

Etude des protocoles de service de localisation dans les réseaux véhiculaires

DHELIM Sahraoui

2013/2014

Table des matières

Introduction général	vi
1 Introduction aux réseaux véhiculaires	1
1.1 Introduction	1
1.2 Les réseaux sans fil	1
1.3 Architecture de réseau sans fil	1
1.4 Les réseaux MANET	2
1.5 Les réseaux véhiculaires	3
1.6 Les types de communications des réseaux véhiculaires	4
1.6.1 Communications Véhicule à Véhicule (V2V)	5
1.6.2 Communications Véhicule à Infrastructure (V2I)	5
1.6.3 Communication Hybride	5
1.7 Caractéristiques des réseaux VANET	7
1.8 Applications des réseaux véhiculaires	8
1.8.1 Application dans la prévention et la sécurité routière	8
1.8.2 Application pour l'optimisation du trafic et aide dans la conduite	8
1.9 Défis et thèmes de recherche dans le domaine de VANET	8
1.10 Conclusion	9
2 Service de localisation	10
2.1 Introduction	10
2.2 Service de localisation	10
2.3 Composantes du service de localisation	11
2.4 Taxonomie des services de localisation	11
2.4.1 Services de localisation basés sur les inondations	11
2.4.2 Services de localisation basés sur le quorum	12
2.5 Les protocoles de service de localisation	13
2.5.1 Les protocoles de service de localisation pour les réseaux MANETs	13
DREAM Location Service (DLS)	13
Simple Location Service (SLS)	14
Reactive Location Service (RLS)	14
Grid Location Service (GLS)	15
Hierarchical Location Service (HLS)	18
2.5.2 Les protocoles de service de localisation pour les réseaux VANETs	26
Intersection Location Service (ILS)	26
DMBLS	26
2.6 Conclusion	30

3	Étude comparative et proposition	31
3.1	Introduction	31
3.2	Etude comparative	31
3.3	Notre contribution	33
3.3.1	Stratégie de mise à jour d'origine	33
3.3.2	La stratégie de mise à jour proposé	33
3.4	Les solutions proposées	33
3.5	Conclusion	39
4	Les résultats de simulation	40
4.1	Introduction	40
4.2	Environnement de simulation	40
4.3	Les protocoles simulés	40
4.4	Les paramètres de simulation	41
4.4.1	Taux de réussite	42
4.4.2	Précision de la position	42
4.4.3	Temps de séjour	42
4.4.4	Surcharge des mises à jour	43
4.5	Les résultats de simulation	43
4.6	Analyse de résultats	47
4.7	Conclusion	49
	Conclusion générale	50

Table des figures

1.1	Exemple d'un BSS.	2
1.2	Exemple d'un IBSS.	3
1.3	Exemple d'un réseau ad hoc.	4
1.4	Exemple d'un RSU(Roadside unit)[2].	4
1.5	Hiérarchie des réseaux sans fil [3].	5
1.6	Communications Véhicule à Infrastructure [4].	6
1.7	Communications Véhicule à Véhicule [4].	6
2.1	La taxonomie des services de localisation [8].	12
2.2	Exemple d'acheminement gourmand, le noeud A est la source, et G la destination.	16
2.3	Des exemples de carrés valides de tailles différentes (les noirs). Le gris n'est pas un carré ordre 2.	18
2.4	Les carrés encastrés sont des régions dans lesquelles B cherchera un serveur de localisation. Les nœuds qui deviennent les serveurs de localisation de B sont encadrés et apparaissent en gras. [12] .	19
2.5	L'organisation de serveur de localisation d'un réseau entier. Chaque noeud est affiché avec la liste de nœuds pour lesquels il a des informations de localisation; les serveurs de localisation de B sont indiqués en gras. Deux requêtes possibles par A pour l'emplacement de B sont présentées. [12]	20
2.6	L'échec d'acheminement gourmand. Le nœud C ne trouve pas un autre nœud qui est plus proche de G que lui-même, même si il ya un chemin d'accès valide (C ->D->E->F-> G) à G	21
2.7	Les niveaux d'hiérarchie.[13]	21
2.8	Exemple des RCs d'un nœud.[13]	22
2.9	L'arbre de candidat. [13]	23
2.10	La mise à jour. [13]	24
2.11	Exemple d'une requête ([13])	25
2.12	Partitionnement géographique de la zone.[16]	27
2.13	La sélection des serveurs de localisation à l'intérieur de la cellule correspondante à l'intersection Wi.[16]	28
2.14	Les requête et les mises à jour en DMBLS.[16]	29
3.1	la stratégie de mise a jour de DMBLS	34
3.2	les serveurs de localisation ne fourniront que la région de niveau inférieur	35
3.3	La stratégie de mise a jour proposé	36
3.4	Solution pour les intersections vides	37

3.5	La requête détecte une intersection vides alors il va visiter les voisinages pour chercher un serveur temporaire	38
4.1	La zone simulée (ville de Laghouat)	41
4.2	TR en fonction de nombre de nœuds	45
4.3	TR en fonction de la vitesse.	45
4.4	EP en fonction de nombre de nœuds	46
4.5	EP en fonction de la vitesse.	46
4.6	TS en fonction de nombre de nœuds	47
4.7	TS en fonction de la vitesse.	47
4.8	Nombre de MAJ en fonction de nombre de nœuds	48
4.9	Nombre de MAJ en fonction de la vitesse.	48

Liste des tableaux

3.1	Comparaison entre les service de localisation.	32
4.1	Les paramètres de simulation.	42
4.2	Les résultats de simulation de DMBLS.	43
4.3	Les résultats de simulation de DMBLSX	43
4.4	Les résultats de simulation de ILS.	44
4.5	Les résultats de simulation de CLS.	44

Introduction général

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

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

Les réseaux Ad hoc peuvent être utilisés à des endroits où l'installation d'infrastructures est trop coûteuse ou impossible. L'une de leurs caractéristiques clés est leur topologie très dynamique, du fait de la mobilité des unités du réseau. Ils ont une large gamme d'applications, dont les réseaux véhiculaire ad hoc (VANETs, pour Vehicular Ad hoc NETWORK). Ce type de réseaux fait communiquer des véhicules entre eux via des interfaces sans fil. La conduite nécessite un besoin constant d'informations sur l'environnement extérieur du conducteur. Ces informations peuvent renforcer la prévention routière (informations sur le trafic, état de la chaussée, météo...), simplifier les voyages (emplacement des stations de ravitaillement en carburant, emplacement des bouchons, prix du carburant...) et améliorer le confort des passager (accès à internet, jeux interactifs entre les passagers de voitures proches, service de chat...). Ces informations peuvent être partagées grâce aux VANETs, afin de créer un système collaboratif de prévention et de communication. Le routage basé sur la position a prouvé d'être un moyen efficace et extensible pour le routage des paquets dans les réseaux ad hoc mobiles, pour activer le routage basé sur la position, un nœud doit être capable de découvrir l'emplacement du nœud qu'il veut contacter, cette tâche est typiquement accomplie par un service de

localisation.

Un service de localisation est un algorithme distribué qui permet à n'importe quel nœud source S de connaître l'emplacement d'un nœud de destination T ; simplement en connaissant l'identifiant de réseau de T . Un nœud doit être capable de découvrir l'emplacement du nœud qu'il veut solliciter, cela est le but d'un service de localisation. Afin d'étudier les VANETs, le déploiement sur terrain réel n'est, malheureusement pas envisageable, d'où le recours à la simulation. Plusieurs simulateurs ont été mis à la disposition des chercheurs dans ce but (GloMoSim, ns-2, Omnet++...). Les objectifs principaux de ce travail sont :

1. Etudier et comparer les principaux services de localisation.
2. Proposer une amélioration d'un de ces protocoles, puis prouver cette amélioration par une simulation.

Le reste de cette mémoire est organisé comme suit :

Dans le premier chapitre, nous introduisons au premier lieu les concepts des réseaux sans-fil et ses types. Puis, nous allons détailler les principaux concepts liés aux réseaux de véhicules tels que les types de communication, les caractéristiques ainsi que les applications liées à ce type de réseau.

Dans le deuxième chapitre nous allons discuter le concept de service de localisation et présenter une taxonomie de ces services, puis détailler certains protocoles

Dans le troisième chapitre nous allons comparer certains services de localisation et essayer de proposer une solution pour améliorer performance de DM-BLS.

Dans le quatrième chapitre, nous allons simuler certains protocoles de localisation dans une partie de la carte de la ville de Laghouat. Les résultats obtenus concernant la fiabilité du service de localisation sont ensuite analysés et comparés.

La conclusion de ce mémoire résume les travaux faits ainsi que des possibles améliorations futures.

Chapitre 1

Introduction aux réseaux véhiculaires

1.1 Introduction

La maturité des transmissions sans fil et la popularité des appareils informatiques portables ont fait le rêve de «communication à tout moment et n'importe où » possible. Un réseau sans fil ad hoc est un bon choix pour l'accomplissement de ce rêve.

Dans ce chapitre, nous présenterons les réseaux ad hoc mobile, leurs catégories, leurs caractéristiques ainsi que leurs contraintes. Pour enfin parler des réseaux VANET tels que les types de communications, les domaines d'application et les défis et thèmes de recherche dans le domaine des réseaux véhiculaires.

1.2 Les réseaux sans fil

Les réseaux sans fil sont des réseaux où les nœuds sont connectés et communiqué les uns avec les autres pas par un moyen visible, mais par les émissions des ondes électromagnétique dans l'air [1].

1.3 Architecture de réseau sans fil

Un réseau sans fil peut être structuré de manière à fonctionner dans les deux modes, en mode BSS (Basic Service Set) ou en mode IBSS (Independent Basic Service Set). Les deux modes affectent la topologie et les capacités de mobilité des machines (nœuds) qui composent le réseau.

Le mode BSS En mode BSS, également appelé mode infrastructure, un certain nombre de nœuds mobiles sont connectés sans fil à un point d'accès (Ou AP 'Access point') non-mobile, comme dans la Figure 1.1. Les nœuds communiquent par l'intermédiaire de l'AP, qui peut également fournir une

connectivité avec un réseau externe, par exemple l'Internet, plusieurs réseaux BSS peuvent être joints pour former un ESS (Extended Service Set).

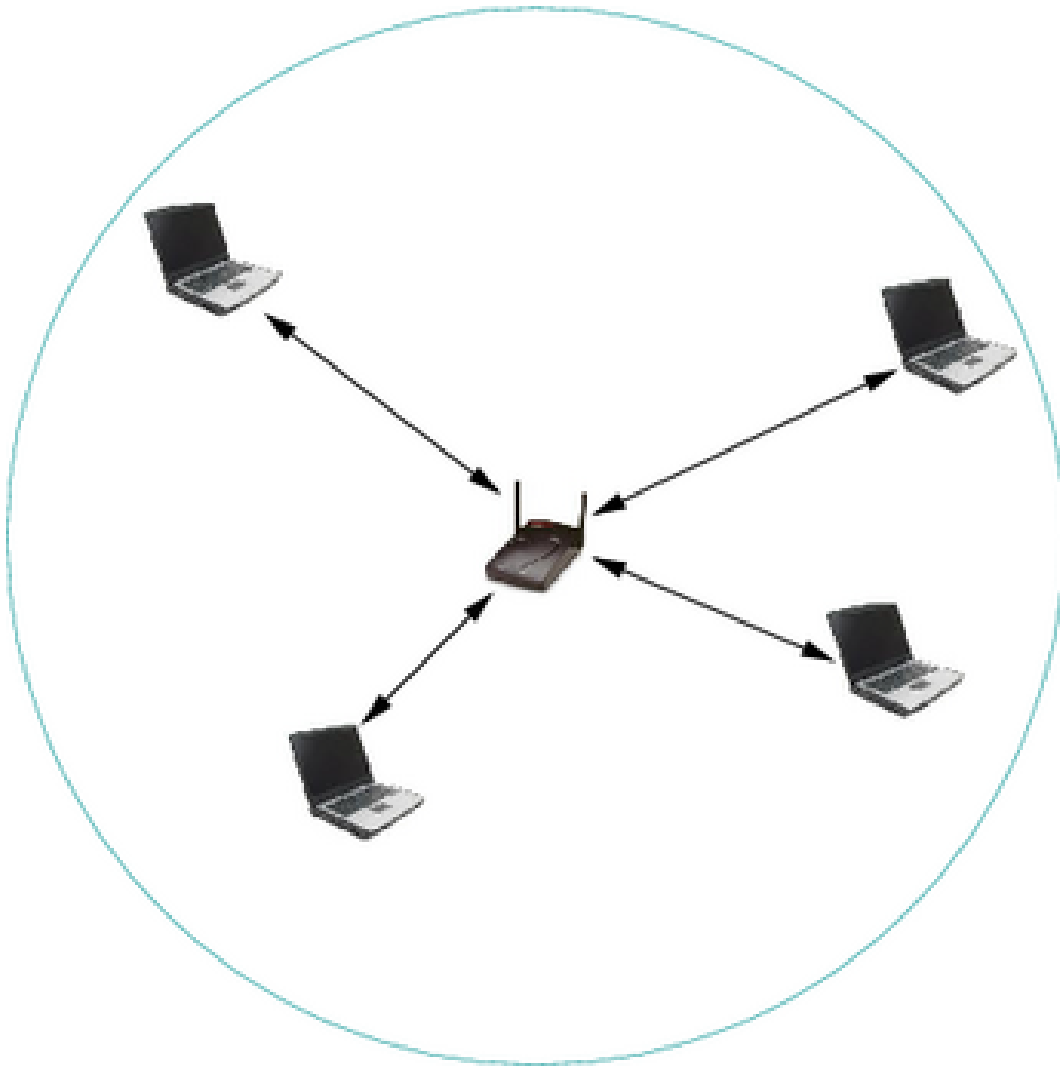


FIGURE 1.1 – Exemple d'un BSS.

Le mode IBSS (Ad-hoc) Le mode IBSS, également appelé peer to peer ou mode ad hoc, permet aux nœuds de communiquer directement (point à point) sans avoir besoin d'un point d'accès. Comme dans la Figure 1.2, les nœuds doivent être à la portée les uns avec les autres afin de communiquer.

1.4 Les réseaux MANET

Les réseaux mobiles ad hoc MANET (Mobile Adhoc NETWORKS) sont constitués d'un ensemble d'ordinateurs mobiles fonctionnant sans l'aide d'une infrastructure bien établie et sans administration centralisée. La communication se fait par des liaisons sans fil entre les ordinateurs mobiles grâce à leurs antennes. En raison de préoccupations telles que la limitation de la puissance de la radio et de l'utilisation des canaux, un hôte mobile peut ne pas être en mesure de communiquer directement avec d'autres hôtes dans un mode mono-saut. Dans ce cas, un scénario multi-sauts se produit, dans lequel les paquets envoyés par

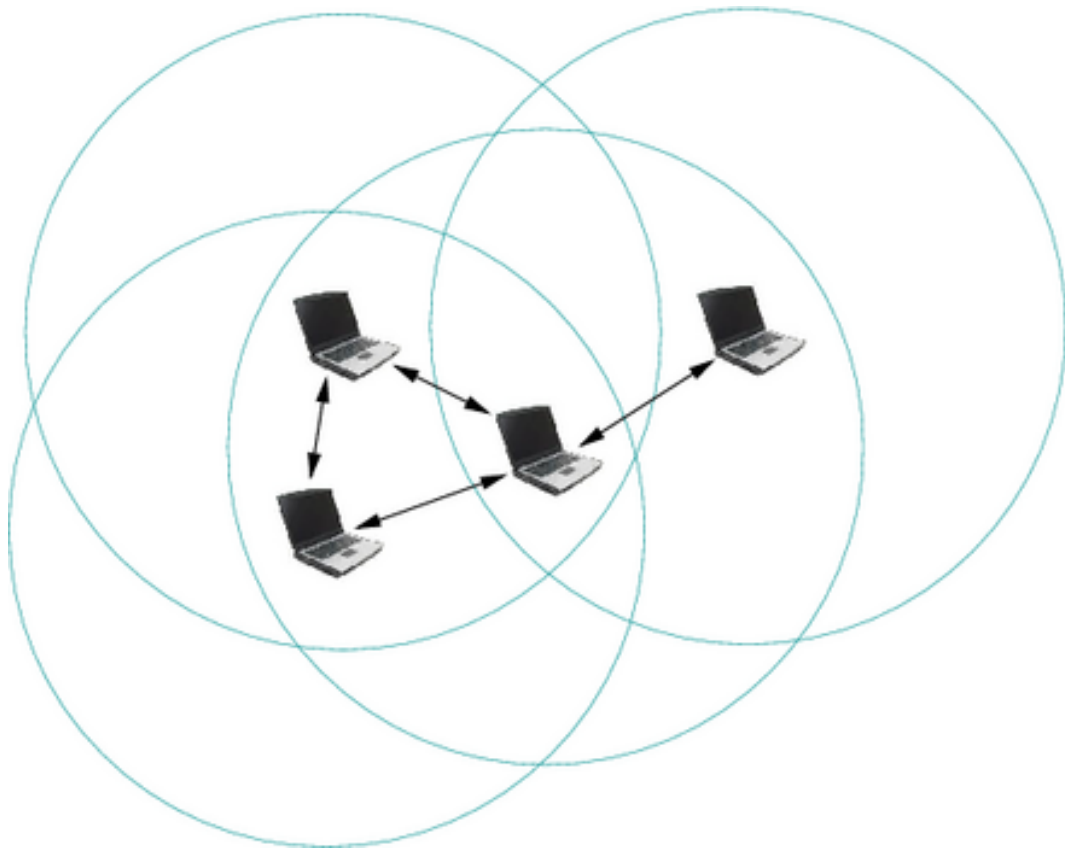


FIGURE 1.2 – Exemple d'un IBSS.

l'hôte source doivent être relayés par plusieurs hôtes intermédiaires avant d'atteindre l'hôte de destination. En fait, un réseau ad hoc a la capacité de rendre les communications possibles même entre deux nœuds qui ne sont pas à portée directe les uns avec l'autre : les paquets qui doivent être échangés entre ces deux nœuds sont transmis par des nœuds intermédiaires, en utilisant un algorithme de routage.

