalise ses arguments en radian en [0, 2 $\pi$ ], en ajoutant répétitivement 2 $\pi$ 
     $\delta_{min} \leftarrow 3\pi;$ 
    foreach (a,l) in N do
        if (a =  $n_{in}$ ) then
            Continue;
         $b_a \leftarrow NORM(ATAN2(self.l.y - l.y, self.l.x - l.x));$ 
        // ATAN2(x,y) calcule l'arc tangente de y/x dans le quadrant approprié
         $\delta_b \leftarrow NORM(b_a - b_{in});$ 
        if ( $\delta_b < \delta_{min}$ ) then
             $\delta_{min} \leftarrow \delta_b;$ 
             $a_{min} \leftarrow a;$ 
    return  $a_{min};$ 

```

4.4 Intelligent Routing protocol using real time Traffic Information in urban Vehicular environment (IRTIV)

L'objectif principal de ce protocole est de trouver le chemin le plus court connecté dans un réseau dynamique. Dans le cas de la présence d'un véhicule dans une intersection, la décision de routage est prise instantanément par ce véhicule, dans le cadre de protocoles de routage proactifs, sur la base d'un poids calculé en permanence et en temps réel pour chaque segment. Une autre technique de carry & forward² est utilisée dans d'autres cas[18].

Ce protocole assume que chaque véhicule est équipé par un système de positionnement global (GPS) pour déterminer sa position géographique, par un service de localisation (GLS) pour déterminer la position de la destination, ainsi équipé par une carte digitale

2. la stratégie carry & forward : lorsqu'un nœud ne peut pas contacter d'autres nœuds, il stocke le paquet et il l'envoie au prochain nœud rencontré

(digital map) pour localiser les jonctions voisines. On assume aussi que les tables de voisins sont créées et mises à jour par tous les véhicules du réseau.

Ce protocole fonctionne en trois (3) étapes :

4.4.1 Estimation du trafic en temps réel

La densité du trafic est calculée d'une façon distribuée à l'aide des messages Hello modifiés qui permettent de calculer le nombre total de véhicules entre deux (2) intersections et mettre les drapeaux de connectivité.

Le paquet Hello modifié contient des champs additionnels avec le paquet original (4.1). Le premier champ indique la position géographique actuelle du véhicule. Le deuxième présente le nombre de voisins immédiats à gauche (DLN), ensuite le nombre de voisins immédiats à droite (DRN), le nombre total des voisins jusqu'à la prochaine intersection de gauche (VL) et le nombre total des voisins jusqu'à la prochaine intersection de droite (VR). Ces deux derniers champs sont calculés comme suit :

$$VL = DLN + VL_{véhicule-plus-éloigné}$$

$$VR = DRN + VR_{véhicule-plus-éloigné}$$

Il contient aussi deux (2) bits de flags indiquant la connectivité sur la route, un vers l'intersection gauche (Lj) et l'autre vers l'intersection droite (Rj).

TABLE 4.1 – Champs additionnels dans le paquet Hello modifié

Position	DLN	DRN	VL	VR	Lj	Rj
4 octets	1 octet	1 octet	2 octets	2 octets	1 bit	1 bit

Le tableau 4.2 présente la signification des valeurs des bits de flags.

TABLE 4.2 – La signification des flags Lj et Rj

Lj	Rj	Indication
0	0	aucune connectivité sur le chemin
0	1	connectivité sur le chemin droit seulement
1	0	connectivité sur le chemin gauche seulement
1	1	ce chemin est connecté

4.4.2 sélection du chemin

La sélection du chemin est utilisée uniquement sur les intersections car elles sont les seuls endroits où une décision de routage est prise en fonction de l'estimation de la densité de trafic et le plus court chemin vers la destination finale[18]. Lorsqu'un véhicule entre dans une intersection, il calcule un poids pour chaque segment allant de cette intersection, en basant sur la densité du trafic et l'algorithme de Dijkstra. Le poids est calculé par :

$$poids_i = \frac{NbrVéhicules}{D_w} (L_j \times R_j)$$

Où : $NbrVéhicules = (V Lou VR) + 1$ D_w : le poids de Dijkstra en terme de distance

Le segment avec le poids le plus élevé signifie qu'il y a suffisamment de véhicules pour assurer la connectivité et c'est le segment de route le plus proche de la destination, donc il est sélectionné pour acheminer le paquet. La figure 4.5 montre un exemple de sélection du chemin.

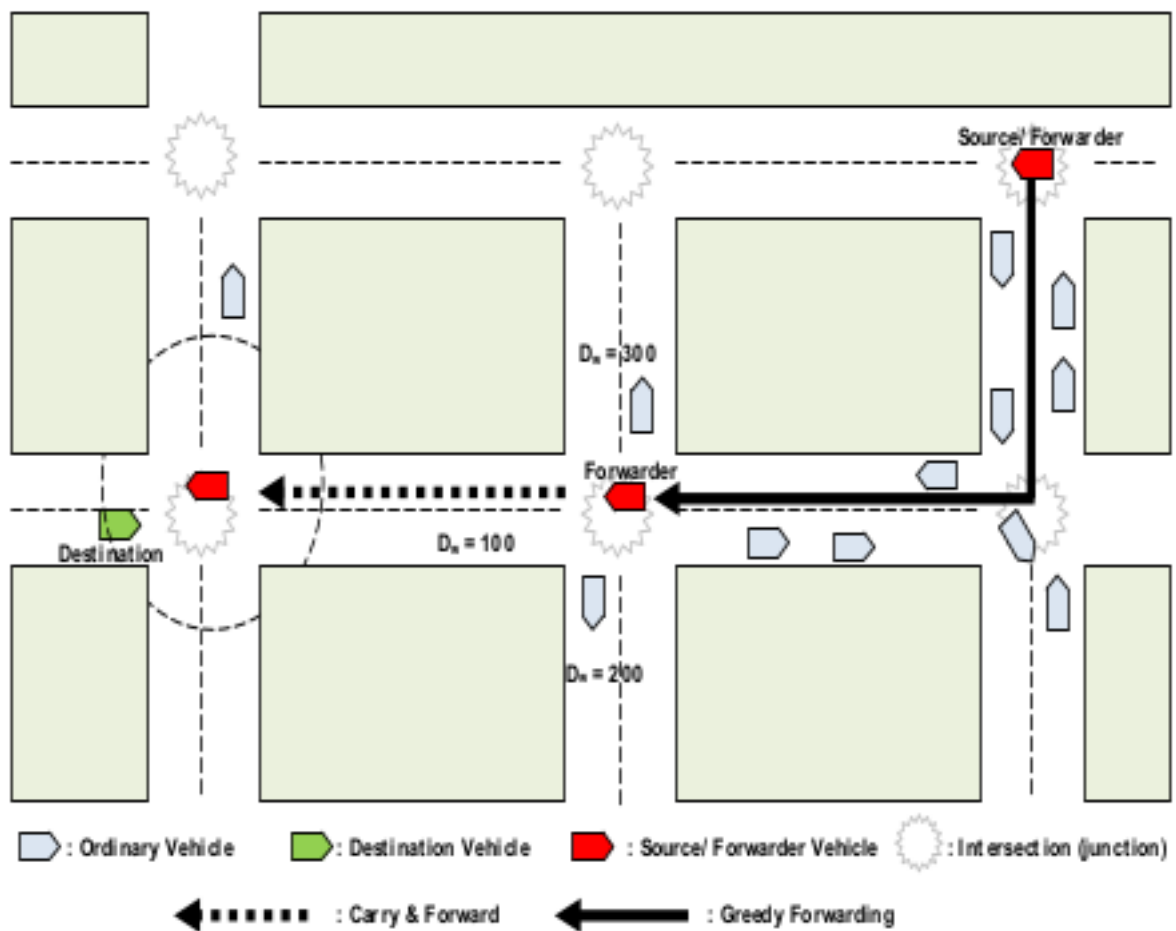


FIGURE 4.5 – Sélection du chemin avec le protocole IRTIV

En cas d'une faible densité de véhicule, les transporteurs sur les intersections n'apprennent pas les informations de connectivité et le nombre de véhicules dans certains segments, donc le véhicule ne possède pas d'informations de densité. Dans ce cas, il base sur le poids de Dijkstra seulement.

4.4.3 Mécanisme de livraison de données

Ce protocole utilise le routage glouton (greedy forwarding) et carry & forward selon la situation du réseau. Il utilise le routage glouton lorsqu'il existe une connectivité sur le chemin entre deux (2) intersections successives. En revanche, Il utilise la stratégie de carry & forward en cas le segment est déconnecté. Un véhicule peut transporter le paquet jusqu'à la prochaine intersection, si sa vitesse³ est vers le plus court chemin à la destination.

L'algorithme 3 [18] illustre le fonctionnement du protocole IRTIV.

Algorithme 3 : IRTIV pseudo code