Par exemple, dans le réseau ad hoc comme le montre la Figure 1.3, le nœud A peut communiquer avec le nœud D via les nœuds B et C, et vice versa. Dans ces réseaux les appareils mobiles agissant à la fois comme routeur et terminal en même temps.

1.5 Les réseaux véhiculaires

Les réseaux véhiculaires ou bien les réseaux véhiculaires (VANET) sont une nouvelle classe de réseaux ad hoc mobile (MANET) voir Figure 1.5, qui sont apparus grâce aux progrès des technologies sans fil et l'industrie automobile, les réseaux de véhicules se forment spontanément entre les véhicules en mouvement équipés d'interfaces sans fil. Ces réseaux sont considérés comme l'application, la plus récentes des réseaux ad hoc mobile, permettant les communications entre les véhicules à proximité ainsi que entre les véhicules et l'équipement fixe à proximité RSU (RoadSide Unit) voir Figure 1.4.

Les nœuds mobiles sont des véhicules (intelligents) équipés de calculateurs, de cartes réseau et des capteurs.

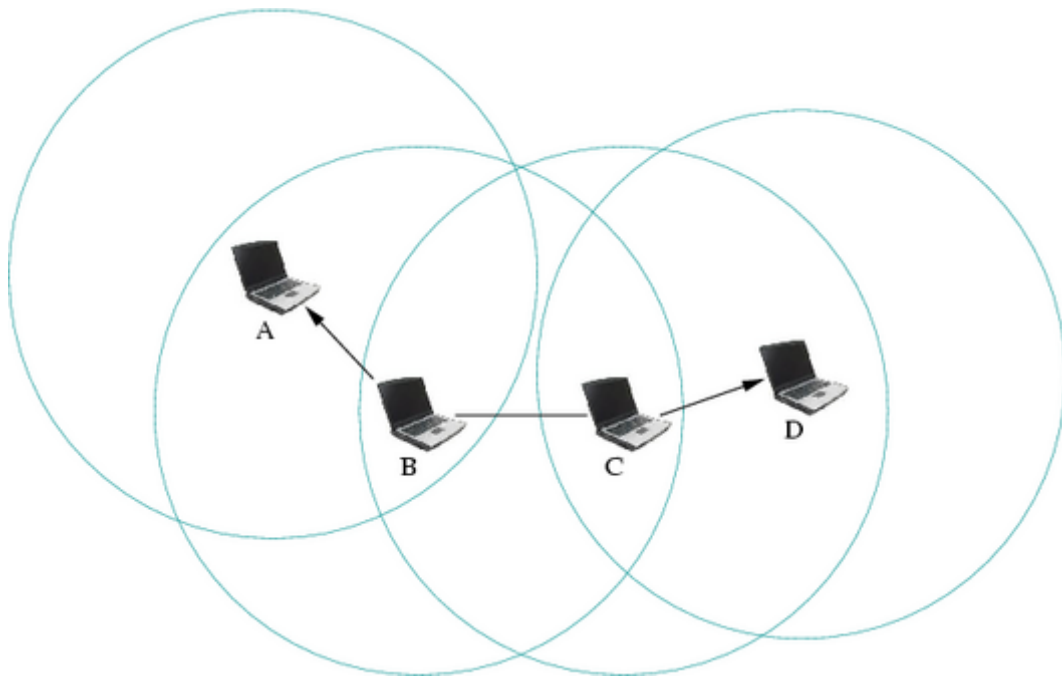


FIGURE 1.3 – Exemple d'un réseau ad hoc.



FIGURE 1.4 – Exemple d'un RSU(Roadside unit)[2].

1.6 Les types de communications des réseaux véhiculaires

Il y a trois types de communication dans les réseaux véhiculaires

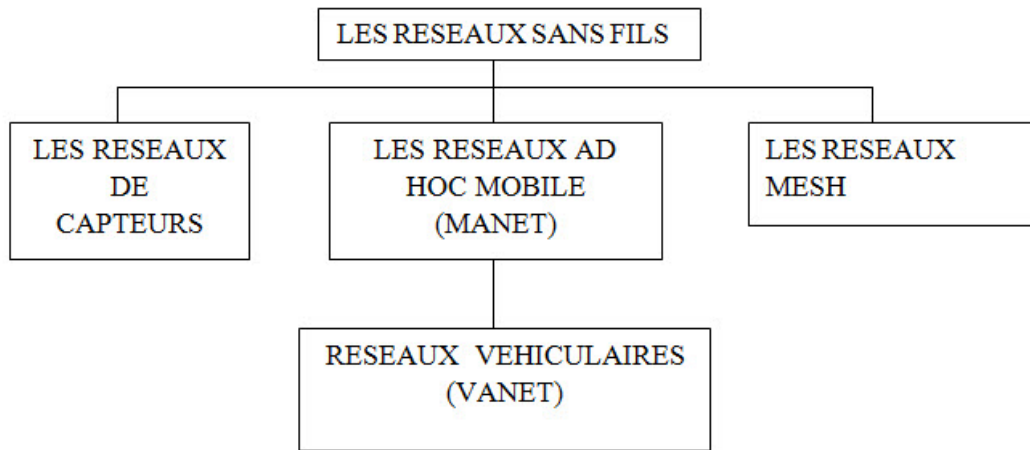


FIGURE 1.5 – Hiérarchie des réseaux sans fil [3].

1.6.1 Communications Véhicule à Véhicule (V2V)

Dans ce type de communication, tous les véhicules sont équipés pour communiquer directement entre eux n'importe où, que se soit sur les autoroutes, des routes de montagnes ou des routes urbaines (voir Figure 1.6), ce qui donne une communication moins coûteuse et plus flexible, parce qu'aucune infrastructure n'est utilisée, aucune installation sur les routes n'est nécessaire. Cette architecture peut être utilisée dans le scénario de diffusion d'alertes (freinage d'urgence, collision, ralentissement...) ou pour la conduite coopérative. Donc dans ce cas, un réseau de véhicules est vu comme un cas particulier du réseau MANET ou les contraintes d'énergie, de mémoire et de capacité sont relaxées. Cette approche souffre de certains inconvénients :

- La sécurité du réseau est très limitée.
- Les déconnexions sont fréquentes, à cause de la haute mobilité.
- Les délais de communication qui sont élevés, étant donné que la communication se fait en utilisant le routage multi sauts.

1.6.2 Communications Véhicule à Infrastructure (V2I)

Cette approche est basée 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 appelés aussi station de base. Les stations sont connectées entre eux via une interface filaire ou sans-fil. Toute communication avec les autres réseaux (ex : internet) doit passer par eux. Ils peuvent aussi offrir aux utilisateurs plusieurs services. 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.

1.6.3 Communication Hybride

La communication hybride est très intéressante, cette communication est la combinaison des communications V2V avec les communications de V2I, en effet que les portées des infrastructures (stations de bases) étant limitées, l'uti-

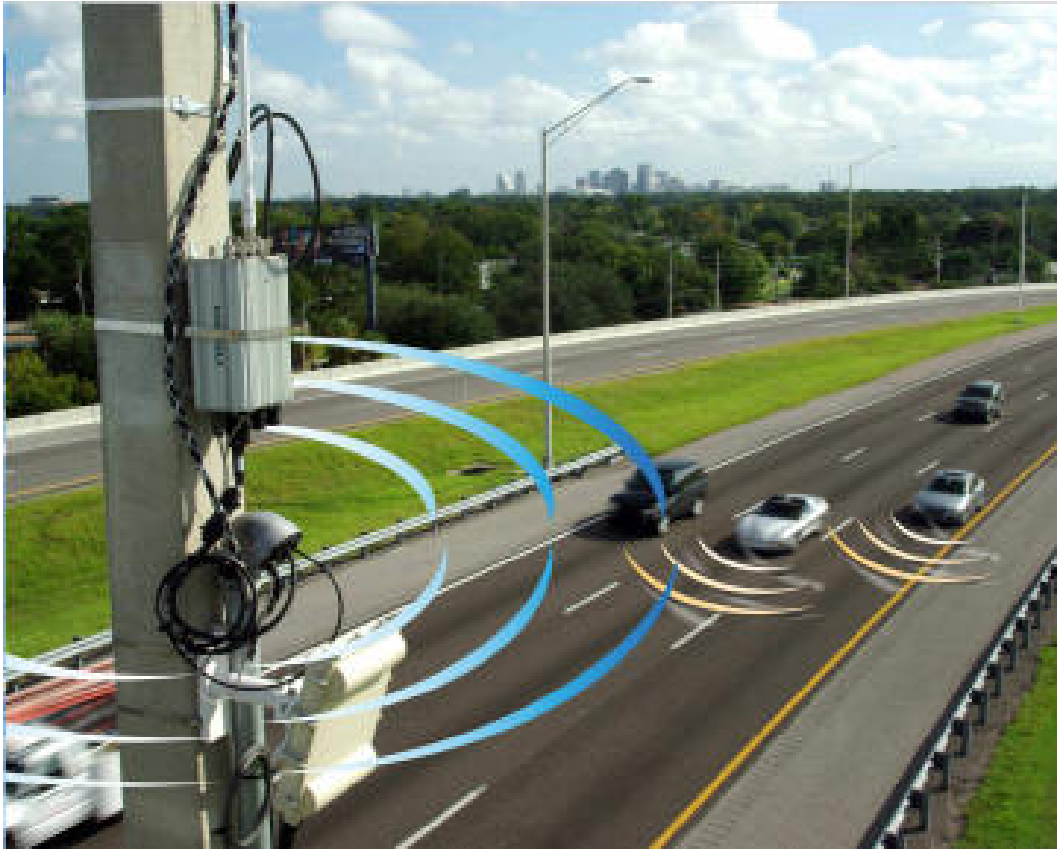


FIGURE 1.6 – Communications Véhicule à Infrastructure [4].

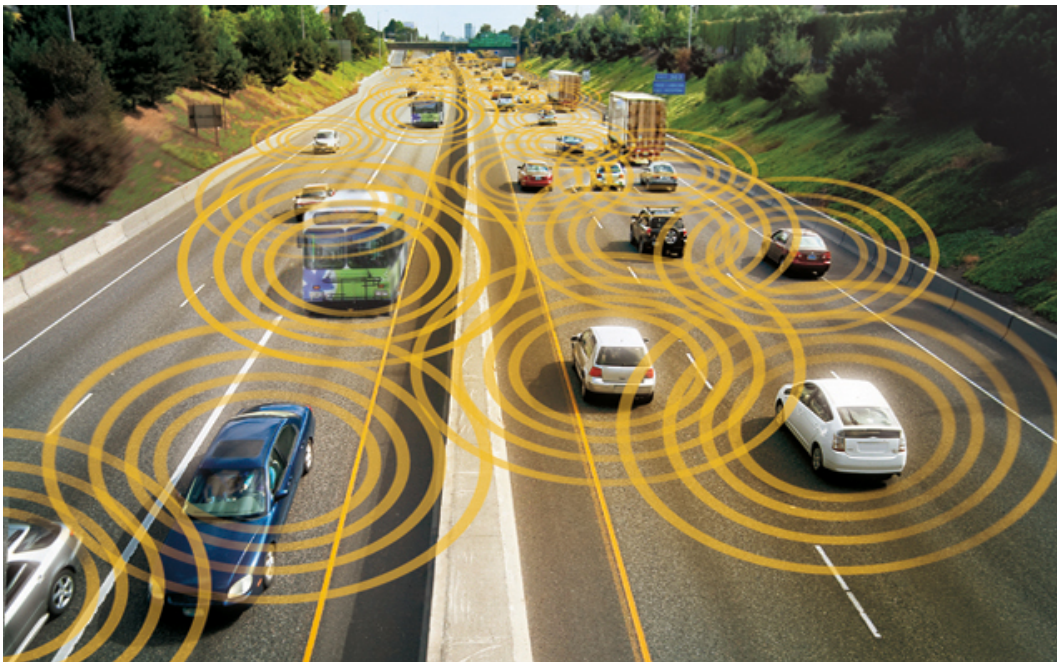


FIGURE 1.7 – Communications Véhicule à Véhicule [4].

lisation 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.

1.7 Caractéristiques des réseaux VANET

Les réseaux de véhicules ont un comportement et des caractéristiques particulières, qui les distinguent des autres types de réseaux mobiles. En comparaison à d'autres réseaux de communication, les réseaux de véhicules viennent avec des caractéristiques intéressantes uniques, comme suit :

Source d'énergie illimité Le problème d'énergie n'est généralement pas une contrainte importante dans les réseaux de véhicules comme dans le cas des réseaux ad hoc classiques ou les réseaux de capteurs, puisque le nœud (véhicule) lui-même peut fournir une énergie continue pour les calculs et les communications.

Grande capacité de calcul Les véhicules peuvent se permettre un calcul intensif, et une longue communication, et des capacités de détection importante.

Mobilité prévisible Contrairement aux réseaux ad hoc mobiles classiques, où il est difficile de prévoir la mobilité des nœuds, les véhicules ont tendance à avoir des mouvements très prévisibles qui sont (généralement) limités sur les routes, les Informations sur la route est souvent disponible par l'utilisation des systèmes de positionnement tel que le GPS, compte tenu de la vitesse moyenne, la vitesse actuelle, la trajectoire de la route, et la position future d'un véhicule peut être prédite.

Taille du réseau Contrairement à la plupart des réseaux ad hoc étudiés dans la littérature qui assument généralement une taille limitée du réseau, les réseaux de véhicules peuvent, en principe, s'étendre sur la totalité du réseau routier et donc inclure de nombreux participants.

Grande mobilité L'environnement dans lequel opèrent les réseaux de véhicules est extrêmement dynamique, et comprend des configurations extrêmes : sur les routes, les vitesses relatives de jusqu'à 300 kilomètres par heure peuvent se produire, alors que la densité de nœuds peut être 1-2 véhicules par 1 km sur routes moins occupés. D'autre part, dans la ville, des vitesses relatives peuvent atteindre jusqu'à 60 km / h et la densité de nœuds peut être très élevée.

Réseau partitionné Les réseaux véhiculaires peuvent être partitionnés fréquemment. La nature dynamique du trafic peut entraîner des grandes lacunes entre les véhicules dans les scénarios de faible densité de population et par conséquent dans plusieurs isolés amas de nœuds.

1.8 Applications des réseaux véhiculaires

Les principales applications des réseaux VANET peuvent être classées en trois catégories :

1.8.1 Application dans la prévention et la sécurité routière

La sécurité routière est devenue une priorité dans la plupart des pays développés, cette priorité est motivée par le nombre croissant d'accidents sur ses routes associé à un parc de véhicules de plus en plus important. Les applications de la sécurité visent à minimiser les accidents et d'améliorer les conditions de circulation en fournissant des informations utiles, y compris les avertissements de collision, alarmes de signe de route, A titre d'exemple, alerter un conducteur en cas d'accidents permet d'avertir les véhicules qui se dirigent vers le lieu de l'accident que les conditions de circulations se trouvent modifiées et qu'il est nécessaire de redoubler de vigilance. Les messages d'alertes et de sécurité doivent être de taille réduite pour être transmis le plus rapidement possible et doivent être émis à des périodes régulières.

1.8.2 Application pour l'optimisation du trafic et aide dans la conduite

Le trafic automobile peut être grandement amélioré grâce à la collecte et au partage de données collectées par les véhicules, ce qui devient un support technique pour les conducteurs. Une voiture peut, par exemple, être avertie en cas d'un ralentissement anormal (bouchon, embouteillage, éboulement de rochers ou travaux).

Applications au confort du conducteur et des passagers Les réseaux véhiculaires peuvent aussi améliorer le confort des conducteurs et des passagers. Ce confort est illustré par l'accès à internet, la messagerie, le chat inter-véhicule, etc. Les passagers dans la voiture peuvent jouer en réseaux, télécharger des fichiers MP3, envoyer des cartes à des amis, etc.

1.9 Défis et thèmes de recherche dans le domaine de VANET

Dans les réseaux VANET, ils existent plusieurs domaines de recherche :

Sécurité Sécurité des véhicules de communication est un défi majeur, avoir un grand impact sur le déploiement et l'application de réseaux véhiculaires. En effet, la sécurité et la confidentialité sont des préoccupations majeures dans le développement et l'acceptation des services et ne doivent pas être compromises

par la facilité d'utilisation des protocoles de découverte de services. Comme la demande de découverte du service est de plus en plus, les passagers peuvent utiliser les services des réseaux étrangers et créer des problèmes de sécurité immense pour eux-mêmes et pour les autres utilisateurs du réseau.

Accès au canal Les réseaux véhiculaires utilisent des communications radio. Par conséquent, il est important de concevoir des solutions spécifiques aux réseaux VANET qui permettent d'apporter de la qualité de service et de gérer les priorités en résolvant les problèmes d'interférences radio, des problèmes de propagation à multi-trajets des ondes.

Routage Le routage dans les réseaux VANET est un problème très difficile à gérer et un axe de recherche pour beaucoup de chercheurs. Les réseaux de véhicules se diffèrent des réseaux sans fil ad hoc classiques non seulement des changements rapides dans les connexions de liaison sans fil, mais aussi d'avoir à traiter avec différents types de densités de réseau. Pour que les véhicules puissent communiquer entre eux, un protocole de routage doit être utilisé. En effet, quand les terminaux ne sont pas à une portée de transmission radio directe, le routage est exigé pour établir la communication entre les véhicules.

Configuration IP et Gestion de la mobilité L'architecture potentielle de V2I est prometteuse en permettant l'accès à Internet aux véhicules ainsi que la fourniture de services liés à l'Internet pour les conducteurs et les passagers. Cependant, deux défis techniques existent sous cette question : configuration de l'adresse IP et la gestion de la mobilité. Ces défis peuvent menacer la qualité de service et la continuité du service. En ce qui concerne les caractéristiques du réseau de véhicules, la configuration d'adresse IP doit être effectuée d'une manière automatique et distribuée.

Localisation des véhicules Si l'un des véhicules du réseau doit être localisé (dans le cas d'un accident par exemple), les autres doivent être informés de sa position. Le problème est que tous les véhicules ne sont pas équipés d'un système de positionnement par satellite (GPS). Pour cette raison, un mécanisme de localisation sans utilisation de GPS est nécessaire.

1.10 Conclusion

Dans ce chapitre nous avons présenté les réseaux sans fil, les réseaux mobiles Ad Hoc ainsi que les réseaux véhiculaires (VANET) qui ne sont qu'une particularité des réseaux MANET.

Dans le chapitre suivant, nous présenterons le concept de service de localisation (location service), dans laquelle nous discuterons certains points comme, ce qu'est un service de localisation, comment on peut améliorer les performances des protocoles de routage en utilisant un service de localisation, et nous présenterons quelques protocoles qui l'utilisent.

Chapitre 2

Service de localisation

2.1 Introduction

Le routage basé sur la position a prouvé d'être un moyen efficace et extensible pour le routage des paquets dans les réseaux ad hoc mobiles. Pour activer le routage basé sur la position, un nœud doit être capable de découvrir l'emplacement du nœud qu'il veut solliciter. Cette tâche est typiquement accomplie par un service de localisation.

Dans ce chapitre, nous allons voir comment un service de localisation peut accomplir cette tâche ; en présentant certains protocoles de service de localisation, puis, nous allons présenter une taxonomie des services de localisation existants.

2.2 Service de localisation

Un service de localisation (Location service) pour les réseaux ad hoc est un algorithme distribué qui permet à n'importe quel nœud source S de connaître l'emplacement d'un nœud destinataire T , en connaissant simplement l'identifiant de T . En utilisant un service de localisation, un nœud doit être capable de connaître la position du nœud qu'il veut solliciter. Afin de savoir la position actuelle d'un nœud particulier, l'aide d'un service de localisation est nécessaire, les nœuds mobiles peuvent enregistrer leur position actuelle avec un service de localisation, quand un nœud source ne connaît pas la position d'un partenaire de communication désiré, il utilise le service et demande la position de nœud cible.

Dans les réseaux cellulaires classiques, il ya des serveurs dédiés de la position (avec des adresses bien connues) qui maintiennent les informations de position sur des nœuds du réseau. Dans les réseaux ad hoc mobiles, une telle approche centralisée n'est viable que pour un service externe qui peut être atteint par des moyens non-ad-hoc, puisque il serait difficile d'obtenir la position d'un serveur de position si le serveur faisait partie du réseau ad hoc lui-même, cela représente le problème de la poule et de l'œuf . Des protocoles de service de localisation supposent généralement qu'un nœud reçoit sa position géographique à l'aide d'un système de positionnement global (GPS), avec les informations

de localisation. Un émetteur peut transmettre des messages à un récepteur en collaboration avec les nœuds voisins, le problème majeur de ces protocoles est la façon de fournir les services de localisation efficaces pour tous les nœuds dans le réseau . Voici quelques points importants :

1. Comment faire pour contrôler la fréquence de mise à jour de localisation ;
2. Comment diminuer la surcharge lors de la mise à jour de localisation ;
3. Comment réduire le temps de retard d'acheminement.

L'un des principaux usages de services de localisation est les algorithmes de routage basé sur la position. Les protocoles de routage géographique peuvent acheminer les messages de façon plus efficace en fonction de la position géographique du nœud destinataire. Ce qui est fourni par un service de localisation.

2.3 Composantes du service de localisation

Un service de localisation est constitué de deux composantes algorithmiques, (1) la mise à jour de localisation (location update) et (2) la demande de la position (location request). La mise à jour de localisation est responsable de la diffusion d'informations sur l'emplacement actuel d'un nœud cible T à un ensemble de nœuds appelés les serveurs de localisation de T . Si une source S veut découvrir l'emplacement de T , il lance une demande d'emplacement qui est acheminée par le réseau à l'un des serveurs de localisation de T . le serveur de localisation peut soit répondre à la demande elle-même ou la transférer à T pour répondre.

2.4 Taxonomie des services de localisation

La Figure 2.1 résume les différentes classes d'un service de localisation.

Au premier niveau de la taxonomie, les services de localisation peuvent être divisés en deux grandes familles, les approches basées sur l'inondation (Flooding-based) et les approches basé sur quorum ou bien la délégation (quorum-based). Pour plus de détails voir [8]

2.4.1 Services de localisation basés sur les inondations

Les services de localisation basés sur les inondations (en anglais, Flooding-based) sont divisés en des approches proactives et réactives :

Les approches proactives Dans cette approche, chaque nœud (destination) inonde périodiquement son emplacement aux autres nœuds dans le réseau. Chacun gère une table qui contient les positions les plus récentes des autres nœuds. L'intervalle et la portée de ces inondations peuvent être optimisés en fonction de la mobilité des nœuds et l'effet de la distance. Par exemple,

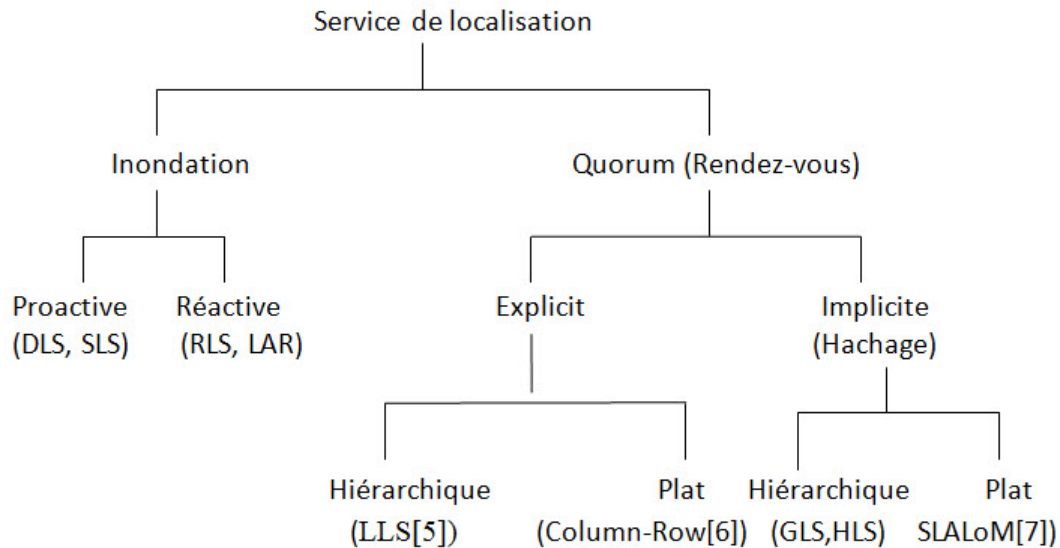


FIGURE 2.1 – La taxonomie des services de localisation [8].

les inondations devraient être plus fréquentes pour les nœuds avec une plus grande mobilité, et les inondations des nœuds lointains peuvent être moins fréquentes que pour les nœuds voisins.

DREAM Location Service [9] et Simple Location Service [9] sont deux exemples de services de localisation basés sur les inondations. Dans SLS, les nœuds transmettent périodiquement à leurs voisins des tableaux contenant les informations de localisation de quelques nœuds dans le système. Tandis que dans DLS, un nœud transmet ses propres informations de localisation aux nœuds proches à un taux particulier, et aux nœuds lointains à un autre taux inférieur. Un nœud met à jour sa table de localisation s’il reçoit un paquet d’informations de localisation plus récent.

Les approches réactives Dans les approches basées sur l’inondation réactive (sur demande) (par exemple LAR [10] et RLS [9]), si un nœud ne peut pas trouver un emplacement récent d’une destination à laquelle il veut d’envoyer des paquets de données, il inonde une requête de la position dans le réseau.

2.4.2 Services de localisation basés sur le quorum

Dans les protocoles à base de quorum, tous les nœuds (émetteurs ou récepteurs potentiels) dans le réseau sont d’accord, implicitement ou explicitement, sur un mappage qui associe l’identifiant unique de chaque nœud à un ou plusieurs autres nœuds du réseau. Les nœuds mappés sont les serveurs de localisation pour ce nœud. Ils seront les nœuds de rendez-vous où les mises à jour de localisation périodiques seront stockées et les requêtes de localisation seront acheminées. Lorsque la recherche atteint l’un des serveurs qui a été mis à jour sur la position correspondant à un identifiant donné, la position correct sera trouvée.

Les protocoles basés sur le quorum sont aussi appelés protocoles basés sur

rendez-vous. Parce que les serveurs de localisation seront les points de rendez-vous pour les mises à jour et les requêtes.

La mise en correspondance des nœuds de quorums peut être faite soit dans un mode statique ou aléatoire, tout en utilisant les identifiants des nœuds ou des informations géographiques et en appliquant les différentes méthodes de hachage. Les deux catégories quorum-implicites et quorum-explicites peuvent être divisés en hiérarchique ou plat. Si une hiérarchie de sous-zones définies de manière récursive est utilisée pour mettre en œuvre le système de quorum on dit que ce système est hiérarchique, autrement on dit qu'il plat.

2.5 Les protocoles de service de localisation

Plusieurs protocoles de localisation ont été proposés dans la littérature. Dans cette section nous allons présenter certains de ces protocoles.

2.5.1 Les protocoles de service de localisation pour les réseaux MANETs

Plusieurs protocoles de localisation ont été proposés pour les réseaux MANETs, nous citons :

DREAM Location Service (DLS)

DLS [9] est un service de localisation proactif semblable au service de localisation utilisé dans DREAM [11], en DLS chaque paquet de localisation (Location Packet « LP »), met à jour les tables de localisation, il contient les coordonnées du nœud source basés sur un système de référence comme GPS, l'identifiant de la source, sa vitesse, et le moment où le LP a été transmis. Chaque nœud mobile dans le réseau ad hoc transmet un LP aux nœuds à proximité à un taux donné, et à des nœuds lointains à un autre taux inférieur. Le taux de transmission des LPs s'adapte en fonction du moment où le nœud mobile a parcouru une distance déterminée de son dernier emplacement de dernière mise à jour. Depuis les nœuds lointains semblent se déplacer plus lentement que les nœuds à proximité. Il n'est pas nécessaire pour un nœud mobile de maintenir la mise à jour des informations de localisation sur des nœuds lointains, puisque en différenciant entre les nœuds voisins et lointains, la surcharge de paquets de localisation est limitée. Dans DLS le taux de transmission de LP est :

Transmettre un LP aux nœuds à proximité chaque :

$$\frac{P}{aV} \text{seconde}$$

Transmettre un LP aux nœuds lointains : toute X LP envoyé aux nœuds à proximité. Ou au moins toute Z secondes.

Où P est la portée de transmission du nœud mobile, V est la vitesse moyenne du nœud mobile, et a est un facteur d'échelle. On note que les LPs à des

nœuds lointains mis à jour tous les nœuds du réseau, y compris les nœuds voisins.

Avantages de DLS La différenciation entre les nœuds proches et loins.

Inconvénients de DLS L'inondation de mise à jour peut surcharger le réseau.

Simple Location Service (SLS)

Les deux protocoles SLS et DLS sont des protocoles proactifs. La différence principale entre eux est dans le type des informations échangées et la distance que l'information propage. Dans SLS, le nœud transmet des tables contenant les positions des autres nœuds à ses voisins, tandis que dans DLS, un nœud inonde sa position à tout autre nœud dans le réseau. Un nœud qui utilise SLS transmet un paquet de localisation à ses voisins à une vitesse donnée, le taux d'un nœud transmettant les LPs s'adapte selon le changement de la vitesse, comme le montre la procédure suivante :

Transmettre un LP aux nœuds à proximité chaque :

$$\frac{P}{aV} \text{seconde}$$

Ou au moins chaque Z secondes.

Chaque LP contient jusqu'à E entrées de la table de localisation du nœud, les entrées E sont choisies de la table dans un mode de tournoi à la ronde. En d'autres termes, chaque LP porte des informations de localisation sur d'autres nœuds du réseau. Comme plusieurs LPs sont transmis, toutes les informations de l'emplacement d'un nœud qui le sait seront partagées avec ses voisins. Chaque nœud reçoit également périodiquement un paquet de localisation de l'un de ses voisins. Le nœud va alors mettre à jour sa table de localisation selon les entrées de la table reçu de telle sorte que chaque information de localisation avec la période la plus récente (entre la table du nœud et la LP reçu soit maintenue.

Avantage de SLS La fréquence de mise à jour s'adapte à la vitesse du nœud.

Inconvénients de SLS L'inondation de mise à jour peut surcharger le réseau.

Reactive Location Service (RLS)

Dans RLS, quand un nœud demande la position d'un autre nœud dans lequel l'information de localisation est inconnue ou expiré pour ce nœud. Le nœud

demandeur demande d'abord à ses voisins l'information de localisation demandée. Si les voisins du nœud ne répondent pas dans un délai d'expiration, le nœud va inonder un paquet de demande dans tout le réseau. Lorsqu'un nœud reçoit un paquet de demande d'emplacement et ne connaît pas les informations d'emplacement demandées, il propage la demande de localisation. Si un nœud reçoit une requête de position et il possède la position requise dans sa table de position, il renvoie un paquet de réponse de localisation via le chemin de source inverse contenu dans le paquet de requête. En d'autres termes, chaque paquet de demande d'emplacement comporte la route complète (une liste de séquences de nœuds).