```

begin
  C ← le véhicule courant;
  D ← le véhicule destinataire;
  J ← la prochaine intersection;
  Nc ← l'ensemble de voisins immédiats de C;
  if (C = D) then
    | paquet reçu avec succès;
  if (D ∈ Nc) then
    | Forward (paquet, D);
  if (Position(C) ∈ zone d'intersection) then
    | foreach segmenti do
    |   | poidsi =  $\frac{NbrVéhicules}{D_w} (L_j \times R_j)$ ;
    |   | J ← max de tout (segmenti, poidsi);
  if (∃ véhicule ∈ Nc) then
    | Greedy Forwarding (paquet, véhicule);
  Carry & Forward (paquet, C);
  Attente de voisins;

```

4.5 Greedy Traffic-Aware Routing (GyTAR)

GyTAR est un nouveau protocole de routage géographique basé sur l'intersection, il est capable de trouver des routes robustes dans les environnements urbains. En utilisant GyTAR, un paquet se déplacera successivement plus près de la destination le long des chemins où il existe suffisamment de véhicules pour assurer la connectivité.

On suppose que chaque véhicule connaît sa position géographique, celle de ses voisins immédiats et celle de la destination (GPS/Beacon/GLS), et est équipé par une carte digitale pour pouvoir localiser les intersections voisines. Ce protocole se compose de deux modules : la sélection des jonctions à travers lesquelles un paquet doit passer pour atteindre sa destination, et un mécanisme d'acheminement gourmand amélioré entre deux jonctions.

3. vitesse : vitesse et direction

4.5.1 Sélection de jonction

GyTAR adopte l'approche de routage basée sur l'ancrage avec la conscience de l'état de la route. Ainsi, les paquets de données seront acheminés entre les véhicules, en suivant la topologie du réseau[19]. Cependant, le paquet doit traverser pour atteindre la destination, les jonctions intermédiaires qui sont choisies dynamiquement et une par une, en tenant compte à la fois de la variation du trafic véhiculaire et de la distance à la destination : lors de la sélection de la prochaine jonction de destination, un nœud (le véhicule émetteur ou un véhicule intermédiaire dans une jonction) recherche la position des jonctions voisines à l'aide de la carte. Un score est attribué à chaque jonction en fonction de la densité du trafic et de la distance vers la destination. La meilleure jonction de destination et celle qui a le score le plus élevé, c'est-à-dire qui a la connectivité la plus élevée, et est la jonction géographiquement la plus proche de la destination. Le score donné à chaque intersection est calculé par la formule suivante :

$$Score(J) = \alpha(1 - D_p) + \beta(\min(\frac{N_{avg}}{N_{con}}, 1))$$

Dont :

J : l'intersection candidate suivante

D_p : la proximité de l'intersection candidate avec la destination, calculée par $D_p = \frac{D_j}{D_i}$

tel que :

D_j : la distance de l'intersection candidate J à la destination

D_i : la distance de l'intersection courante I à la destination

I : l'intersection courante

N_{avg} : le nombre moyen de véhicules par cellule, calculé par : $N_{avg} = \frac{N_v}{N_c}$

N_v : le nombre de véhicules entre I et J

N_c : le nombre de cellules entre I et J

N_{con} : constante représente le degré idéal de connectivité

α, β : coefficients de pondération ($\alpha + \beta = 1$)

4.5.2 Routage de données entre deux jonctions

Une fois la jonction de destination est déterminée, la stratégie de greedy améliorée est utilisée pour transférer des paquets entre les deux jonctions impliquées. Pour cela, tous les paquets de données sont marqués par l'emplacement de la prochaine jonction. Chaque véhicule maintient une table de voisins dans laquelle la position, la vitesse et la direction de chaque véhicule voisin sont enregistrées. Ce tableau est mis à jour par l'intermédiaire de messages Hello envoyés périodiquement par tous les véhicules. Ainsi, lorsqu'un paquet est reçu, le véhicule expéditeur calcule la nouvelle position prédite de chaque voisin à

l'aide des informations enregistrées (vitesse, direction et dernière position connue), puis sélectionne le voisin du saut suivant (le plus proche de la jonction de destination)[19].

4.5.3 Stratégie de rétablissement

Malgré l'amélioration de la stratégie de greedy, le risque demeure qu'un paquet se bloque dans un optimum local (Le véhicule expéditeur pourrait être le plus proche de la prochaine jonction). Une stratégie de rétablissement est donc nécessaire basée sur l'idée de carry & forward : le véhicule transporte le paquet jusqu'à l'intersection destinataire, ou jusqu'à ce qu'un autre véhicule, plus proche de l'intersection cible, atteigne sa portée.

Le pseudo code 4 donné montre le principe du protocole.

Algorithme 4 : GyTAR pseudo code

```

begin
  i ← le véhicule courant;
  p ← le véhicule destinataire;
  JNi ← l'ensemble d'intersections voisines de i;
  Ni ← l'ensemble de voisins immédiats de i;
  if (i = p) then
    | paquet reçu avec succès;
  if (p ∈ Ni) then
    | Forward (paquet, p);
  foreach junctionj ∈ JNi do
    | Score(J) = α(1 - Dp) + β(min( $\frac{N_{avg}}{N_{con}}$ , 1));
  J ← max(scoreJ);
  if (∃ véhicule ∈ Ni) then
    | Greedy Forwarding (paquet, véhicule);
  Carry & Forward (paquet, i);
  Attente de voisins;

```

4.6 Spatial and Traffic-Aware Routing (STAR)

L'utilisation de l'approche de connaissance de la topologie qui limite les déplacements des véhicules signifie qu'un véhicule expéditeur peut voir où transmettre les paquets. Bien que la connaissance de topologie soit un grand avantage, cette approche peut échouer en cas où il n'y a aucun véhicule qui se bouge sur les routes, ces routes seront considérées comme des liens brisés. Ce problème peut être résolu par la connaissance de topologie réelle, i.e. transférer les paquets par les routes où le trafic de véhicules existe. Cette connaissance peut être obtenue grâce à la coopération entre les véhicules afin de détecter des situations anormales dans la distribution des véhicules sur les routes, et de recueillir et de diffuser des informations sur la topologie réelle et l'état du réseau en temps réel[20].

La collecte et l'échange d'informations sur la topologie peuvent être coûteuses, et ces informations sont très volatiles à cause de la mobilité. Pour minimiser ce coût, il est préférable que chaque nœud rassemble une connaissance partielle du réseau concernant une zone restreinte autour de sa position. De même il est convenable de calculer seulement

un chemin partiel pour approcher la position de la destination, en déterminant un sous-ensemble de points d'ancrage (PA). Lorsqu'un paquet arrive au dernier PA calculé pour celui-ci, le nœud responsable de l'acheminement prend en charge la caractérisation des PA suivants.

Le protocole STAR est organisé en deux (2) couches : une couche inférieure qui gère la collecte et l'échange d'informations sur l'état du réseau, et une couche supérieure pour le calcul des chemins.

4.6.1 Surveillance du trafic

La couche inférieure est chargée de trouver et diffuser des informations sur l'état du réseau. On s'intéresse à détecter deux (2) cas : la présence d'une file de véhicules, ou l'absence totale des véhicules.

Les routes où les files de véhicules doivent être préférées car elles fournissent plusieurs alternatives pour le transfert des paquets, et éviter les routes où les véhicules ne sont pas présents car les paquets ne peuvent pas être acheminés.

La propagation de ces informations est effectuée par l'échange de beacon transportant des informations des nœuds du quartier dans des structures de données gérées par le module de surveillance du trafic.

Chaque nœud a une table de trafic. Chaque entrée dans cette table comporte cinq (5) champs pour suivre les conditions du trafic, à savoir : position, direction, bit de trafic (Tbit), bit déjà envoyé (ASbit) et durée de vie (TTL). La position indique les coordonnées où une situation de trafic a été enregistrée. La direction est la direction dans laquelle l'état de la circulation se déroule en fonction de la position. Tbit spécifie le type de trafic (haut ou bas). ASbit enregistre si une entrée de trafic a déjà été propagée aux voisins. TTL est le nombre de sauts que l'information doit être envoyée et diminue chaque fois que l'entrée est renvoyée. Chaque entrée a un trafic-temporisateur associé. Lorsque le temporisateur expire, l'entrée est supprimée de la table de trafic afin d'oublier les informations obsolètes sur les anomalies de trafic qui n'existent plus[20].

Un nœud maintient une table des voisins avec la position de ses voisins (latitude, longitude, altitude) découverts via le beacon. Dans le vecteur de présence (PRV) un nœud maintient quatre (4) compteurs de nœuds qui représentent le nombre des voisins qu'il a vers les points cardinaux (nord, est, ouest, sud) calculés en divisant la cellule du nœud en secteurs. Si un élément du PRV est supérieur à un tel seuil supérieur HighPR, une concentration élevée de véhicules existe dans la direction correspondante. Ainsi, si un élément du PRV est inférieur à un tel seuil inférieur LowPR, un trafic de véhicules rare dans une route située dans la direction correspondante. Pour l'un des cas, une modification d'un élément apparenté dans le vecteur de persistance (PEV) est déclenchée. le vecteur PEV a quatre (4) éléments comme PRV, chacun peut être dans une de trois (3) conditions :

valeur = 0, valeur < 0 et valeur > 0. Ces valeurs sont mises à jour selon certaines conditions comme le montre l'algorithme suivant :

Algorithme 5 : MAJ du PEV