Avantage de RLS Ce n'est pas nécessaire d'envoyer les mises à jour des nœuds non demandés.

Inconvénient de RLS L'inondation de requête peut surcharger le réseau.

Grid Location Service (GLS)

GLS [12] est un service de localisation hiérarchique distribué, qui utilise le mécanisme d'acheminement gourmand pour acheminer les paquets après l'obtention de l'information de localisation de la destination, ainsi que l'emplacement des nœuds voisins (à l'intérieur de la zone de transmission).

Dans GLS, chaque nœud va choisir un ensemble de nœuds pour servir en tant que serveurs de localisation, et va mettre à jour périodiquement ces serveurs de localisation avec sa position actuelle, assistée par un système d'identification prédéfini et d'une hiérarchie géographique prédéfinie, un nœud envoie ses mises à jour de position à ses serveurs de localisation sans connaître leurs identités réelles, les requêtes pour l'emplacement d'un nœud utilisent également la hiérarchie et les identificateurs prédéfinis pour trouver un serveur de localisation pour ce nœud.

Acheminement gourmand Ce mécanisme est utilisé par GLS pour acheminer les paquets après avoir obtenu la position de la destination. Dans ce mécanisme, chaque nœud peut obtenir sa position géographique à l'aide d'un système de référence tel que GPS. Les positions se composent de latitude et de longitude.

Chaque nœud annonce sa présence par l'envoi périodique d'un paquet de Bonjour (Hello Packet) à ses voisins (autres nœuds à portée radio). Ce paquet contient sa position et sa vitesse. Chaque nœud du réseau maintient une table de position (Location Table) de ses voisins actuels, ce table contient leurs identités et leurs positions géographiques.

L'en-tête d'un paquet contient l'identité de la destination ainsi que sa position géographique, et lorsque un nœud veut transmettre un paquet vers une destination P, il choisit dans sa table de voisins, le voisin le plus proche de P. Ensuite, il envoie le paquet à ce voisin, qui à son tour applique le même algorithme de

transfert. L'acheminement du paquet s'arrête lorsqu'il atteint la destination, La Figure 2.2 montre un exemple d'acheminement gourmand.

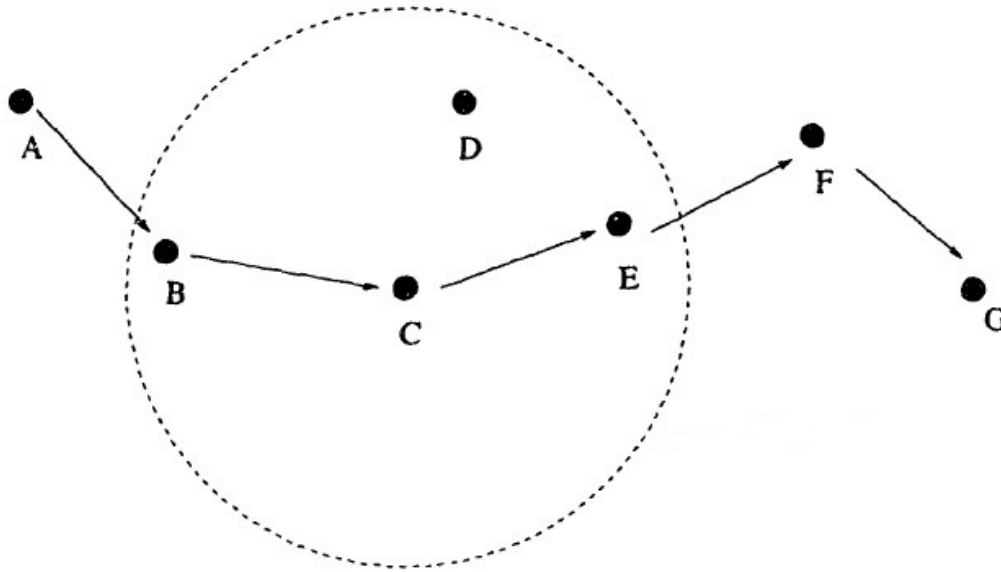


FIGURE 2.2 – Exemple d'acheminement gourmand, le noeud A est la source, et G la destination.

Choix des serveurs En GLS, chaque nœud envoie son emplacement actuel à un ensemble de serveurs de localisation distribués dans tout le réseau. Ces serveurs de localisation ne sont pas spécialement désignés, chaque nœud sert comme un serveur de localisation pour certains nœuds. Les serveurs de localisation d'un nœud sont relativement denses près du nœud mais clairsemés plus loin du nœud. Cela garantit qu'un nœud proche n'a pas besoin d'interroger un serveur lointain, alors que la destination est proche de lui et les requêtes longues distances ne sont pas pénalisées disproportionnée. Les longueurs de chemin d'accès de requête sont proportionnelles aux longueurs de chemin d'accès de données.

En répliquant la connaissance de l'emplacement actuel d'un nœud à un petit sous-ensemble de nœuds du réseau (serveurs de localisation), GLS offre un service de localisation distribué. Si un nœud A veut communiquer avec le nœud B, il faut d'abord trouver un nœud qui connaît l'emplacement de B, le problème est que A connaît seulement l'identificateur de B. Pour trouver les serveurs de B, A utilise le même système utilisé par B quand il recrute ses serveurs de localisation.

Dans la phase de sélection des serveurs de localisation, GLS évite des techniques telles que l'élection de leader pour déterminer la responsabilité de serveur de localisation, puisque ces mécanismes vont mettre beaucoup de pression sur les nœuds assez malchanceux pour être élu en tant que leader ou placés à des niveaux supérieurs de la hiérarchie. GLS permet à un nœud X de sélectionner un ensemble de serveurs, de façon probabiliste, cette ensemble est différent de l'ensemble de serveurs sélectionnés par d'autres nœuds. Les nœuds à la recherche de X a besoin de savoir l'ID de X afin d'être en mesure de trouver les serveurs de localisation de X, afin de trouver X, ces nœuds appliquent le

même algorithme que X a appliqué pour sélectionner ses serveurs en premier lieu.

En utilisant un unique attribué ID, tels que les adresses IP ou toute autre unique attribué, GLS utiliser une fonction de hachage forte pour déterminer les serveurs de localisation d'un nœud donné, il fait ça pour balance l'emplacement de serveur uniformément sur tous les nœuds.

Lorsque le réseau est chargé, tous les nœuds connaissent le même partitionnement global de la surface occupée par le réseau, cette zone est divisée en une hiérarchie de grilles de carrés de taille croissante, comme le montre la Figure 2.3. Le plus petit carré on l'appelle carré d'ordre-1, un carré d'ordre-2 est constitué par la combinaison de quatre carrés d'ordre-1, et ainsi de suite. Il est important de remarquer que ce n'est pas vrai que chaque carré composé de quatre carrés d'ordre n est aussi un carré d'ordre- $(n+1)$. Au contraire, pour éviter tout chevauchement, un carré d'ordre- n particulier fait partie qu'un seul carré d'ordre- $(n+1)$, pas quatre.

Un nœud est situé dans un carré de chaque taille, en divisant la superficie occupée par le réseau par ce système de tailles croissantes carrées, GLS fournit un contexte dans lequel un nœud sélectionne moins en moins de serveurs de localisation à de plus grandes distances.

Si un nœud arbitraire dit B, veut déterminer les nœuds (serveurs de localisation). Selon la hiérarchie de grille prédéterminée, B utilisera le même algorithme utilisé par d'autres nœuds qui veulent déterminer la position de B, l'algorithme est la suivant : B va choisir les nœuds avec des ID "proches" de sa propre identité pour servir comme serveur de localisation, GLS définir le nœud le plus proche de B dans l'ID d'être le nœud avec le moins ID supérieur à B. L'espace des IDs est considéré comme circulaire, (par exemple, 5 est plus proche de 20 que 18 est à 20). B choisira un serveur de localisation dans chaque carré voisin du carré qui contient B, par exemple dans la Figure 2.4, le nœud B choisir trois serveurs dans chaque niveau de la hiérarchie de grille.

Les requêtes La Figure 2.5 présente l'état d'un réseau de grille, une fois tous les nœuds ont fourni leurs coordonnées pour les nœuds qui agiront comme leurs serveurs de localisation. Avec l'état du réseau complet comme référence, nous pouvons revenir à la question comment A va trouver la position de B. Pour trouver B en utilisant l'acheminement gourmand "A" aura le choix entre les nœuds pour lesquels A a des informations de localisation, le nœud qui a le moins ID supérieur ou égal à B, ce nœud à son tour faire de même, et ainsi de suite, finalement, la requête atteindra un serveur de localisation de B, qui transmet la requête à B lui-même, B envoie ensuite en utilisant l'acheminement gourmand sa position à A (l'emplacement de A est dans la requête).

Avantage de GLS Il utilise une hiérarchie donc les communications des nœuds proches seront satisfaites localement (dans la région le plus basse qui contient les deux nœuds).

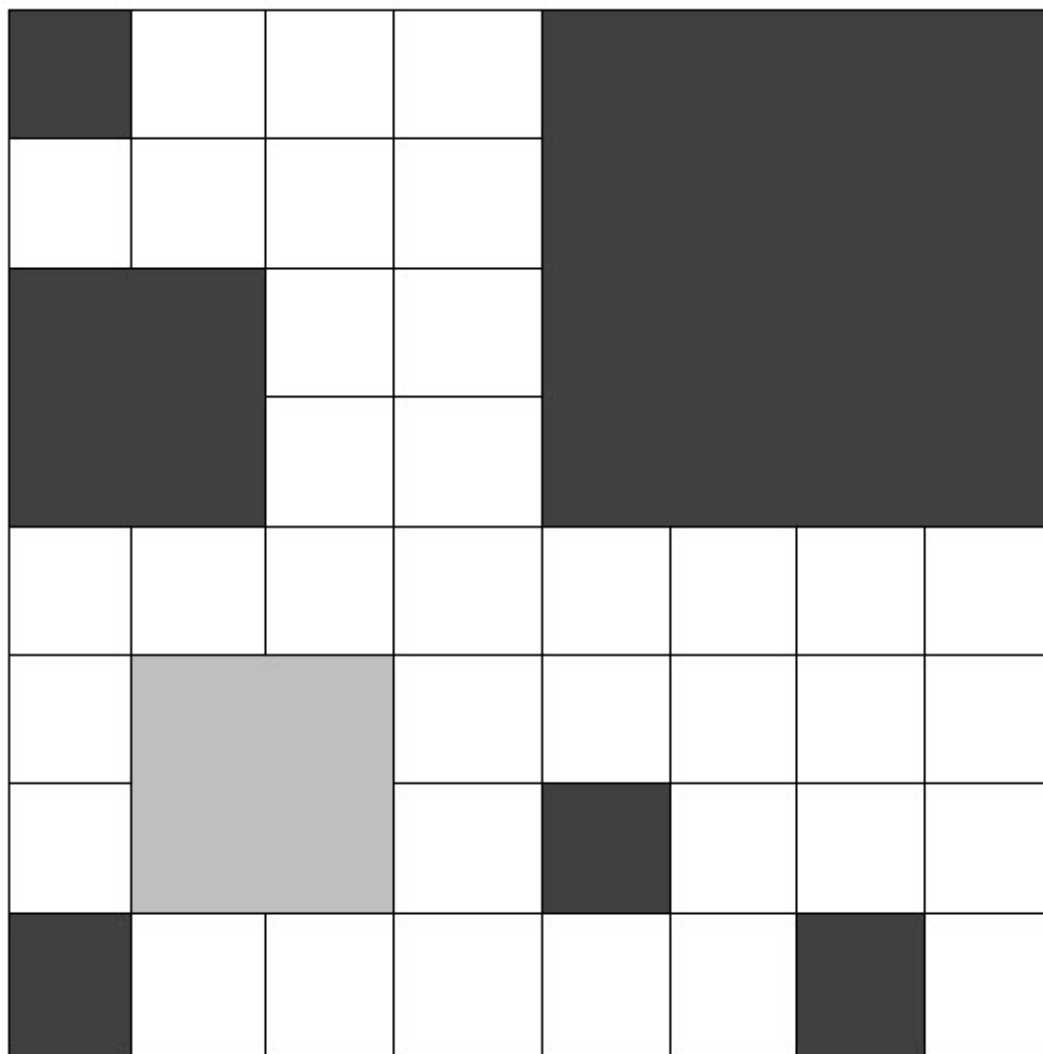


FIGURE 2.3 – Des exemples de carrés valides de tailles différentes (les noirs).
Le gris n'est pas un carré ordre 2.

Inconvénients de GLS L'inconvénient majeur de GLS est que l'acheminement gourmand va échouer face à des topologies indésirables, par exemple, un paquet peut arriver à un nœud qui ne connaît pas un voisin plus près que lui-même à la destination finale, par exemple, dans la Figure 2.6, C pense qu'il y a pas un chemin vers G et ne peut pas transmettre les paquets en cours. Cependant, nous pouvons voir qu'il y a un chemin d'accès valide par C->D->E->F->G, car D est plus loin de G que C, C refuse de transmettre les paquets vers l'arrière donc l'acheminement gourmand échoue. Cette impasse indique qu'il y a un trou «hole» dans la répartition géographique des nœuds, dans ce cas, GLS simplement s'abandonne et ne tente pas de récupérer de l'erreur.

Hierarchical Location Service (HLS)

Comme son nom l'indique HLS [13] est un service de localisation hiérarchique semblable à GLS dans son essence avec quelques différences. Parmi ces différences est que HLS est très robuste aux échecs des nœuds et à la mobilité, car il utilise des régions pour sélectionner les serveurs de localisation et n'est pas

	90	38					
70			37		50	45	
91	62	5			51		11
	1				35	19	
26		41	63		41	72	
87	44	7	2	B: 17		28	10
	98	55	61		6	83	20
32					21		
81	31	43	12		76	84	

FIGURE 2.4 – Les carrés encastrés sont des régions dans lesquelles B cherchera un serveur de localisation. Les nœuds qui deviennent les serveurs de localisation de B sont encadrés et apparaissent en gras. [12]

une chaîne des nœuds mobiles comme GLS.

Partitionnement de la zone HLS divise la zone occupée par les réseaux en des cellules, la taille et la forme des cellules n'a pas d'importance, la seule condition est que si un nœud à l'intérieur d'une cellule, il doit être en mesure d'envoyer des paquets à tout nœud à l'intérieur de cette cellule, Ceci peut être réalisé en choisissant une taille de cellule approprié, c'est à dire, la distance entre deux points quelconques dans la cellule doit être plus petite que la zone de portée radio, tout nœud participant au réseau doit connaître cette partitionnement au début. Les cellules sont regroupées en régions hiérarchiquement de différents niveaux. Un certain nombre de cellules de niveau 0 former une région de niveau 1, un certain nombre de régions de niveau 1 forme une zone de niveau 2 et ainsi de suite, les régions de même niveau ne doivent pas se chevaucher, c'est à dire, chaque région de niveau n est membre d'exactly une région de niveau n + 1. Un exemple de partitionnement de la zone est représenté sur la Figure 2.7.

Cellules responsables HLS place des informations de localisation pour un nœud T à un ensemble de cellules. On appelle ces cellules les cellules responsables (RC) de T. Dans ce qui suit, on dit que "T met à jour une cellule

	70, 72, 76, 81 82, 84, 87	1, 5, 6, 10, 12 14, 37, 62, 70 90, 91				19, 35, 37, 45 50, 51, 82	
	A: 90	38				39	
1, 5, 16, 37, 62 63, 90, 91			16, 17, 19, 21 23, 26, 28, 31 32, 33	19, 35, 39, 45 51, 82		39, 41, 43	
70			37	50		45	
1, 6, 2, 70, 90	1, 5, 16, 37, 39 41, 43, 45, 50 51, 55, 61, 91	1, 2, 16, 37, 62 70, 90, 91			35, 39, 45, 50		19, 35, 39, 45 50, 51, 55, 61 62, 63, 70, 72 76, 81
91	62	5			51		11
	62, 91, 98				19, 20, 21, 23 26, 28, 31, 32 51, 82	1, 2, 5, 6, 10, 12 14, 16, 17, 82 84, 87, 90, 91 98	
	1				35	19	
14, 17, 19, 20 21, 23, 26, 87		2, 17, 23, 63	2, 17, 23, 26 31, 32, 43, 55 61, 62	28, 31, 32, 35 37, 39		10, 20, 21, 28 41, 43, 45, 50 51, 55, 61, 62 63, 70	72
26		23	63	41		72	
14, 23, 31, 32 43, 55, 61, 63 81, 82, 84	2, 12, 26, 87 98	1, 17, 23, 63, 81 87, 98	2, 12, 14, 16 23, 63		6, 10, 20, 21 23, 26, 41, 72 76, 84	6, 72, 76, 84	
87	14	2	B: 17		28	10	
31, 81, 98	31, 32, 81, 87 90, 91	12, 43, 45, 50 51, 61	12, 43, 55	1, 2, 5, 21, 76 84, 87, 90, 91 98	6, 10, 20, 76	6, 10, 12, 14 16, 17, 19, 84	
32	98	55	61	6	21	20	
31, 32, 43, 55 61, 63, 70, 72 76, 98	2, 12, 14, 17 23, 26, 28, 32 81, 98	12, 14, 17, 23 26, 31, 32, 35 37, 39, 41, 55 61	2, 5, 6, 10, 43 55, 61, 63, 81 87, 98		6, 21, 28, 41 72	20, 21, 28, 41 72, 76, 81, 82	
81	31	43	12		A: 76	84	

FIGURE 2.5 – L'organisation de serveur de localisation d'un réseau entier. Chaque noeud est affiché avec la liste de noeuds pour lesquels il a des informations de localisation ; les serveurs de localisation de B sont indiqués en gras. Deux requêtes possibles par A pour l'emplacement de B sont présentées. [12]

responsable R" lorsque T envoie un paquet de mise à jour à un noeud arbitraire à l'intérieur ou à proximité de R, ce noeud devient serveur de localisation pour T. Il est possible que des mises à jour ultérieures arrivent à différents noeuds à l'intérieur de cette cellule, par exemple parce que les noeuds se déplacent. Une cellule peut donc contenir plusieurs serveurs de localisation d'un noeud. Contrairement à GLS qui utilisent l'acheminement gourmand, les auteurs de HLS supposent que tous routage nécessaire pour HLS sont réalisés avec un protocole de routage basé sur la position comme GPSR [14]. Un noeud T sélectionne une cellule responsable de chaque niveau de la hiérarchie. Pour un niveau donné n, le RC est sélectionné selon l'algorithme suivant : 1. Calculer l'ensemble S (T, n) de cellules appartenant à la région de niveau n qui contient T. 2. Sélectionner l'une de ces cellules avec une fonction de hachage sur la base de caractéristiques de S et l'identifiant de noeud T. Une fonction de hachage possible est la fonction base de modulo : $H(T, n) = ID(T) \text{ mod } S(T, n)$, avec le nombre calculé avec cette fonction de hachage, l'une des cellules par S (T, n) est choisie. A la suite de la sélection ci-dessus, T a exactement une cellule

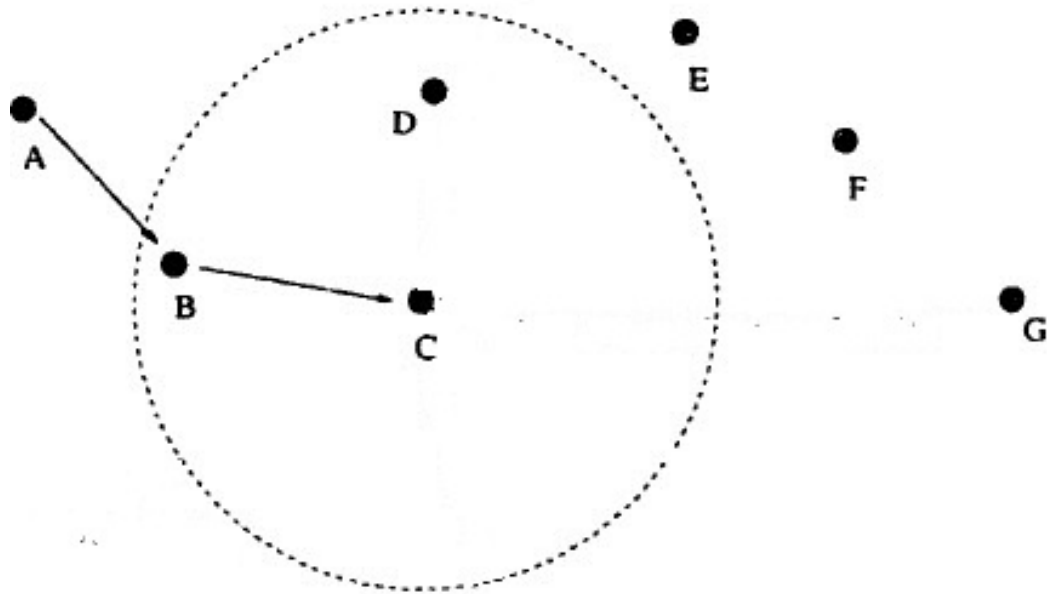


FIGURE 2.6 – L'échec d'acheminement gourmand. Le nœud C ne trouve pas un autre nœud qui est plus proche de G que lui-même, même si il ya un chemin d'accès valide (C ->D->E->F-> G) à G

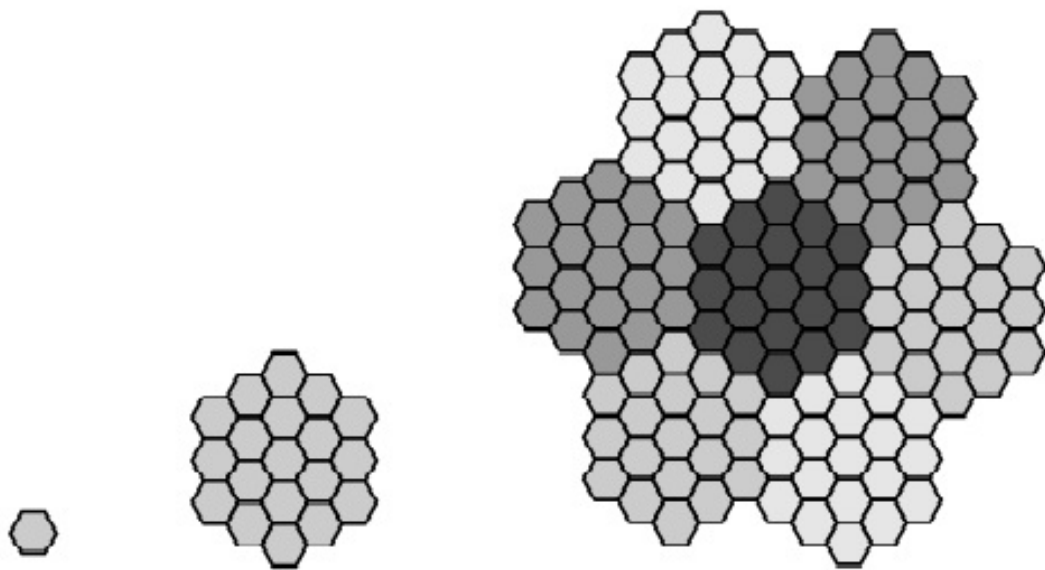


FIGURE 2.7 – Les niveaux d'hierarchie.[13]

responsable à chaque niveau et il est garanti que le RC de T de niveau n et le

nœud T partagent la même région au niveau n. Un exemple pour la sélection des CR est représenté sur la Figure 2.8 pour une hiérarchie à trois niveaux. Les grands cercles marquent les régions. Les cellules avec les numéros sont les cellules responsables. Le nœud et sa RC au niveau 1 partagent la même région de niveau 1, la RC de niveau 2 situé dans le niveau 2 avec le nœud et ainsi de suite.

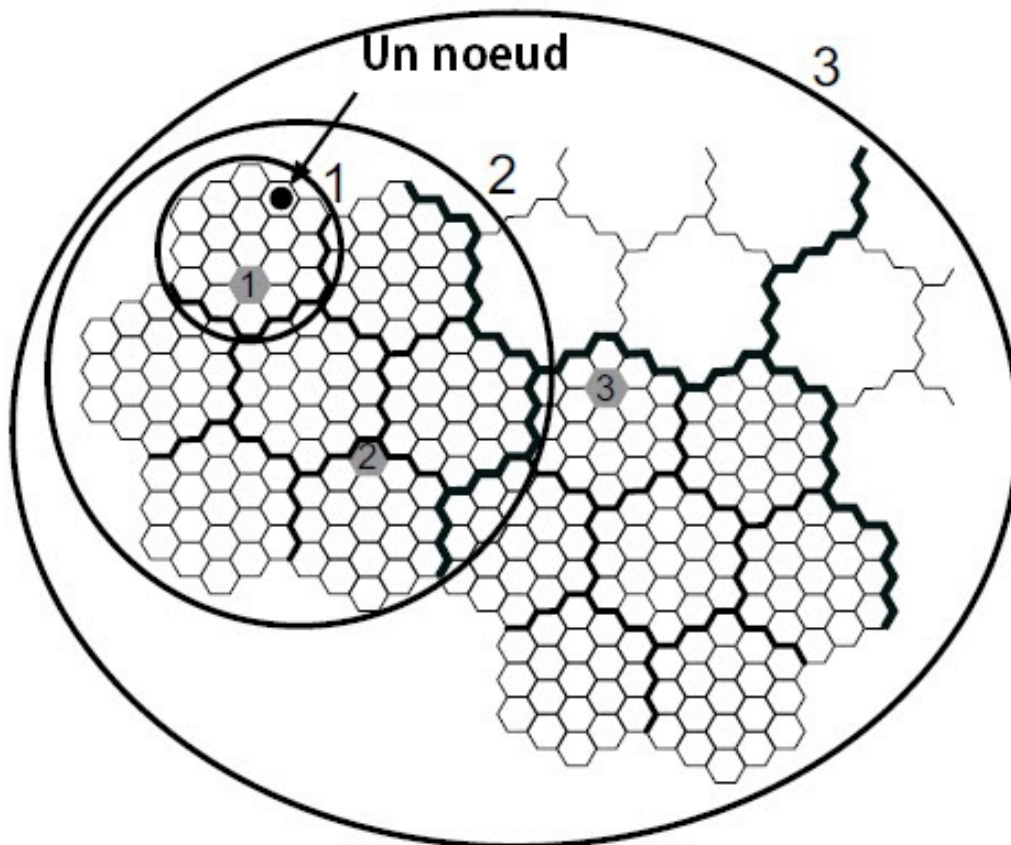


FIGURE 2.8 – Exemple des RCs d'un nœud.[13]

Si on calcule toutes les cellules qui peuvent devenir des CR pour un nœud donné lorsque ce nœud se déplace à travers le réseau, nous allons obtenir une structure en forme d'arbre (voir Figure 2.9), nous appelons cet arbre (l'arbre de candidat), et les cellules sont appelées, les cellules candidates. La racine de l'arbre est le CR unique de plus haut niveau. Les feuilles sont les candidats pour les CR au niveau le plus bas. Lorsque un nœud T se déplace à travers le réseau, un sous-ensemble différent de cellules candidates deviendra cellules responsables. L'arbre de candidat est différent pour chaque nœud, et peut être calculé à partir de la fonction de hachage et l'identificateur de nœud. La sélection des cellules responsables pour un nœud T peut être considérée comme la sélection de la branche de l'arbre.

Mise à jour de la position Pour mettre à jour les serveurs de localisation, un nœud calcule ses CRs, les mises à jour de position sont ensuite envoyées à tous les CRs à la même vitesse. Dans les Figures 2.10 les informations de localisation dans le CR sont représentées comme un pointeur sur la position du nœud, Toutes les cellules responsables doivent être mises à jour chaque fois

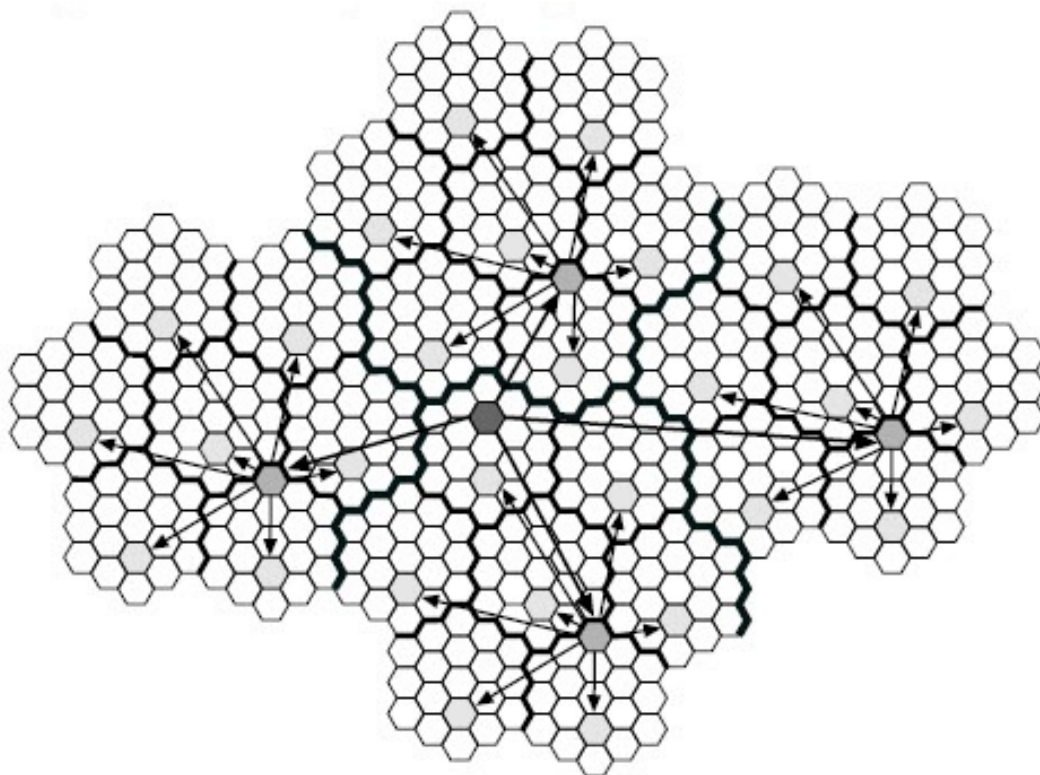


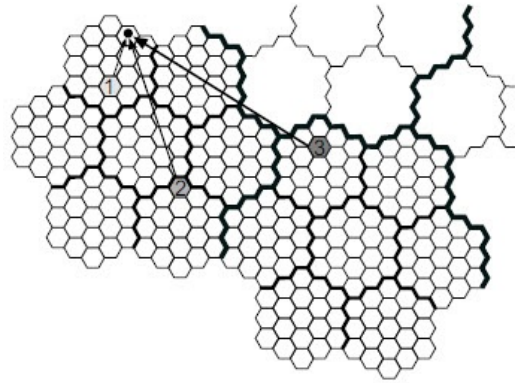
FIGURE 2.9 – L'arbre de candidat. [13]

que le nœud a parcouru une distance assez grande pour rendre l'information de position dans les serveurs de localisation inutiles (Figure 2.10 (b)).

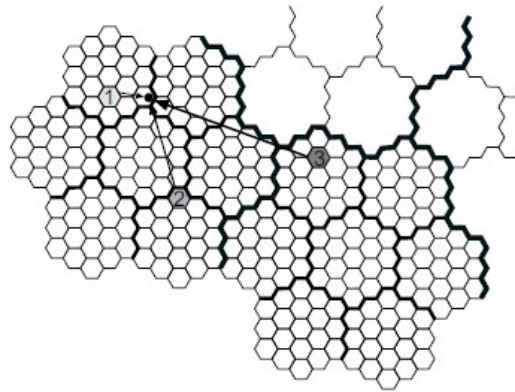
Handovers Etant donné que l'identification d'un serveur de localisation ne dépend que de sa position, un nœud serveur va quitter une cellule responsable, il ne peut plus être serveur de cette CR. Dans ce cas, les informations appartenant à la cellule vient de quitter est remis à cette cellule et traité comme une mise à jour : le paquet de transfert est transmis à un nœud dans ou à proximité de la cellule qui devient le nouveau serveur de localisation.