```

begin
  for ( $i \in north, east, south, west$ ) do
    if ( $PRV[i] < lowPR$ ) then
      if ( $PEV[i] \leq 0$ ) then
         $PEV[i] \leftarrow PEV[i] - 1;$ 
       $PEV[i] \leftarrow 0;$ 
    if ( $PRV[i] > highPR$ ) then
      if ( $PEV[i] \leq 0$ ) then
         $PEV[i] \leftarrow PEV[i] + 1;$ 
       $PEV[i] \leftarrow 0;$ 
     $PEV[i] \leftarrow 0;$ 

```

Chaque nœud diffuse périodiquement à ses voisins un beacon de couche réseau qui contient l'identifiant de l'expéditeur, les coordonnées de l'expéditeur et les conditions de circulation des véhicules qu'il a dans sa table de trafic. La durée de diffusion est déterminée par le temporisateur du beacon. À la réception du beacon, le nœud récepteur met à jour sa table de voisins par l'identifiant et la position de l'expéditeur. Ensuite, met à jour sa table de trafic si nécessaire selon les vecteurs PEV et PRV, dont lorsque la valeur de PEV sort de l'intervalle $[LowPE, HighPE]$ les informations sur le trafic seront enregistrées sur la table de trafic, et le vecteur PEV sera initialisé. En effet, il est utilisé pour enregistrer des situations critiques uniquement lorsqu'elles durent longtemps.

4.6.2 Routage des paquets

Dans la couche supérieure, les itinéraires sont calculés à la demande exploitant les informations détenues sur la circulation et les voisins. Lorsqu'une source S a un paquet à envoyer à une destination D, tout d'abord S construit un graphe pondéré à l'aide de la carte et des informations du trafic. Les arcs correspondants aux routes sans circulation ont associé un poids élevé, de manière à décourager leur utilisation. Et les arcs correspondants aux routes de forte circulation ont associé un poids diminué afin de privilégier le choix de cette route. En raison de la mobilité des véhicules, les poids des arcs sont ajustés dynamiquement.

L'algorithme de Dijkstra est appliqué sur ce graphe pour trouver le plus court chemin. Les PA sont calculés le long des routes appartenant à l'itinéraire pour forcer le paquet de se déplacer le long d'eux. L'en-tête du paquet comprend l'identifiant et la position de la destination, et un nombre limité d'PA égal à $maxPA$, la valeur de $maxPA$ doit être en relation avec $maxTTL$ [20]. Ensuite, le paquet est transmis avec greedy forwarding vers le premier PA, lorsqu'il est atteint, le paquet sera transmis vers le PA suivant. Lorsque le dernier PA calculé a été atteint, le nœud qui doit rediriger le paquet va calculer d'autres PA vers la destination.

Un pseudo code peut résumer tout ce que précède dans l'algorithme 6 [20] :

Algorithme 6 : STAR pseudo code

```

begin
  while (beacon received) do
    update neighbors table;
    update PRV;
    update PEV;
    // Traffic information generation
    for (i ∈ north, east, south, west) do
      if (PEV[i] > highPE or PEV[i] < lowPE) then
        if (traffic information is already present in Traffic Table) then
          remove traffic information;
        create new traffic information (TI);
        TI.position ← node.position;
        TI.TTL ← maxTTL;
        TI.direction ← i;
        if (PEV[i] > highPE) then
          TI.Tbit ← high;
        if (PEV[i] < lowPE) then
          TI.Tbit ← low;
          TI.ASbit ← 0;
      merge traffic entries in beacon with traffic table entries;
    while (beacon timer expires) do
      create new beacon;
      put node identifier and position in beacon;
      foreach entry in traffic table do
        if (!ASbit and TTL > 0) then
          add traffic table entry to beacon;
          ASbit ← True;
      broadcast beacon;
    while (data packet forwarding) do
      if (packet is for me) then
        send packet to upper layer;
      if (current AP is now reached) then
        remove it from the packet;
      if ((no APs in packet header) or (local maximum)) then
        build updated graph from map and traffic table;
        recalculate next APs and write them in packet header;
      if (local maximum) then
        DROP packet;
      greedy forward packet to a neighbor;

```

4.7 Comparaison entre les protocoles étudiés

La conception d'un protocole de routage efficace qui peut fournir un paquet dans les plus brefs délais et avec peu de paquets abandonnés est un défi majeur dans la technologie VANET tenant en compte la mobilité élevée des nœuds et des changements topologiques rapides. Plusieurs chercheurs se sont concentrés sur la conception d'un protocole de routage adapté à des environnements très denses dans lesquels de nombreux véhicules ont des distances étroites entre eux.

Dans le tableau 4.3, nous présentons une comparaison des protocoles de routage VANET présentés dans les sections précédentes. À partir de l'étude de ces protocoles, on peut les comparer selon certains critères : la stratégie du routage qui décrit la première décision de routage lorsque certains paquets doivent être renvoyés. Ainsi, la classification

du protocole, le besoin d'une connaissance sur le trafic sur le réseau, les différents environnements de communication où le protocole fonctionne efficacement, et enfin le besoin d'une carte digitale.

TABLE 4.3 – Comparaison entre les protocoles de routage étudiés

Protocole	Stratégie de routage	Type	Sous-type	En-tête	Connaissance sur le trafic	Scénario	carte digitale
AODV	Multi-sauts	Basé sur la topologie	Réactif	État du chemin	Non	Ville	Non
GPSR	Greedy	Basé sur la position	Non-DTN, Non-Overlay	Beacon	Non	Auto-routes	Oui
IRTIV	Greedy	Basé sur la position	Non-DTN, Non-Overlay	Beacon	Oui	Ville	Oui
GyTAR	Greedy	Basé sur la position	Non-DTN, Overlay	Beacon	Oui	Ville	Oui
STAR	Greedy	Basé sur la position	Non-DTN, Overlay	Beacon	Oui	Ville	Oui

4.8 L'outil d'identification des facteurs (méthode 2^k)

Généralement, le processus de conception d'un protocole de routage efficace est gouverné par certains facteurs. Pour cela, on a besoin de déterminer ces facteurs, et leurs impacts sur la performance du routage. Ainsi, déterminer les plus influents pour concentrer les tests de recherches sur ces facteurs, évitant donc des simulations inutiles et réduire le temps de simulation requis.

4.8.1 Les facteurs définis dans la littérature du VANET

Certains travaux antérieurs ont étudié les facteurs les plus importants dans les MANET. Néanmoins, les VANET ont des caractéristiques particulières qui les rendent différentes des MANET. Par conséquent, une plus grande recherche est requise afin d'identifier

les facteurs clés qui influent sur leur performance. Dans cette section, nous décrivons les facteurs les plus importants :

4.8.1.1 Densité du réseau

Dans VANET, le nombre de véhicules peut être particulièrement élevé, ce qui entraîne habituellement que les simulations VANET nécessitent beaucoup de temps pour finir. En outre, ce facteur semble être important pour mesurer les performances de routage dans les scénarios VANET.

4.8.1.2 Vitesse des véhicules

Les réseaux VANETs sont caractérisés par la mobilité élevée des nœuds, qui peuvent déplacer d'un endroit à un autre par une vitesse instable. Cette vitesse peut changer la topologie du réseau, ainsi influe d'une façon ou d'une autre sur le routage.

4.8.1.3 Largeur du canal

Dans les communications V2V ou/et V2I, l'efficacité de l'utilisation des canaux est importante dans la gestion des transmissions des messages. L'utilisation efficace des canaux permet de réduire le niveau général d'interférence et, à son tour, d'avoir une incidence sur les performances de la réception des messages.

4.8.1.4 Schéma de diffusion

Un autre facteur important dans les VANET est le schéma de diffusion sélectionné. Dans les VANET, les véhicules intermédiaires agissent comme des relais pour supporter les communications de véhicules de bout en bout. Pour les applications telles que la planification des itinéraires, le contrôle de la congestion du trafic et la sécurité routière, l'inondation des messages diffusés se produit habituellement. Par conséquent, l'impact de ce facteur devient important.

4.8.1.5 Priorité des messages

Les messages d'application sont classés dans différentes classes d'accès (AC), où AC0 a la plus basse et AC3 la plus haute priorité. Par exemple, les messages d'avertissement qui contiennent des informations sur des situations anormales telles que des accidents ont toujours la priorité la plus élevée (AC3), tandis que les beacons contenant des informations telles que les positions et les vitesses des véhicules ont une priorité basse.

4.8.1.6 Périodicité des messages

Les véhicules informent les uns les autres de leurs statuts en envoyant des messages périodiquement. Les véhicules participent à la diffusion de ces paquets en transmettant des beacons périodiques avec des informations telles que leurs positions, leur vitesse, etc. Ainsi, la périodicité des messages semble être un facteur important qui offre un compromis entre performance et en-tête.

4.8.1.7 Modèle de mobilité

L'un des défis posés par l'étude des VANET est la définition d'un modèle de mobilité des véhicules offrant une description de la mobilité des véhicules précise et réaliste aux niveaux macroscopiques et microscopiques[21].

4.8.1.8 Modèle de propagation

Le modèle de propagation offre une précision suffisante pour les environnements de véhicules. En particulier, les obstacles physiques présents dans les environnements urbains (principalement les immeubles), ou même les montagnes. Donc, ce facteur prend tout son importance.

4.8.1.9 Roadmap simulé

La carte routière (topologie routière) est un facteur important pour la mobilité dans les simulations, car la topologie limite les mouvements des voitures. D'une manière approfondie, une topologie urbaine est un graphe où les sommets et les arêtes représentent, respectivement, les éléments de jonction et de route.

4.8.2 L'analyse factorielle 2^k

Dans la littérature du réseau, nous pouvons trouver plusieurs travaux qui ont adopté l'approche factorielle 2^k pour discriminer les nombreux paramètres disponibles afin de déterminer ,parmi eux ,les plus pertinents.

Les simulations VANET impliquent souvent des scénarios importants et hétérogènes. Le nombre de facteurs possibles et leurs valeurs, ou leurs niveaux, peuvent être très importants. Dans ce que suit, nous expliquerons comment l'analyse factorielle 2^k peut être utilisée pour déterminer les facteurs les plus pertinents qui régissent les performances d'un système.

L'utilisation de l'analyse factorielle 2^k est importante pour plusieurs raisons : (i) réduire le nombre total de simulations nécessaires ; (ii) évaluer la relation entre différents

facteurs ; Et (iii) réduire le temps de simulation requis. L'approche de base de cette méthode est basée sur la sélection d'un ensemble de k paramètres et la détermination de 2 niveaux extrêmes (marqués avec -1 et 1). Une expérience est exécutée pour toutes les $\binom{k}{2}$ combinaisons possibles des paramètres. De chaque expérience, nous pouvons également extraire les interactions $\binom{k}{2}$ à deux facteurs, les $\binom{k}{3}$ interactions à trois facteurs, etc[21]. Ensuite, on construit la matrice de signes qui représente les expériences exécutées pour toutes les combinaisons. On explique et détaille cette approche dans l'exemple[21] suivant.

4.8.2.1 Exemple applicatif

Supposons que nous avons proposé un système de diffusion des messages d'avertissement et que nous voulons étudier l'impact de la densité du réseau (facteur A) et la vitesse des véhicules (facteur B) sur le temps de notification d'avertissement, c'est-à-dire le temps requis par les véhicules normaux pour recevoir un message d'avertissement envoyé par un véhicule en mode d'avertissement.

Le tableau 4.4 montre les différentes expériences définies par la conception 2^2 , et le tableau 4.5 montre les résultats obtenus après les simulations.

TABLE 4.4 – Les expériences définies par la conception 2^2

expériences	A	B	y
1	-1	-1	y_1
2	1	-1	y_2
3	-1	1	y_3
4	1	1	y_4

TABLE 4.5 – Exemple de résultats obtenus en termes de temps de notification d'avertissement variant deux facteurs.

Densité de véhicules	Vitesse 10km/h	Vitesse 80km/h
25veh/km ²	1s	0.8s
150veh/km ²	0.5s	0.4s

Dans une analyse factorielle 2^k , en utilisant la méthode de la table des signes, nous pouvons obtenir les résultats et détecter les variations qui dépendent de la combinaison des facteurs. Pour une conception 2^2 , les effets peuvent être facilement calculés en préparant une matrice de signes 4×4 comme indiqué dans le tableau 4.6. La première colonne de la matrice est appelée I et tous ses éléments sont égaux à 1. Les deux colonnes suivantes, intitulées A et B, contient essentiellement toutes les combinaisons possibles de -1 et 1 .

La quatrième colonne, étiquetée AB, est le produit des entrées dans les colonnes A et B. Les quatre observations sont répertoriées dans un vecteur de colonne à côté de cette matrice. Le vecteur de colonne est marqué y et se compose des résultats correspondant aux niveaux de facteurs répertoriés sous les colonnes A et B. L'étape suivante consiste à multiplier les entrées dans la colonne I par celles de la colonne y et à mettre leur somme sous la colonne I . Les entrées dans la colonne A est maintenant multipliée par celles de la colonne y et la somme est saisie sous la colonne A. Cette opération de multiplication des colonnes est répétée pour les deux autres colonnes de la matrice. Les sommes sous chaque colonne sont divisées par 4 pour donner les coefficients correspondants du modèle de régression.

TABLE 4.6 – Tableau des signes pour calculer les effets des facteurs dans une conception 2^2

I	A	B	AB	y
1	-1	-1	1	$1s$
1	1	-1	-1	$0.5s$
1	-1	1	-1	$0.8s$
1	1	1	1	$0.4s$
2.7	-0.9	-0.3	0.1	total
0.675	-0.225	-0.075	0.025	total/4

L'importance d'un facteur dépend de la proportion de la variation totale métrique expliquée par le facteur. La variation totale de y est également connue sous le nom Sum of Squares Total (SST), qui peut être calculée comme suit :

$$\text{Variation totale de } y = SST = \sum_{i=1}^{2^k} (y_i - \bar{y})^2 = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$$

$$\bar{y} = \sum_{i=1}^{2^k} y_i$$

On peut donc calculer la fraction de variation expliquée par chaque facteur :

$$\text{Fraction de variation expliquée par A} = \frac{SSA}{SST} = \frac{2^2 q_A^2}{SST}$$

Indiquant que q sont les coefficients correspondants du modèle de régression.

Par conséquent, nous pouvons indiquer le pourcentage de variation de chaque métrique étudiée expliquée par chaque facteur. Plus le pourcentage de variation est élevé, plus l'impact de ce facteur est important dans la métrique mesurée. Dans notre exemple, nous avons constaté que la densité des véhicules représente 89,01% (soit $\frac{2^2(-0.225)^2}{0.2275}$) de la variation totale du temps de notification d'avertissement, la vitesse des véhicules représente 9,89% (c-à-d. $\frac{2^2(-0.075)^2}{0.2275}$) et leurs combinaison compte le reste 1,10% (c-à-d. $\frac{2^2(0.025)^2}{0.2275}$). Par

conséquent, dans notre exemple sélectionné, la densité des véhicules est le facteur le plus important qui affecte le temps de notification d'avertissement.

Remarque : pour les facteurs où il n'y a pas des valeurs exactes (comme le modèle de propagation, le roadmap simulé, etc.) les deux niveaux extrêmes de la méthode proposée, sont pris par deux (2) scénarios opposés.

L'algorithme 7 illustre la méthode expliquée :

Algorithme 7 : algorithme χ^2