Requête de position Pour interroger avec succès l'emplacement actuel d'un nœud cible T, la demande d'un nœud source S doit être routée vers un serveur de localisation de T. Lors de l'interrogation de la position de T, S connaît l'identité de T et donc la structure d'arbre de candidat défini par la fonction de hachage et l'ID de T. On sait en outre que T a sélectionné une RC pour chaque région dans laquelle il réside, alors la requête va visiter chaque cellule candidate des régions contenant S. La cellule candidate de la région avec le niveau le plus bas contenant à la fois S et T est par définition une cellule responsable. S calcule la cellule que T choisiraient comme cellule responsable si elle était dans la même région de niveau 1 et envoie sa demande à cette cellule, lorsque le paquet de requête arrive au niveau du premier nœud A dans les limites de la cellule candidatent, elle est traitée comme suit :

1. Un nœud émet la requête à tous les nœuds au sein de la cellule candidate. C'est ce qu'on appelle la demande de Cellcast.



(a) Tous les RCs ont une connaissance de la position des nœuds

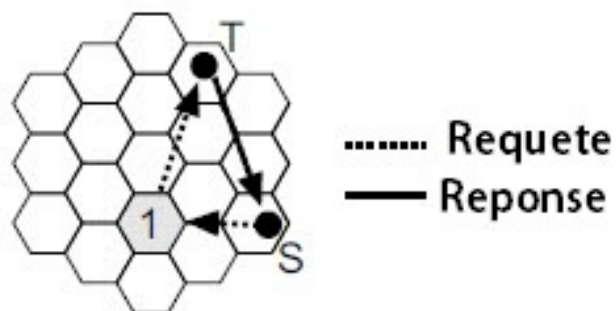


(b) le nœud se déplace : tous les RCs doivent être mis à jour

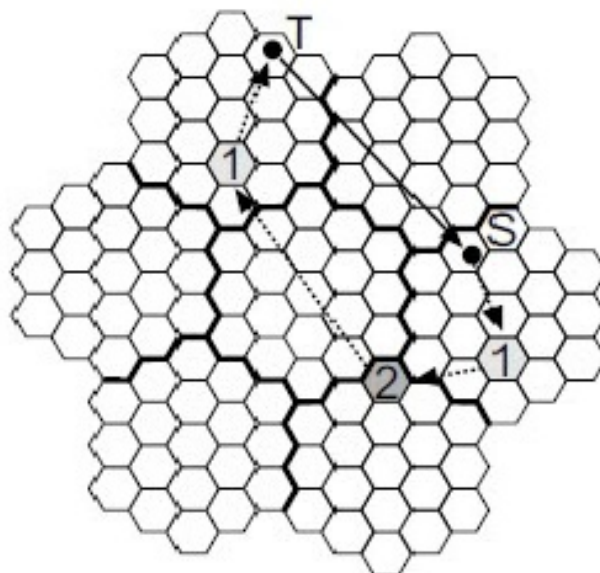
FIGURE 2.10 – La mise à jour. [13]

2. Tout nœud qui reçoit cette demande de Cellcast et a des informations de localisation dans sa table de position envoie une réponse à A.
3. Si A reçoit une réponse à sa requête Cellcast, la demande est transmise à T.
4. Sinon A transmet la demande à la cellule candidate correspondant à un niveau supérieur.

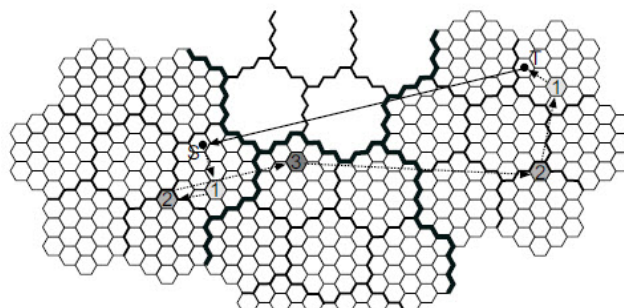
Avec ce mécanisme, la demande est transmise à partir de cellules candidates à une autre cellule candidate jusqu'à ce qu'un serveur de localisation pour T se trouve où la cellule de candidate du plus haut niveau a été atteint e. L'algorithme assure que dans le pire des cas, la demande est transmise a la RC au niveau le plus haut, la cellule candidate qui est garanti d'être également une cellule responsable. La Figure 2.11 illustre trois cas de requête de position, en (a) le nœud source S et le nœud cible T sont situés dans la même région de niveau 1. En (b) S et T sont différentes régions de niveau 1, mais dans la même région de niveau 2. En (c) est la pire situation dans laquelle les nœuds S et T sont dans des régions de niveau (n-1) différent, dans ce cas la demande doit parcourir une longue distance pour trouver une réponse, un paquet de requête est transmis uniquement à l'intérieur des limites de la région la plus basse du niveau où se trouvent les deux nœuds.



(a) Requête locale de niveau-1



(b) La source et la destination situés dans des différentes régions de niveau-1



(c) La source et la destination situés dans des différentes régions de niveau-2

FIGURE 2.11 – Exemple d’une requête ([13])

Avantage de HLS HLS est très robuste contre la défaillance des nœuds, car il des entités logiques (place) comme serveur de localisation et n'est pas une chaîne de nœuds comme dans GLS.

2.5.2 Les protocoles de service de localisation pour les réseaux VANETs

Plusieurs protocoles de localisation ont été proposés pour les réseaux VANETs, nous citons :

Intersection Location Service (ILS)

ILS[15] est un algorithme de service de localisation à base de hachage distribuée pour les réseaux de véhicules. L'algorithme se base sur les caractéristiques qu'une intersection de rues serve comme un serveur de localisation. Un serveur de localisation pour une intersection particulière est l'une des voitures qui sont à proximité de l'intersection et se change avec le temps quand les voitures se déplacent, Chaque véhicule est haché à associer à une intersection, les véhicules vont mis à jour les serveurs par leurs positions lorsqu'ils changent leurs positions en fonction d'un seuil. Quand il ya une recherche d'une voiture, l'emplacement de l'intersection associé à la voiture désignée est d'abord trouvé par hachage, en interrogeant le serveur de localisation du point d'intersection, la position de la voiture désignée est connue. Le routage géographique est utilisé après l'obtention de la position. En ILS, les intersections des rues couvertes par le réseau sont numérotés de 0 à N. Chaque nœud se voit attribuer un serveur de localisation unique, par l'hachage d'identifiant unique du noeud à un identificateur d'intersection. Les voitures dans le rayon r de l'intersection M cachent les informations de position de tous les véhicules K pour lesquels $M = \text{Hash}(K)$ Où Hash est une fonction de hachage qui peut distribuer les voitures de manière égale entre les intersections. Une fois que le serveur de localisation est déterminé, la requête émise par une voiture sera acheminé vers le serveur de localisation cible avec un protocole de routage basé sur la position comme GPSR [14]. Le routage basé sur la position est effectuée par les voitures qui servent de relais intermédiaires.

Avantage de ILS La charge de service est distribuée sur tout le réseau.

Inconvénients de ILS La distance entre un nœud et son serveur (intersection) peut être très longue.

DMBLS

DMBLS [16] est un service de localisation hiérarchique pour les environnements urbains de véhicules. DMBLS fait usage de l'information de densité des véhicules aux intersections de la route tout en recrutant des serveurs de localisation dans une zone urbaine donnée. En d'autres termes, les serveurs de

localisation sont des véhicules situés à proximité des intersections qui ont une forte densité de trafic.

Partitionnement de la zone La région géographique qui représente une ville donnée est partitionnée en une hiérarchie de carrés comme illustré sur la Figure 2.12. La place de plus haut niveau (niveau 3) est en outre divisée en quatre régions de niveau 2 qui sont ensuite divisées en des régions de niveau-1. Le serveur de localisation L1 garde les mises à jour des nœuds appartenant à la zone de bas niveau, et les serveurs de localisation L2 et L3 gardent les informations sur tous les nœuds appartenant respectivement aux zones qu'ils ont à gérer.

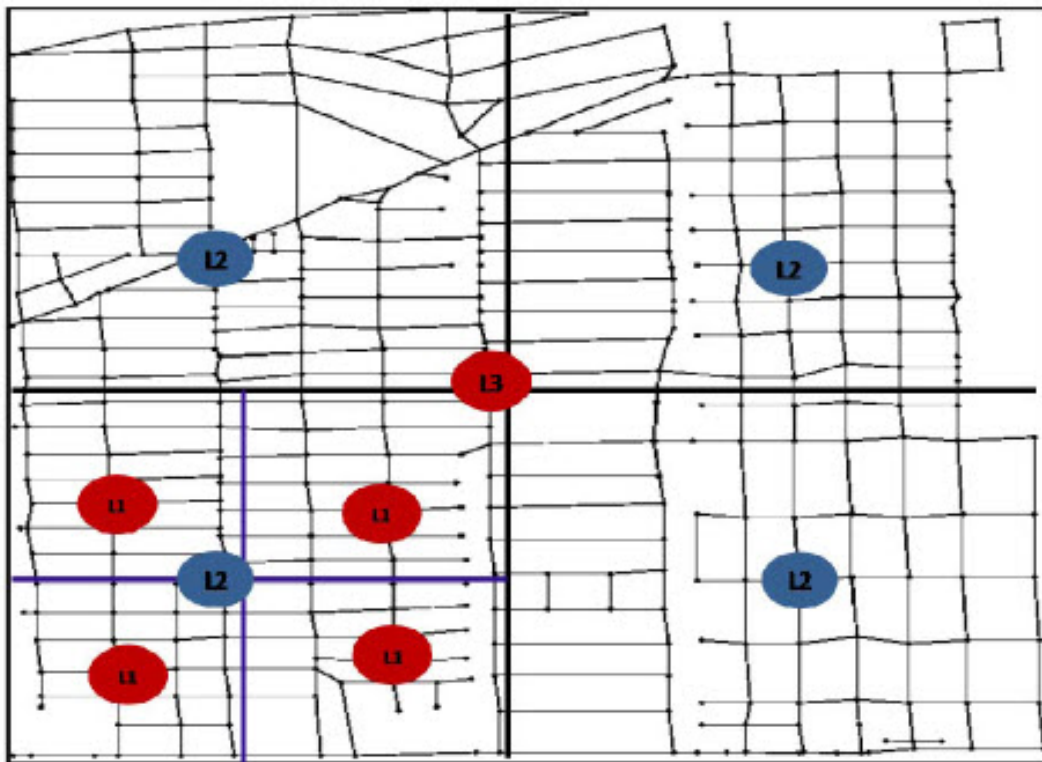


FIGURE 2.12 – Partitionnement géographique de la zone.[16]

Sélection des serveurs Considérant que les véhicules sont susceptibles de se concentrer plus à l'intersection de la route, les nœuds agissant comme serveurs de localisation sont choisis uniquement parmi les nœuds mobiles à proximité des intersections, à cet effet, un ensemble de cellules de localisation (LC) centrées sur les intersections qui ont une forte densité de trafic dans les différents niveaux de la hiérarchie, les véhicules présents dans une cellule de localisation représentent les serveurs potentiels de localisation de la zone correspondante, car une cellule peut contenir plusieurs serveurs de localisation. Le rayon est délimité à la moitié de la portée d'émission R pour faire en sorte que les serveurs de localisation sont dans la portée de communication de l'autre et, par conséquent, ils peuvent échanger des informations. Un véhicule A sélectionne une cellule d'emplacement à chaque niveau de la hiérarchie, pour un niveau donné n , la cellule d'emplacement est choisi en fonction de la procédure suivante :

1. Déterminer la région de niveau n qui contient le véhicule en fonction de ses coordonnées géographiques.
2. Sélectionner l'intersection de la route W_i ayant la plus forte densité de trafic dans la région correspondante, la cellule responsable où les informations d'emplacement seront mises à niveau n est défini comme le cercle avec le centre W_i et un rayon égal à $0,5 R$.

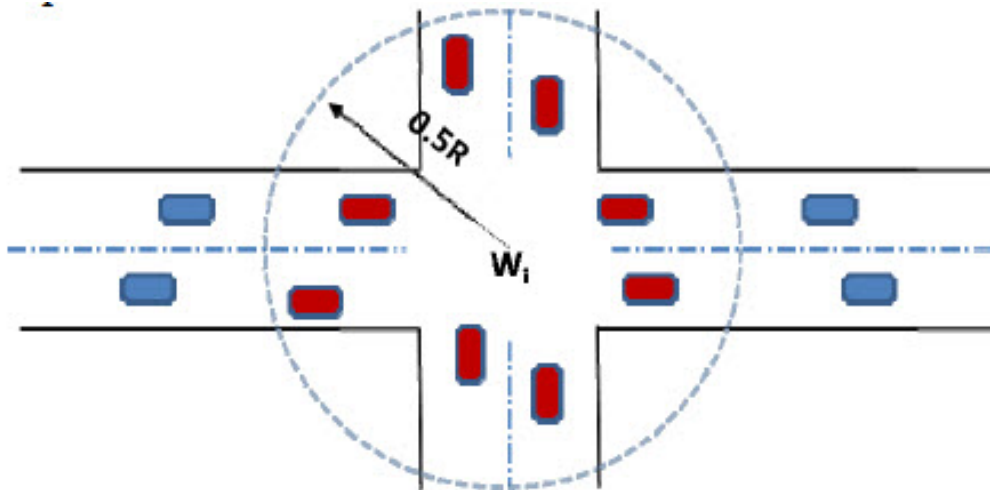


FIGURE 2.13 – La sélection des serveurs de localisation à l'intérieur de la cellule correspondante à l'intersection W_i . [16]

À la suite de la sélection ci-dessus, un véhicule A a exactement trois cellules pour placer ses informations de localisation, une dans chaque niveau. Après avoir déterminé la cellule de localisation avec la meilleure densité dans chaque niveau, un nœud doit mettre à jour ses serveurs de localisation ; à ce niveau, on utilise un routage basé sur la position comme GPSR [14] pour envoyer les paquets de mises à jour de localisation vers les cellules de localisation responsables.

Les Mise à jour En raison de la grande mobilité, les positions des véhicules continuent à changer très rapidement et par conséquent, le serveur de localisation doit être informé pour mettre à jour les informations qu'il stocke, cependant, les mises à jour fréquentes génèrent beaucoup de surcharge qui consomme beaucoup de ressources du réseau et limite les performances de la solution. Par conséquent, un compromis doit être fait entre la surcharge du réseau et de la validité de l'information de localisation. Pour ce faire, chaque nœud envoie une mise à jour d'emplacement à chaque fois qu'il passe une nouvelle intersection et se déplace d'un segment de route à l'autre. Les paquets de mise à jour de localisation contiennent en plus de l'ID de véhicule et la position, la vitesse. Le moment où la mise à jour est générée est également inclus pour estimer la fraîcheur des mises à jour. Lorsque le véhicule s'éloigne de la région de niveau 1 actuelle à une nouvelle région, il doit sélectionner une nouvelle cellule de localisation en utilisant le même algorithme expliqué ci-dessus et envoie sa mise à jour d'emplacement aux nouveaux serveurs de localisation, un ancien serveur de localisation va supprimer l'entrée de nœud correspondant de son cache s'il ne reçoit pas un paquet de mise à jour dans le délai de

validité T . Les mises à jour de la localisation sont transmises successivement pour informer les serveurs de localisation des niveaux supérieurs.

Les Requêtes Quand un nœud source S a besoin de communiquer avec un nœud D hors de sa portée de transmission, il lance une requête de localisation pour obtenir sa position, un paquet de requête contient l'identifiant de la source, de sa position et l'identifiant de la destination, d'abord, la requête est envoyée vers la cellule d'emplacement sélectionné au niveau bas à laquelle le nœud source fait partie. Si A a l'information dans sa base de donnée, il envoie la réponse au nœud S . Sinon il va émettre la requête à tous les nœuds au sein de l'intersection. Tout véhicule qui reçoit la demande et possède les informations d'emplacement dans la mémoire cache renvoie une réponse à A , qui va forwarder la réponse à S . Dans le cas contraire, la demande est transmise à la cellule correspondant sur le niveau supérieur jusqu'à ce que l'identifiant du nœud cible est trouvé. Lorsque les informations demandées à propos de D se trouvent dans la base de données de serveur de localisation, ce dernier traite la demande et envoie un paquet de réponse. (Voir Figure 2.14)

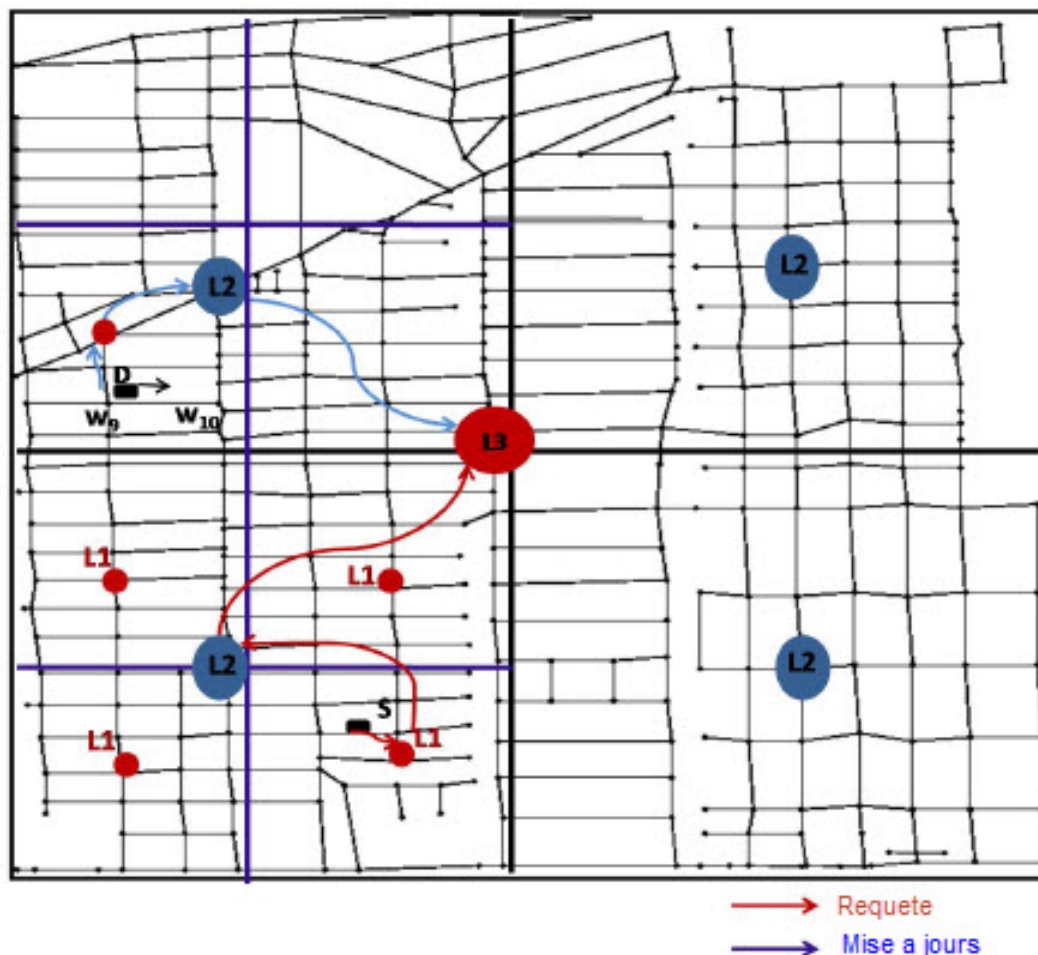


FIGURE 2.14 – Les requête et les mises à jour en DMBLS.[16]

Un véhicule S veut découvrir les informations de localisation d'un véhicule cible D . S envoie une requête à son serveur de localisation correspondant à la région de niveau 1 qui se trouve à w_1 . Tout véhicule plus proche de w_1 , lors de la réception de la requête, il recherche les informations de localisation

de D dans sa base de données, cependant, puisque le noeud D est situé dans une région différente de niveau 2, la demande est transmise à l'autre niveau supérieur, puis de plus haut niveau. Une fois que le serveur de localisation au niveau 3 reçoit le paquet de requête, il recherche l'identifiant de D et conclut que D est situé à la région de plus bas niveau, et il se déplace entre les points de cheminement w_9 et w_{10} , un paquet de réponse contenant l'emplacement géographique de la destination, sa vitesse est renvoyé au nœud source, une fois que le nœud source S reçoit la réponse, il estime la position courante du D.

Avantage de DMBLS L'utilisation d'information de densité pour le choix du serveur.

Inconvénients de DMBLS Les mises à jour envoyées à tous les serveurs de tous les niveaux d'hierarchie, alors que la position exacte ce n'est pas nécessaire dans les niveaux supérieurs.

2.6 Conclusion

Dans ce chapitre, Nous avons discuté les services de localisations et présenté une taxonomie des ces services.

Dans le chapitre suivant, nous allons comparer certains services de localisation et essayer d'améliorer la performance de DMBLS.

Chapitre 3

Étude comparative et proposition

3.1 Introduction

Dans ce chapitre, nous allons comparer certains services de localisation et essayer de proposer une amélioration de la performance de DMBS.

3.2 Etude comparative

Les métriques de performance Pour comparer les services de localisation, nous devons d'abord énumérer les métriques que nous allons utiliser dans cette comparaison. Dans notre cas, nous disons qu'un service de localisation se porte bien si elle respecte les métriques suivantes :

- Extensible (Scalable) : la capacité de service pour poursuivre à fonctionner même si la taille de réseau s'augmente, c'est à dire la capacité de l'algorithme pour traiter une quantité croissante de travail, et sa capacité à être élargi pour tenir en compte cette croissance.
- Robuste contre la haute mobilité : la grande mobilité de nœud ne doit pas affecter le fonctionnement du service.
- La précision de la position.
- Il n'y a pas de goulot d'étranglement, le travail du service de localisation doit être réparti sur les nœuds.
- La défaillance d'un nœud ne devrait pas affecter l'accessibilité d'autres nœuds.
- Les requêtes pour les positions des hôtes à proximité doivent être satisfaites en utilisant une communication locale.
- Le stockage par nœud et le coût d'une communication de service de localisation doivent croître comme une petite fonction du nombre total de nœuds.

	SLS	DLS	RLS	GLS	HLS	ILS	DMBLS
Stratégie de mise à jour	Flood (à ses voisins)	Flooding	Pas de MAJ	Unicast	Geocast	Geocast	Geocast
Stratégie de requête	localement	localement	Flooding	parcours d'arborescence	parcours d'arborescence	Geocast	parcours d'arborescence
Recouvrement en cas de trou dans la topologie	Non	Non	Non	Non	Possible	Possible	Possible
Conçu pour	MANET	MANET	MANET	MANET	MANET	VANET	VANET
Robustesse	Elevée	Elevée	Elevée	Moyenne	Elevée	Moyenne	Elevée
Type	Plat	Plat	Plat	Hierarchie	Hierarchie	Plat	Hierarchie
CI	Faible	Faible	Faible	Moyenne	Elevée	Moyenne	Elevée
LSI				Position + identificateur	Position	Position	position
Scalabilité	Faible	Faible	Faible	Elevée	Elevée	Moyenne	Elevée

TABLE 3.1 – Comparaison entre les service de localisation.

LSI : Location Server Identification (LSI), si le service de localisation utilise les serveurs de localisation, ces serveurs peuvent être identifiés par leur identifiant de nœud (LSI basé sur l'identité) ou par leur position réelle (LSI basé sur la position).

Robustesse : La robustesse d'un service de localisation est considéré comme faible, moyenne ou élevée, selon que l'on prend la défaillance d'un nœud unique, l'échec d'un petit sous-ensemble de tous les nœuds, ou l'échec de tous les nœuds pour rendre la position d'un nœud donnée inaccessible. Dans RLS la robustesse est élevée, parce que la position dépend sur l'inondation dans la demande de localisation, de sorte que la demande de localisation est inondée dans tout le réseau, n'importe quel nœud peut répondre, La même chose pour SLS et DLS,

parce qu'ils utilisent des inondations, ils ne seront pas affectés par l'échec d'un nœud ou une sous ensemble des nœuds. La robustesse de GLS est moyenne puisque la défaillance de serveurs d'un nœud va rendre la localisation d'un nœud indisponible.

Type : Le type indique si la zone de réseau est divisée de façon récursive en une hiérarchie.

CI : La complexité d'implémentation (CI) décrit la façon dont le service de localisation est comprise et comment il est complexe à mettre en œuvre et de le tester. L'implémentation de protocoles hiérarchies est très complexe puisque il est très complexe à manipuler le mouvement des nœuds à travers les régions d'hiérarchies. Par contre il est facile de mettre en œuvre les protocoles basé sur l'inondation.

3.3 Notre contribution

Nous allons essayer d'améliorer la performance de DMBLS en changeant la stratégie de mise à jour, et ajouter la possibilité de prédiction du position.

3.3.1 Stratégie de mise à jour d'origine

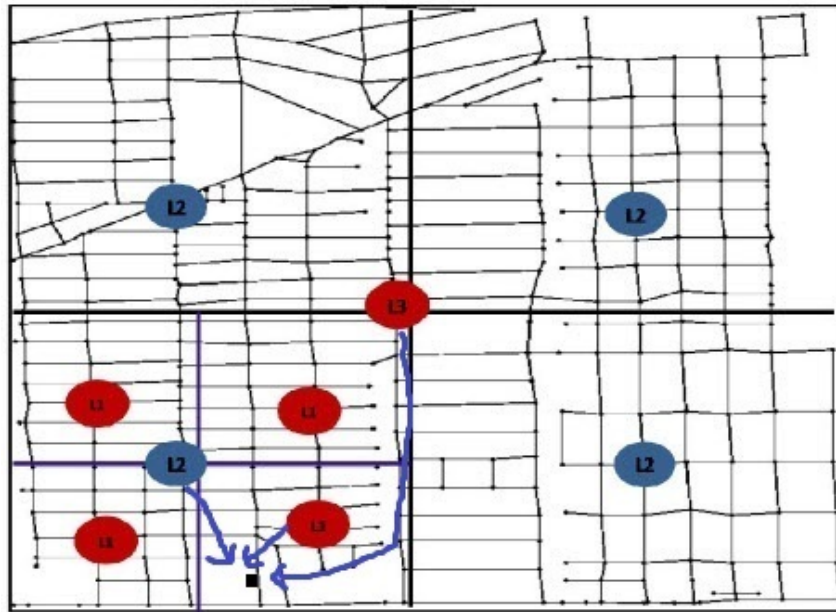
Dans DMBLS, les mises à jour de la localisation sont transmises successivement pour informer les serveurs de localisation de tous les niveaux supérieurs, c'est à dire si on a trois niveaux d' hiérarchie, le nœud va envoyer 3 paquets de mise à jour à chaque fois (voir Figure 3.1), ce que va consommer beaucoup de ressources de réseau.

3.3.2 La stratégie de mise à jour proposé

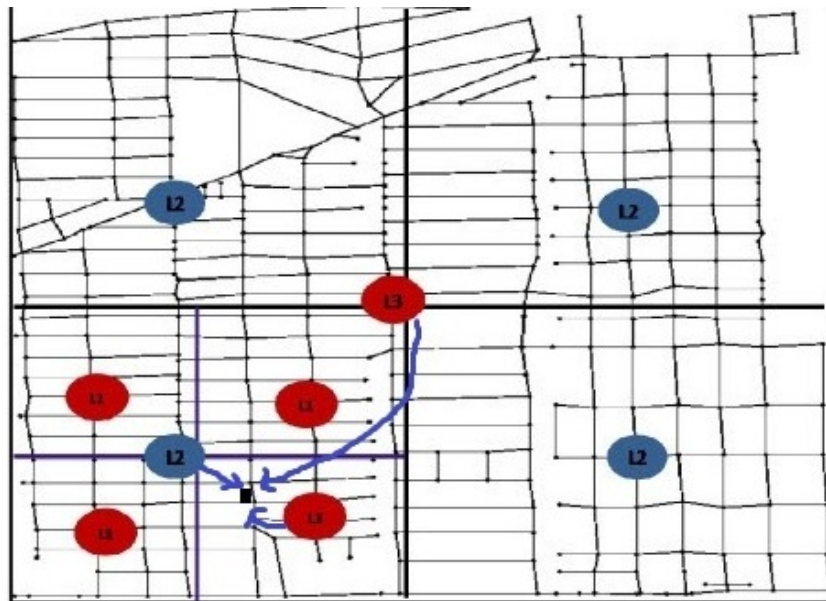
La charge du réseau en raison des mises à jour de localisation peut être encore réduite , si nous supposons que l'information précise n'est pas nécessaire sur les niveaux supérieurs d'hiérarchie, en d'autres termes, les serveurs de localisation ne fourniront que la région du niveau inférieur suivant pour laquelle le véhicule appartient (voir Figure 3.2). Par exemple, un serveur de localisation au niveau N ne sera pas informé si le véhicule se déplace à une nouvelle région de niveau $(n-1)$, afin de réduire le taux des mises à jour envoyées aux serveurs éloignés. Par conséquent, le trafic de mise à jour généré par un nœud est surtout local. La majorité des paquets de mise à jour doivent emprunter des courts chemins, alors que les mises à jour nécessitant le parcours des longs chemins sont rarement envoyées. Selon le modèle de mouvement des nœuds, ce système de mises à jour peut donc réduire les coûts pour les mises à jour de manière significative. Comme le montre la Figure 3.3, les lignes rouges représentent l'information de la position.

3.4 Les solutions proposées

Dans [16] il n'y a pas de détails sur la mise à jour et les requêtes, alors nous allons essayer de compléter ces détails et résoudre des problèmes qui peut se



(a) les serveur de tous les niveaux ont une connaissance sur la position de nœud.



(b) le nœud se déplace, tous les serveurs doivent être mis a jour.

FIGURE 3.1 – la stratégie de mise a jour de DMBLS

produire par exemple quand une intersection qui serve comme un serveur est vides de véhicules.

Les Intersections vides La mise à jour de localisation ou une requête peuvent être envoyées à une cellule qui ne contient pas un nœud ou qui est inaccessible en raison de partitionnement de réseau, pour résoudre ce problème nous proposons :

Solution 1 Si une mise à jour ne peut pas être transmis à l'intersection cible, le nœud qui détecte ça devient un serveur de localisation temporaire (voire figure 3.4(a)), par la suite, le serveur temporaire tente régulièrement de

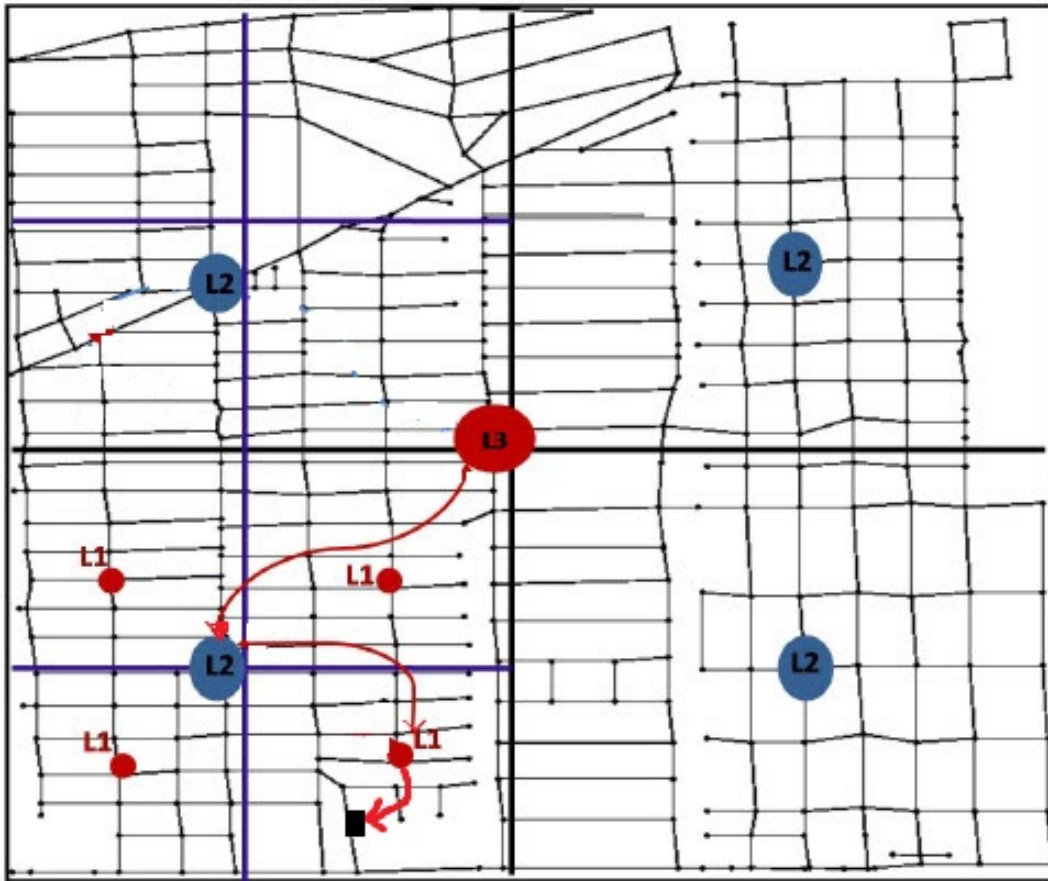


FIGURE 3.2 – les serveurs de localisation ne fourniront que la région de niveau inférieur

remettre les informations sur l'intersection cible (voir Figure 3.4(b)).