```

var m : tableau de réels[1..l][1..l];
ft : tableau de réels[1..2][1..l];
y : tableau de réels[1..l];
prcssf : tableau de réels[1..l];
f : fichier;
i,k,l : integer;
// fonction retourne le factoriel d'un nombre entier
Function factoriel (n : integer) : integer
  var a, fact : integer;
  fact ← 1;
  for a = 1 to n do
    fact ← fact * a;
  return fact;

// procédure remplit la matrice par les données
Procédure remplir-matrice (m : tableau réels[1..l][1..l], ft : tableau réels[1..2][1..l])
  var i, j, r, p, p1, p2, p3, p4, p5, p6, p7, p8, o, c2, c3, c4, c5, c6, c7, c8, c9, c10 : integer;
  // remplir la 1ère colonne par des uns
  for i = 1 to l do
    m[i, 1] ← 1;

  // remplir jusqu'à la kième colonne par des valeurs extrêmes des facteurs
  for j = 2 to k do
    for r = 1 to ((2j) - 1) + 2 pas do
      for i = ((l * 2)/(2j)) to ((l * (r + 1))/(2j)) do
        m[i, j] ← ft[1, j];
        // valeur min du facteur
      for i = ((l * (r + 1))/(2j)) to ((l * (r + 2))/(2j)) do
        m[i, j] ← ft[2, j];
        // valeur max du facteur

  j ← k + 1;
  // remplir la matrice par les combinaisons de 2
  c2 ← factoriel(k)/(factoriel(2) * factoriel(k-2));
  for p = 2 to k - 1 do
    for o = p + 1 to k do
      if j ≤ k + c2 then
        for i = 1 to l do
          m[i, j] ← m[i, p] * m[i, o];
        j ← j + 1;

  // remplir la matrice par les combinaisons de 3
  if k > 2 then
    j ← k + c2 + 1; c3 ← factoriel(k)/(factoriel(3) * factoriel(k-3)); for p = 2 to k - 2 do
      for p1 = p + 1 to k - 1 do
        for o = p1 + 1 to k do
          if j ≤ k + c2 + c3 then
            for i = 1 to l do
              m[i, j] ← m[i, p] * m[i, p1] * m[i, o];
            j ← j + 1;

  // ... jusqu'à 10

// procédure calcule selon la méthode  $\chi^2$ 
Procédure calcul (m : tableau réels[1..l][1..l], y, prcssf : tableau réels[1..l])
  var i, j : integer;
  t, q, qcssf : tableau de réels[1..l];
  sst : réel;
  for j = 1 to l do
    t[j] gets 0;
    for i = 1 to l do
      t[j] ← t[j] + m[i, j] * y[i];
    q[j] ← t[j]/l;

  // calcul de Sum of Square Total
  for i = 2 to l do
    qc[i] ← q[i]2;

  for i = 2 to l do
    sst ← sst + qc[i];

  sst ← sst * l;
  // calcul de Sum of Square of Factor
  for i = 1 to l - 1 do
    ssf[i] ← (l * qc[i])/sst;

  // calcul de pourcentage d'influence
  for i = 1 to l - 1 do
    prcssf[i] ← ssf[i] * 100;

```

```
begin
  lire(k);
  l ← k2;
  ouvrir(f,lecture);
  while !EOF(f) do
    [ y[i] ← lire(f, e);
  remplir-matrice(m,ft);
  calcul(m,y,pcssf);
```

4.9 Conclusion

Dans ce chapitre, nous avons étudié les protocoles de routage unicast des VANETs. Ainsi, nous avons décrit les différents facteurs à prendre en compte lors de la simulation d'un tel protocole dans les réseaux VANETs. Tandis que le nombre de facteurs possibles peut être très important, nous avons identifié les facteurs représentatifs en utilisant l'analyse factorielle 2^k (la méthode χ^2). L'objectif est de réduire le temps de simulation requis dans les travaux de recherche futurs. Dans le chapitre suivant, On va appliquer cette méthode sur les observations obtenues après les simulations des protocoles de routage mentionnés dans les sections précédentes, puis analyser et interpréter les résultats obtenus de l'application.

Chapitre 5

Résultats et discussion

5.1 Introduction

La simulation connaît de nos jours un essor considérable. Ceci est dû aussi bien à l'intérêt théorique que présente la modélisation des systèmes simulés, que par les besoins croissants de simuler par ordinateur des réalisations de plus en plus complexes, très coûteuses et leurs résultats prennent beaucoup de temps. Donc, la simulation joue un rôle important en mettant à la disposition de l'utilisateur un environnement expérimental où on peut varier plusieurs paramètres, mais pour qu'elle soit utile, elle doit être réaliste le plus possible.

Pour évaluer les performances des protocoles de routage en variant certains paramètres, on va effectuer une simulation avec le simulateur **NS-2**, puis extraire les observations et appliquer sur lesquelles la méthode 2^k , afin d'identifier les paramètres les plus influents sur le système de routage.

Ce chapitre a pour objectif de présenter les résultats obtenus après la simulation et l'application de la méthode 2^k . D'abord nous présentons l'outil de simulation utilisé, ainsi les résultats de la simulation de chacun des protocoles de routage choisis précédemment. Nous présentons ensuite, les résultats de l'application 2^k sur les valeurs des observations de la simulation.

5.2 L'outil de simulation

NS-2[22] est un outil logiciel de simulation de réseaux informatiques à événements discrets, il est parmi les simulateurs les plus utilisés dans les laboratoires de recherche, afin de simuler et étudier les performances des protocoles réseau. Il offre une plateforme de développement de nouveaux protocoles et permet de les tester. Il se base sur deux langages de programmation, C++ qui est un langage compilé très efficace pour l'implémentation des protocoles, et OTCL (Object Tools Command Language) qui représente l'extension orienté objet du langage interprété TCL (Tool Command Language). (Plus de détails cf. Annexe A)

5.3 Modélisation de la simulation et scénarios

Dans cette section, nous allons définir les paramètres de simulation pour étudier les cinq (5) protocoles de routage tels que : AODV, GPSR, IRTIV, GyTAR et STAR, en utilisant l'outil de simulation **NS-2**. Pour cela, on a simulé ces protocoles en variant certains paramètres tels que : le nombre de véhicules, le protocole de routage, le modèle de propagation et le roadmap. Pour cet objectif, on a utilisé CityMob for Roadmaps (C4R)[23], un générateur de mobilité qui peut importer des cartes réelles (maps) directement de OpenStreetMap (OSM), pour obtenir les scénarios aléatoires de mobilité. À chaque simulation on a varié le protocole de routage (AODV, GPSR, IRTIV, GyTAR, STAR), le modèle de propagation (Free Space, Two Ray Ground), le roadmap simulé (city, highway) et le nombre de véhicules (50, 100, 150, 200, 250, 300), en fixant les coordonnées de la carte en 3000 x 3000. Chacune des simulations dure 200 secondes.

Les résultats obtenus de la simulation sont en moyenne de 48 simulations. On vise à obtenir le délai moyen de livraison des paquets de bout en bout, le taux de délivrance des paquets (PDR) qui est calculé en divisant le nombre de paquets bien reçus par le nombre de paquets transmis.

Le tableau 5.1 résume la configuration de notre simulation et les paramètres utilisés :

TABLE 5.1 – Paramètres de simulation

Paramètre	Valeur
Simulateur	NS-2
taille du roadmap	3000 x 3000
Modèle de propagation	Free Space, Two Ray Ground
Roadmap	City, Highway
Nombre de nœuds	50, 100, 150, 200, 250, 300
Portée de transmission	300 m
Temps de simulation	200 s

5.4 Résultats de simulation et analyse de performances

Dans cette section, nous présentons les résultats obtenus de la simulation, ainsi leur interprétation pour les deux (2) métriques :

5.4.1 Délai moyen de bout en bout (End-to-End Delay)

Dans cette expérimentation, on veut comparer le délai moyen de délivrance d'un paquet d'un nœud source vers un nœud destinataire pour les cinq protocoles de routage dans une zone urbaine et une autoroute pour deux modèle de propagation FreeSpace et TwoRayGround. La figure 5.1 montre que le délai moyen de délivrance des paquets d'AODV est petit lorsqu'il y a peu de véhicules, et ce délai vient d'augmenter en croissant le nombre de véhicules atteignant 0.55 secondes. Tandis que le délai de délivrance des paquets IRTIV varie entre 0.01 et 0.07 secondes pour toutes les densités du réseau. Par ailleurs, le délai obtenu du protocole GPSR varie entre 0.07 et 0.12 secondes. Le délai de délivrance des paquets GyTAR diminue en augmentant le nombre de véhicules, contrairement au protocole STAR, où le délai s'augmente. L'analyse de ces courbes montre que le protocole IRTIV est plus efficace en terme de délai puisqu'il fait le moindre pour délivrer les paquets, puis GPSR le célèbre adapté aux VANETs. En revanche, AODV fait plus de temps lorsque le réseau devient dense, ce qui peut produire des collisions et retarder la délivrance des paquets. Pour la même raison le délai de STAR s'augmente. Ainsi, la diminution du délai de GyTAR peut être expliquée par la connectivité élevée du réseau car il choisit le segment le plus connecté. Ces résultats sont obtenus pour les deux modèles de propagation.

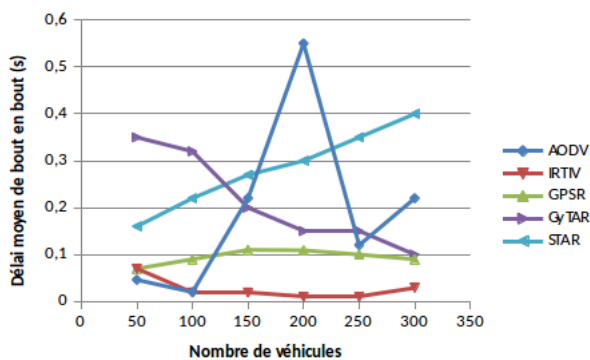


FIGURE 5.1 – Délai moyen de bout en bout en fonction de la densité dans une zone urbaine

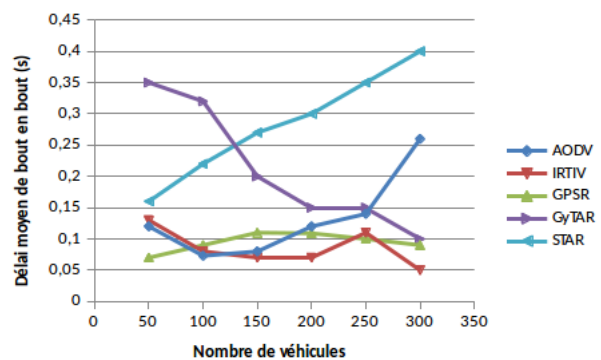


FIGURE 5.2 – Délai moyen de bout en bout en fonction de la densité dans une autoroute

La figure 5.2 présente les résultats sur une autoroute pour les deux modèles de propagation. le délai obtenu d'AODV varie entre 0.07 et 0.26 secondes, et celui d'IRTIV varie entre 0.05 et 0.14 secondes. Les délais des autres protocoles restent les mêmes. La courbe d'AODV signifie qu'il existe une connectivité élevée sur une autoroute donc une délivrance plus rapide des paquets. Mais, celle d'IRTIV reste inexplicable puisqu'on devrait avoir un délai plus petit sur une autoroute.

5.4.2 Taux de délivrance des paquets (Packets Delivery Ratio, PDR)

Nous avons choisi de comparer la performance des cinq protocoles entre eux en terme de taux de délivrance des paquets (PDR) pour le cas d'une ville et une autoroute. La figure 5.3 montre les résultats dans une ville en modèle FreeSpace. Cette figure montre que le PDR d'AODV s'augmente avec l'augmentation de nombre de véhicules. Ainsi, le PDR d'IRTIV se varie entre 0.07 et 0.49. Par ailleurs, le taux des protocoles GPSR et STAR se diminue en croissant le nombre de véhicules. En revanche, les paquets sont bien délivrés en GyTAR lorsque le nombre de véhicules s'augmente. On sait bien que pour une perte des paquets on aura un délai élevé en raison la retransmission. Donc, ces courbes montrent une contradiction, d'où on devrait avoir un PDR élevé d'IRTIV puisqu'on a un délai réduit. Mais ce n'est pas le cas d'AODV, dont on a obtenu un PDR élevé dans un délai élevé en raison de la retransmission. De plus, le PDR de GPSR reste inexplicable comparablement avec le délai. Pour les autres protocoles, les résultats semblent plus logiques, dont GyTAR délivre plus de paquets en un délai réduit en croissant le nombre de véhicules en raison de la stratégie de rétablissement, et STAR perd des paquets en raison de la collision.

Les résultats dans une ville pour un modèle TwoRayGround sont identiques avec ceux du modèle FreeSpace. La seule différence est en la courbe d'IRTIV, où le PDR diminue variant entre 0.12 et 0.33. Cette diminution peut être expliquée par le changement du modèle de propagation, mais ça reste illogique et inexplicable d'avoir un PDR réduit d'IRTIV par rapport aux autres protocoles surtout AODV.

Sur une autoroute, on a obtenu des résultats différents pour les protocoles AODV et IRTIV. La figure 5.5 montre que le PDR d'AODV varie entre 0.58 et 0.93 et celui d'IRTIV varie entre 0.4 et 0.88. Le PDR élevé d'AODV est logique comparablement avec le délai, mais celui d'IRTIV n'est pas, comparant ces courbes avec les courbes du délai de la figure 5.2, on peut pas les expliquer parce qu'elles montrent une contradiction. De plus, on peut pas avoir un PDR d'AODV plus grand qu'IRTIV. Par ailleurs, on a obtenu les mêmes résultats pour le modèle TwoRayGround, mais avec un PDR d'IRTIV plus diminué. Malheureusement, ces résultats restent inexplicables.

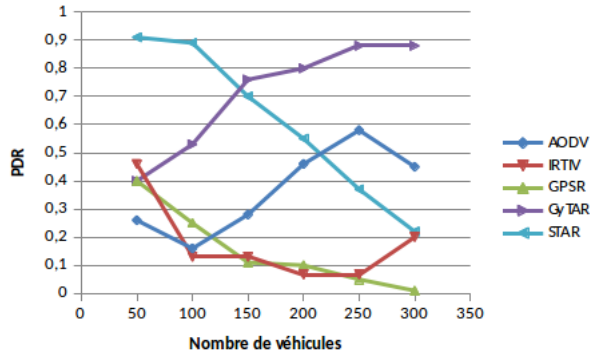


FIGURE 5.3 – Taux de délivrance des paquets en fonction de la densité dans une zone urbaine en modèle Free Space

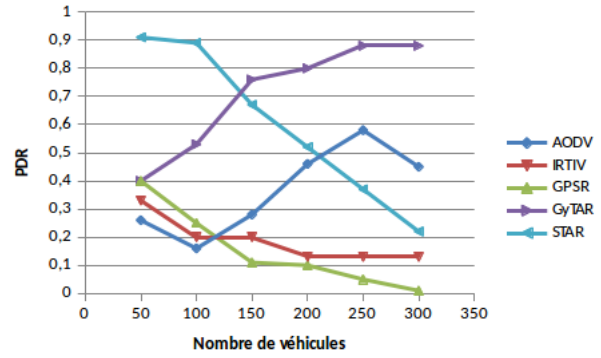


FIGURE 5.4 – Taux de délivrance des paquets en fonction de la densité dans une zone urbaine en modèle Two Ray Ground

5.5 Application 2^k

D'après les résultats obtenus de la simulation des différents protocoles de routage dans différentes densités pour les deux modèles de propagation en considérant ces résultats corrects, on peut extraire les facteurs, ses valeurs extrêmes et les observations nécessaires pour appliquer la méthode 2^k . Le tableau 5.2 présente les facteurs considérés et leurs valeurs extrêmes :

TABLE 5.2 – Les valeurs des facteurs considérés

Indicateur	Facteur	Niveau -1	Niveau +1
F1	Nombre de véhicules	50	300
F2	Modèle de propagation	TwoRayGround	FreeSpace
F3	Protocole de routage	AODV	GPSR
F4	Roadmap	Highway	City

À partir des résultats des simulations, on extrait les observations pour les facteurs

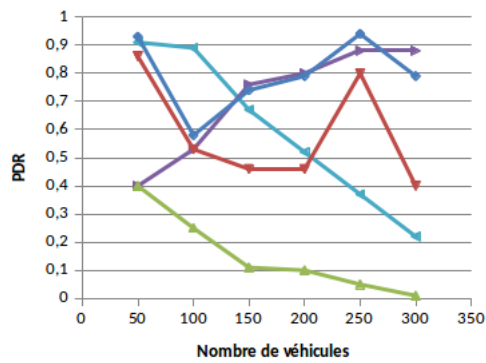


FIGURE 5.5 – Taux de délivrance des paquets en fonction de la densité dans une autoroute en modèle Free Space

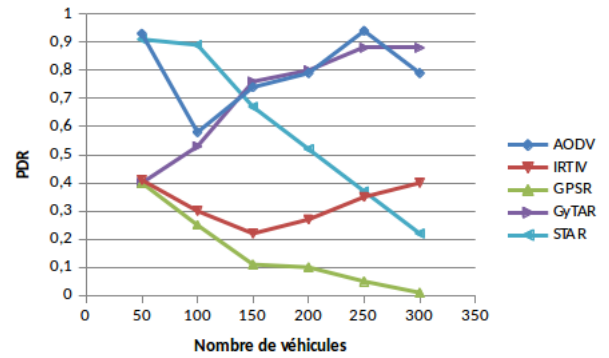


FIGURE 5.6 – Taux de délivrance des paquets en fonction de la densité dans une autoroute en modèle Two Ray Ground

variés précédemment. Ces observations peuvent être rassemblées dans le tableau 5.3 suivant :

TABLE 5.3 – Les observations

Nombre de véhicules	Modèle de propagation	Protocole de routage	Roadmap	Délai (s)	PDR
50	TRG	AODV	Highway	0.12	0.93
50	TRG	AODV	City	0.046	0.26
50	TRG	GPSR	Highway	0.07	0.4
50	TRG	GPSR	City	0.07	0.4
50	FS	AODV	Highway	0.12	0.93
50	FS	AODV	City	0.046	0.26
50	FS	GPSR	Highway	0.07	0.4
50	FS	GPSR	City	0.07	0.4
300	TRG	AODV	Highway	0.26	0.79
300	TRG	AODV	City	0.22	0.45
300	TRG	GPSR	Highway	0.09	0.01
300	TRG	GPSR	City	0.09	0.01
300	FS	AODV	Highway	0.26	0.79
300	FS	AODV	City	0.22	0.45
300	FS	GPSR	Highway	0.09	0.01
300	FS	GPSR	City	0.09	0.01

Notre objectif est de déterminer les facteurs les plus pertinents sur la performance

du routage dans les réseaux VANETs. Pour cela, nous avons implémenté une application Java qui fait extraire les paramètres les plus influents sur la performance d'un tel système. Ainsi, on l'a appliqué sur les valeurs obtenues de la simulation.

Cette application a besoin du nombre de facteurs à considérer, les valeurs extrêmes des facteurs et un fichier texte contenant les valeurs des observations. Elle retourne, par son tour, le pourcentage d'influence de chaque facteur sur le système, celui de plus grande influence et la valeur moyenne de la métrique étudiée (par exp. délai moyen de bout en bout, PDR moyen, ect.)

5.5.1 Délai moyen de bout en bout (End-to-End Delay)

Après avoir appliqué la méthode 2^k sur les données du tableau 5.3, tenant en compte les valeurs des quatre facteurs et la 5^{ième} colonne des valeurs de délai, on a obtenu les résultats présentés dans la figure 5.7, qui est l'interface graphique de notre application. Les résultats montrent que le premier facteur qui est le nombre de véhicules influe le plus sur le routage d'information dans un tel scénario par 37.41%. Ainsi, le protocole de routage par 31.73%, puis la combinaison du nombre de véhicules et le protocole de routage par 22.41%, ensuite le roadmap et la combinaison du protocole de routage et le roadmap par 3.88%, et enfin la combinaison du nombre de véhicules et le roadmap, celle du nombre de véhicules et le protocole de routage et le roadmap par 0.35%. L'application retourne aussi le délai moyen sous l'influence de tous ces facteurs par 0.121 secondes. Donc on conclut que la densité de véhicules influe le plus sur le délai de bout en bout sur un tel réseau.

5.5.2 Taux de délivrance des paquets (Packets Delivery Ratio, PDR)

La figure 5.8 présente les résultats de l'application 2^k avec les mêmes valeurs des facteurs mais les observations de PDR du tableau 5.3. Selon les résultats, la fraction de PDR dans le routage dans le scénario décrit précédemment est affecté essentiellement par le protocole de routage par un pourcentage de 55.16%, puis par le nombre de véhicules et le protocole de routage ensemble par 42.69%, puis par le roadmap et la combinaison du protocole de routage et le roadmap par 16.8%. Ainsi, la combinaison du nombre de véhicules et le protocole de routage par 11.35%, puis le nombre de véhicules par 8.78%, et enfin la combinaison du nombre de véhicules et le roadmap, celle du nombre de véhicules et le protocole de routage et le roadmap par 1.79%. Le PDR moyen est retourné d'une valeur de 0.406. À la fin, le taux de délivrance des paquets est affecté dans un tel réseau essentiellement par le protocole de routage.

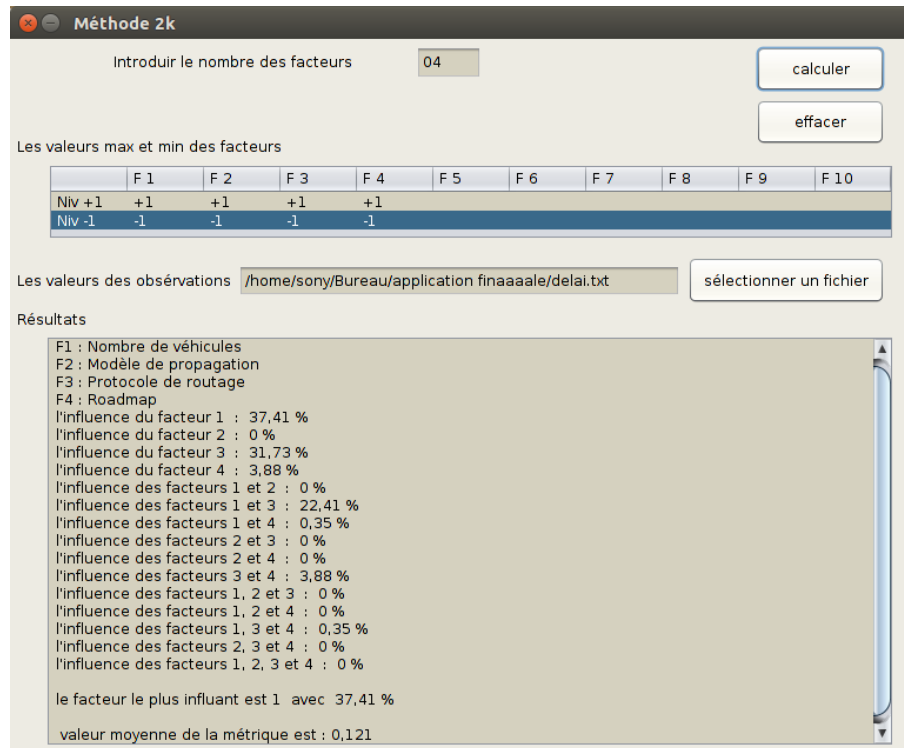


FIGURE 5.7 – Résultats de l'application 2^k sur les observations de délai moyen de bout en bout

5.6 Conclusion

Dans ce chapitre, nous avons comparé les performances des cinq protocoles de routage à savoir AODV, GPSR, IRTIV, GyTAR et STAR, en terme de délai de bout en bout et le taux de délivrance des paquets. À partir des résultats des simulations, on a fait extraire les valeurs extrêmes des facteurs et les observations pour chaque cas. D'après les résultats de l'application, on peut dire que le protocole du routage et la densité du réseau influe le plus sur le routage dans un tel réseau. Ceci, pour concentrer les études sur ce paramètre afin d'améliorer en un temps réduit la performance du système. On termine, dans le chapitre suivant, par une conclusion et perspectives.

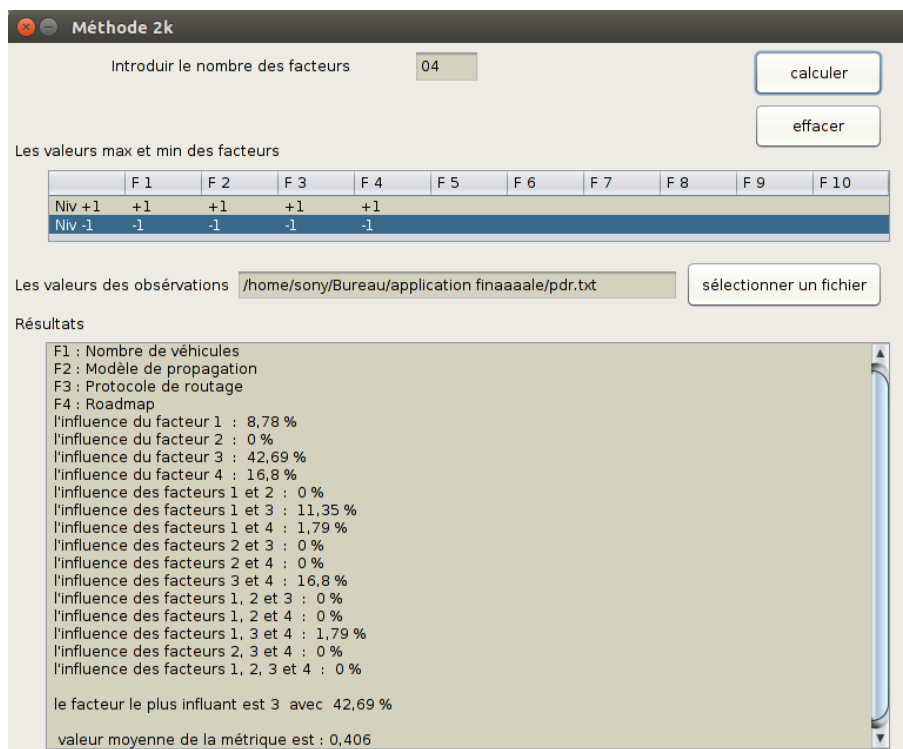


FIGURE 5.8 – Résultats de l'application 2^k sur les observations de PDR

Chapitre 6

Conclusion générale

Le progrès actuel dans le domaine des réseaux VANETs est dû à l'évolution rapide des nouvelles technologies de l'information et de la communication au domaine des transports. Cette évolution vise à rendre les réseaux plus sûrs, plus fiables, et plus efficaces. Les principaux objectifs d'un système de transport intelligent concernent les applications de sécurité routière, les applications de gestion du trafic et les applications de confort de l'utilisateur en lui fournissant une multitude de services d'informations sur le trafic, d'aide à la décision, de guidage, et d'accès à Internet. Les protocoles de routage proposés offrent plusieurs avantages qui sont effectivement meilleurs pour améliorer la performance des différentes applications pour assurer la délivrance des messages aux bonnes destinations.

L'opération de routage dans un système de transport devient donc très importante, celle-ci peut être gouvernée par des paramètres définis dans la littérature des VANETs, qui influent sur cette opération et qui peuvent être variés d'un environnement, scénario et/ou système à un autre, ainsi la conception d'un protocole fiable dans tout cas possible devient difficile. Pour faciliter une telle conception, on a besoin de déterminer ces paramètres et ceux de plus grande pertinence pour concentrer sur lesquels les études et les tests de recherche, afin d'éviter des simulations inutiles et réduire le temps de simulation requis. Une méthode 2^k a été conçue dans le cadre de l'analyse factorielle pour identifier les facteurs les plus influents sur la performance d'un système. Pour cela, nous avons implémenté une application basée sur cette méthode retournant les facteurs les plus influents sur l'opération de routage dans les VANETs.

Dans ce mémoire, nous avons présenté les principaux concepts des réseaux véhiculaires, ainsi les architectures et les applications de ces réseaux, ce qui permet d'avoir une vue générale sur cette nouvelle technologie de réseaux pour pouvoir comprendre son fonctionnement et ses mécanismes de base.

Ensuite, nous avons vu l'analyse factorielle, ses caractéristiques, ses types et ses différentes méthodes pour choisir la plus apte pour l'identification des facteurs. Et puis, nous avons étudié cinq protocoles de routage pour comparer leurs performances, ainsi nous avons présenté la méthode utilisée pour l'identification des facteurs les plus pertinents. Cette méthode permet de déterminer ces facteurs pour minimiser le temps de recherche

et simulation. Après l'obtention des résultats de la simulation et de l'application, nous pouvons dire que le protocole de routage puis nombre de véhicules influent le plus sur le routage dans un tel scénario. Donc, on concentre les études sur ce facteur pour concevoir un protocole de routage plus fiable et efficace.

Ce travail est loin d'être parfait, d'où nous n'avons pu varier que quatre paramètres, mais il peut être considéré comme une petite ouverture vers l'implémentation réelle d'un protocole de routage au sein d'un réseau VANET du fait de l'aide qu'offre cette méthode pour une bonne conception. Comme extensions futures à notre travail nous proposons :

- La variation d'un grand nombre de facteurs pour pouvoir bien identifier ceux de grande influence ;
- l'utilisation d'autres méthodes de l'analyse factorielle comme l'ACP pour résoudre le problème de routage d'informations.

Bibliographie

- [1] Thèse de OLIVIER RIVATON. “Le routage de l’information dans les réseaux véhiculaires mobiles”. Juil. 2016.
- [2] Thèse de MOHAMMED ERRITALI. “Contribution à la sécurisation des réseaux ad hoc véhiculaires”. Oct. 2013.
- [3] Harjit Singh MANJYOT SAINI. “VANET, its Characteristics, Attacks and Routing Techniques : A Survey”. In : *International Journal of Science and Research* (mai 2016).
- [4] Manjusha Deshmukh DIVYALAKSHMI DINESH. “Challenges in Vehicle Ad Hoc Network (VANET)”. In : *IJETMAS* (déc. 2014).
- [5] Md. Humayun KABIR. “Research Issues on Vehicular Ad hoc Network”. In : *International Journal of Engineering Trends and Technology (IJETT)* (déc. 2013).
- [6] “Analyse factorielle”. In : (avr. 2010). URL : http://www.statelem.com/analyse_factorielle.php.
- [7] “analyse en composantes principales”. In : (avr. 2010). URL : http://www.statelem.com/analyse_en_composantes_principales.php.
- [8] “analyse factorielle des correspondances”. In : (avr. 2010). URL : http://www.statelem.com/analyse_factorielle_des_correspondances.php.
- [9] URL : <https://www.xlstat.com/fr/solutions/fonctionnalites/analyse-factorielle-multiple-afm>.
- [10] TANAGRA. “Didacticiel - Études de cas”. In : (2012).
- [11] JÉRÔME PAGÈS SÉBASTIEN LE DIEN. “Analyse factorielle multiple hiérarchique”. In : (2003).
- [12] “Analyse factorielle discriminante”. In : (2010). URL : http://www.statelem.com/analyse_discriminante.php.
- [13] “Régression”. In : (2010). URL : [http://www.statelem.com/regression_\(statistiques\).php](http://www.statelem.com/regression_(statistiques).php).
- [14] PERKINS et E.M. ROYER. “Ad-hoc on-demand distance vector routing”. In : *Mobile Computing Systems and Applications, Second IEEE Workshop* (1999).

-
- [15] URL : https://www.researchgate.net/figure/286734824_fig1_Figure-2-AODV-protocol-algorithm-pseudo-code-The-algorithm-s-composed-of-four-major.
- [16] Talar ATÉCHIAN. “protocole de routage géo-multipoint hybride et mécanisme d’acheminement de données pour les réseaux ad hoc de véhicules (VANETs)”. 2010.
- [17] Thèse de BRAD NELSON KARP. “Geographic Routing for Wireless Networks”. 2000.
- [18] Abderrahmane Lakas OMAR SAMI OUBBATI Nasreddine Lagraa et Mohamed Bachir YAGOUBI. “IRTIV : Intelligent Routing protocol using real time Traffic Information in urban Vehicular environment”. In : *IEEE* (2014).
- [19] Rabah Meraihi MOEZ JERBI Sidi-Mohammed Senouci et Yacine GHAMRI-DOUDANE. “An Improved Vehicular Ad Hoc Routing Protocol for City Environments”. In : *IEEE* (2007).
- [20] Elena Pagani FRANCESCO GIUDICI. “Spatial and Traffic-Aware Routing (STAR) for Vehicular Systems”.
- [21] Manuel FOGUE et al. “Identifying the Key Factors Affecting Warning Message Dissemination in VANET Real Urban Scenarios”. In : *Sensors* (avr. 2013).
- [22] URL : <http://www.isi.edu/nsnam/ns/>.
- [23] FOGUE et al. “A Realistic Simulation Framework for Vehicular Networks”. In : *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques (SIMUTools 2012)* (mar. 2012).
- [24] Thèse de Boudelaâ Imane et NAOUAI NACIRA. “Apport des protocoles de routage geocast au confort des passagers dans les réseaux de véhicules”. Juin 2016.
- [25] Thèse de KERRACHE CHAKER ABDELAZIZ. “Simulation d’un nouvel algorithme de partage de ressources tolérant aux fautes dans les réseaux mobiles ad hoc”. Juin 2012.

Annexe A

annexe

A.1 Le simulateur NS-2

A.1.1 Définition

NS-2 est un simulateur à événements discrets, écrit en C++. C'est le simulateur le plus célèbre dans le domaine de la simulation des réseaux. Il fournit un environnement permettant de réaliser des simulations entre des protocoles IP, TCP, UDP, routage et des protocoles multicast dans les réseaux filaires ainsi que dans les réseaux sans fil avec un support de mobilité des nœuds.

A.1.2 L'architecture de simulation

NS-2 est un logiciel de simulation de réseaux orienté objet par événements discrets, il est très largement utilisé dans la recherche académique et dans l'industrie. NS-2 se base sur deux langages de programmation, C++ qui est un langage compilé très efficace pour l'implémentation des protocoles, et OTCL (Object Tools Command Language) qui représente l'extension orienté objet du langage interprété TCL (Tool Command Language). Ce dernier a deux fonctions fondamentales : il se comporte comme un langage de programmation autonome interprété et comme une interface permettant de contrôler la simulation des protocoles via les commandes du script. La simulation est contrôlée par un script OTCL écrit par l'utilisateur. Pour cela, il faut qu'il y ait un lien entre l'interpréteur OTCL et la bibliothèque du simulateur NS-2 qui contient les protocoles avec leurs variables, (cf. A.2). NS-2 ne permet pas de visualiser le résultat des expérimentations. Il permet uniquement de stocker une trace de la simulation, de sorte qu'elle puisse être exploitée par un autre logiciel, comme le NAM. Ce dernier est un outil de visualisation qui présente deux intérêts principaux : représenter la topologie d'un réseau décrit avec NS-2, et afficher temporellement les résultats d'une trace d'exécution NS-2[24].

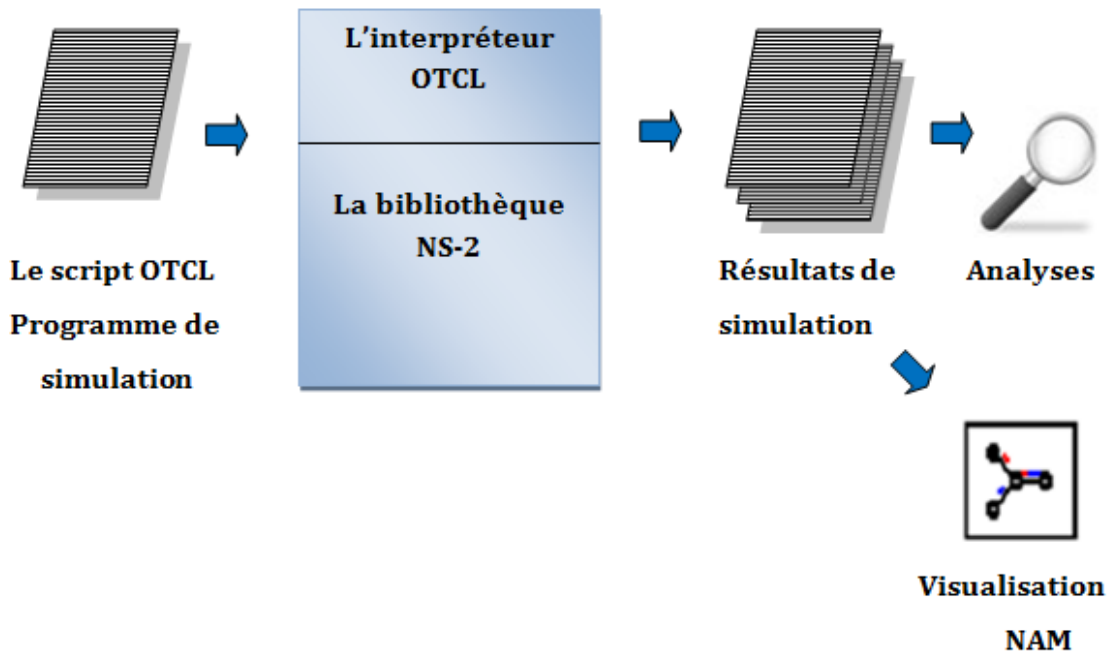


FIGURE A.1 – L'architecture de simulation avec NS-2

A.1.3 les étapes de simulation

La simulation avec NS-2 passe en général par trois phases :

1. Définition de la topologie du réseau : on crée les nœuds, avec les caractéristiques de chacun, ainsi que les liens entre les nœuds. On peut définir sur chaque lien, le délai de transmission, la bande passante, le fait qu'il soit simplexe ou duplexe et le type de file d'attente se trouvant à son extrémité.
2. Spécification du scénario de la simulation : l'utilisateur spécifie les différents agents de la communication qui vont agir pendant la simulation. Il spécifie aussi la succession des différentes opérations (à l'instant t_1 , envoi des données, à l'instant t_2 arrêt d'émission).
3. Exploitation des résultats : cette dernière phase consiste en un recueil des statistiques de la simulation. Ces dernières peuvent être exploitées directement par NS-2 en faisant appel aux outils qui l'accompagnent, ou bien elles seront archivées pour une utilisation ultérieure par d'autres outils de traitements statistiques.

A.1.4 Réalisation d'une simulation

La simulation nécessite des données qui caractérisent l'environnement, tels que la surface du réseau, la topologie utilisée, le protocole à simuler, etc. Généralement, ces données ne sont pas définies en NS. Pour cela, l'utilisateur doit définir les informations

(données) en utilisant le langage C++. Afin de réaliser un algorithme, nous devons créer deux fichiers source écrits en langage C++ (*.h, *.cc). Le premier (*.h) est un fichier d'en-tête, qui contient la structure des messages échangés entre les sites, le deuxième (*.cc) contient les fonctions nécessaires de l'algorithme (envoi et réception des messages, etc).

La compilation de ces fichiers nous permet d'avoir un fichier de type objet (*.o), ce dernier doit être intégré dans le simulateur NS-2, en lui ajoutant à la variable OBJ_CC du fichier (makefile), enfin, on recompile le noyau de NS par la commande make écrite dans un terminal.

On peut résumer les étapes de réalisation d'un algorithme dans la figure

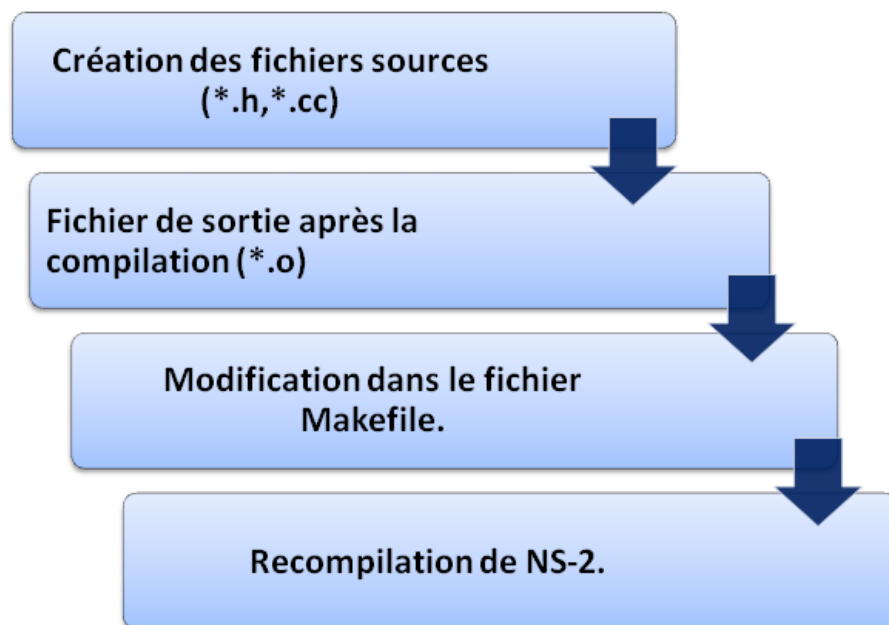


FIGURE A.2 – Les étapes de simulation

Pour tester et utiliser ces programmes, nous avons créé les fichiers (*.tcl) qui contiennent la topologie du réseau, ensuite on appelle les fonctions du code C++, et enfin la visualisation de simulation.

La sortie standard est un fichier d'animation (*.nam) pour visualiser la simulation, et un fichier de trace (*.tr) qui contient toutes les informations obtenues après la simulation, ces informations sont très générales et nécessitent des filtres pour bien les exploiter. C'est la raison pour laquelle, nous avons utilisé un fichier texte (*.txt) qui ne contient que les informations importantes pour notre simulation[25].