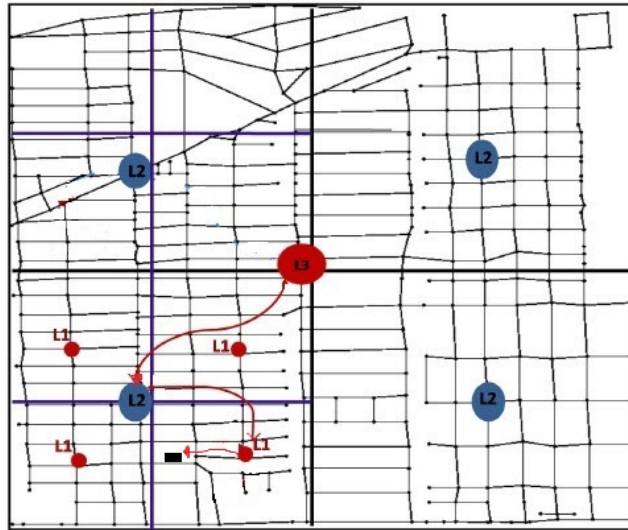
Si une requête ne peut pas atteindre une intersection, il va chercher un serveur de localisation temporaire (voir Figure 3.5).

Si le serveur temporaire de localisation s'éloigne de l'intersection, il va embaucher l'un d'autre véhicule au voisinage pour jouer le rôle d'un serveur téméraire, dans certaine manière que la mise à jour reste près de l'intersection.

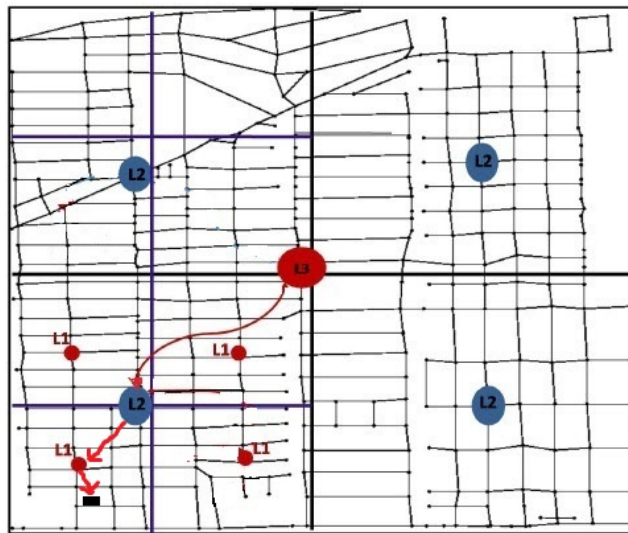
Solution 2 Forwarder la requête au serveur de niveau supérieur si on détecte une intersection vide.

Solution 3 Placer une RSU dans l'intersection qui sert comme serveur de localisation, pour garantir la réponse des requêtes en cas où aucun véhicule est présent dans l'intersection.

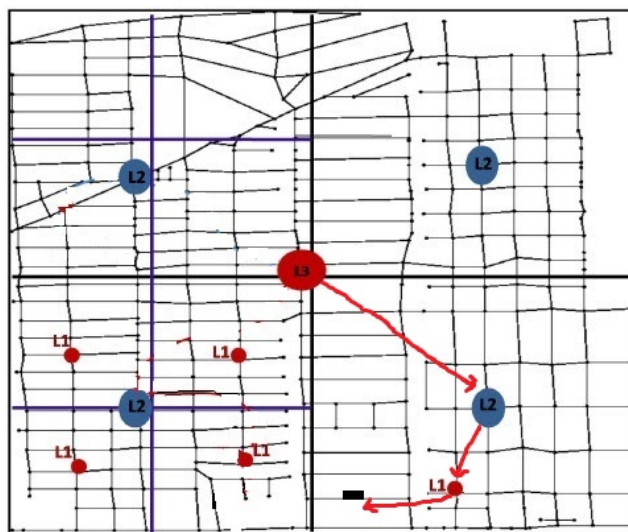
Mais nous allons pas utiliser la troisième solution pour respecter que la solution reste n'est pas basée sur des infrastructure. nous allons utiliser les deux premières solutions, la première solution dans tous les niveaux de la hiérarchie sauf le niveau le plus haut, et la deuxième solution dans le niveau le plus haut d'hiérarchie.



(a) Le nœud se déplace localement dans la région de niveau 1 seul le serveur de niveau 1 qui doit être mis à jour

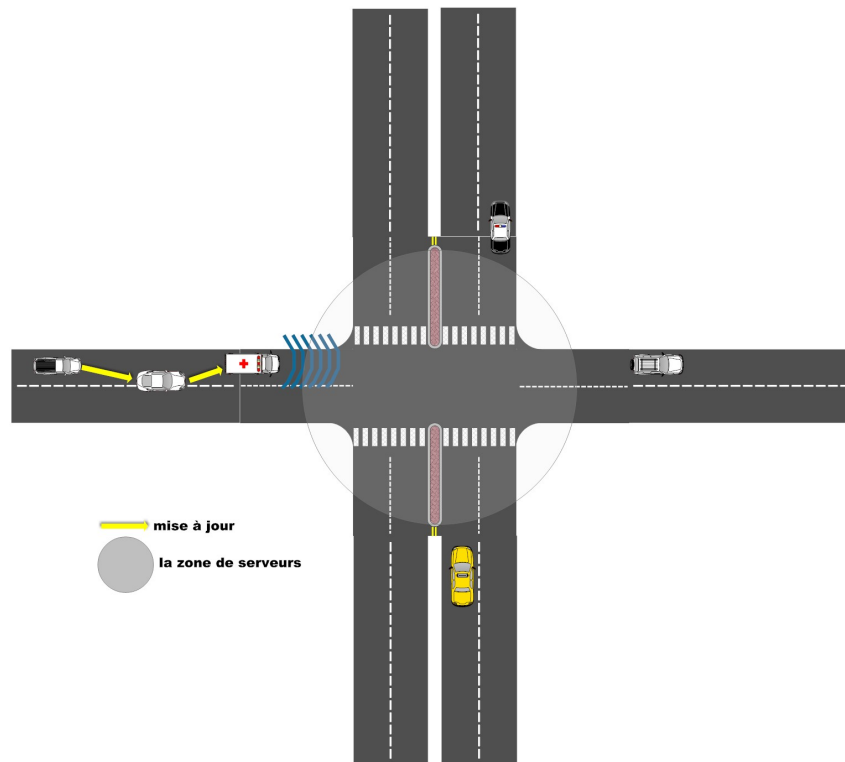


(b) Le nœud se déplace dans la région de niveau 2, les serveurs de niveau 1 et niveau 2 doivent être mis à jour

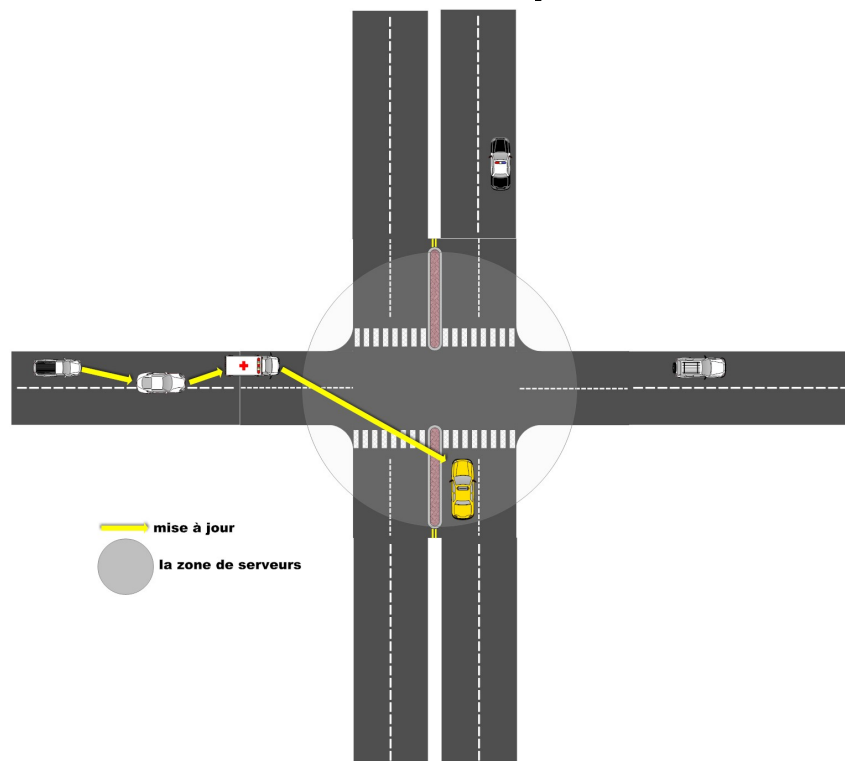


(c) Le nœud se déplace dans la région de niveau 3, tous les serveurs doivent être mis à jours

FIGURE 3.3 – La stratégie de mise a jour proposé



(a) Le véhicule d'ambulance détecte une intersection vide et devient un serveur temporaire



(b) La véhicule de taxi entre la zone de serveur , l'ambulance détecte ça et lui transmette la mise à jour.

FIGURE 3.4 – Solution pour les intersections vides

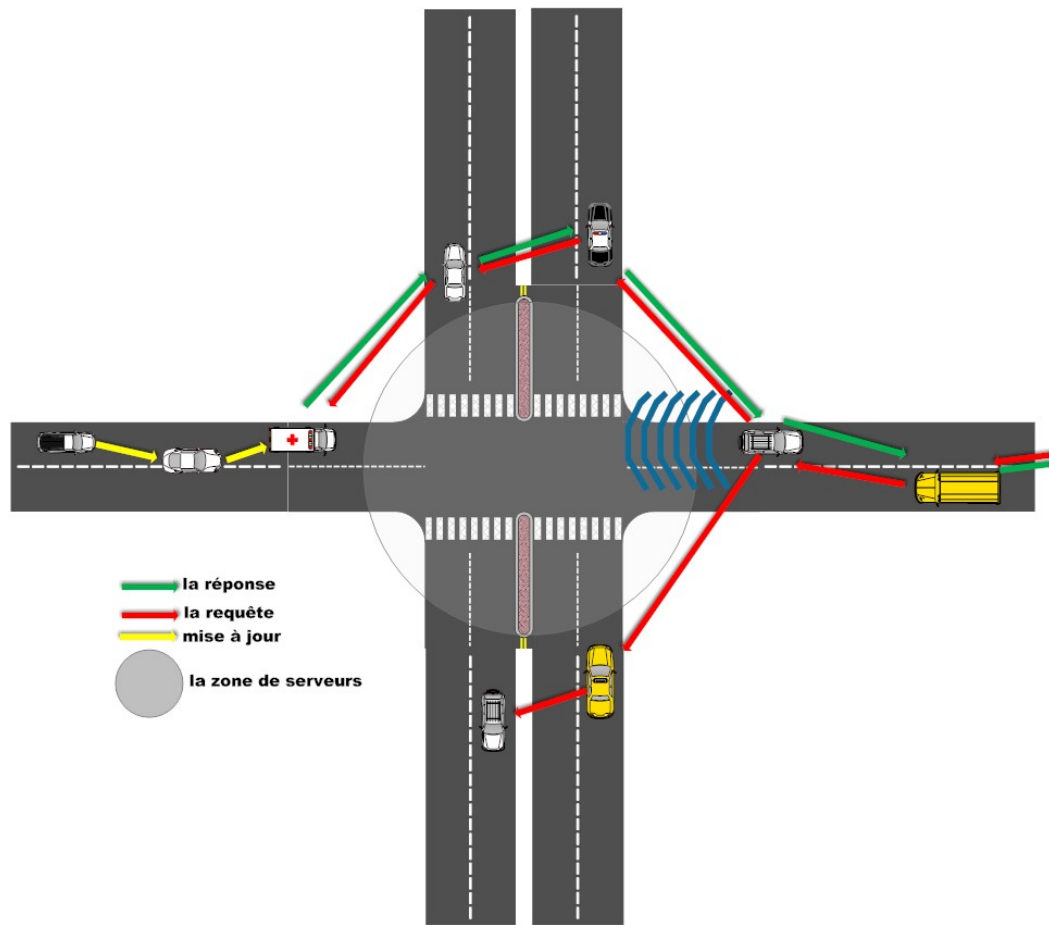


FIGURE 3.5 – La requête détecte une intersection vides alors il va visiter les voisinages pour chercher un serveur temporaire

3.5 Conclusion

Dans ce chapitre nous avons comparé quelques services de localisation et nous avons proposé une amélioration potentielle pour le protocole DMBLS en changeant la stratégie de mise à jour, et ajouter la possibilité de prédiction du position.

Dans le prochain chapitre nous allons simuler le protocole DMBLS en prenant en compte les changements que nous avons proposés, nous allons simuler également d'autres protocoles comme ILS.. Etc., pour faire une comparaison pratique.

Chapitre 4

Les résultats de simulation

4.1 Introduction

Dans ce chapitre nous simulons notre variante de DMBLS, et nous le comparer avec la variante originale de DMBLS et encore d'autres protocoles.

Le code source de ces protocoles n'est pas disponible, donc nous avons été obligé de les implémenter. Les résultats obtenus concernant la fiabilité du service de localisation sont ensuite analysés et comparés.

4.2 Environnement de simulation

Simulateur nous avons implémenté les protocoles en utilisant ns2 [17].

Plateforme nous avons fait les simulations sous la plateforme Linux Ubuntu-12.04 32bit, un CPU Intel i3 et une RAM de 4Go

La zone simulée La simulation est faite sur la carte d'une partie 2000m X 2000m (Almkam, 600 maisons, l'université) (voir Figure 4.1) de la ville de Laghouat obtenue a partir de OpenStreetMaps [18].

Modèle de mobilité Le modèle de mobilité aussi que la vitesse et les type de véhicule est généré en utilisant la simulateur SUMO (Simulation of Urban Mobility) [19].

4.3 Les protocoles simulés

nous avons simulé et évalué la performance des protocoles suivants :

1.Un service centralisé nous avons proposé ce service de localisation naïf pour bien comparer les autres protocoles, autrement dit nous le fait comme une référence de base, dans ce que suit nous l'appelons CLS (Central Location Service). Dans CLS nous considérons qu'il y un seul serveur de localisation dans le centre de la carte (dans la rue central de Mkam), tous les nœuds envoient leurs mises à jour à cette zone, et toute les requêtes seront acheminées vers cette zone, les véhicules présents dans cette zone jouent le rôle d'un serveur de localisation.



FIGURE 4.1 – La zone simulée (ville de Laghouat)

2. ILS

3. DMBLS Nous avons simulé le protocole comme il est spécifié dans [16] sans aucune modification.

4. DMBLS changé Finalement, nous avons simulé DMBLS en prenant en compte les modifications que nous avons proposée dans le chapitre 3, dans ce que suit on l'appelle DMBLSX.

4.4 Les paramètres de simulation

La simulation est faite sous plusieurs conditions et en utilisant des différents paramètres. Nous avons simulé les protocoles En utilisant des fichiers traces de mobilité pour une zone de la ville de Laghouat. Le point de départ et d'arrivée et la vitesse de chaque véhicule est générée aléatoirement. Chaque nœud génère deux requêtes de localisation par seconde, les nœuds interrogés sont sélectionnés de manière aléatoire dans le réseau.

Le tableau suivant résume les paramètres de simulation utilisés :

Paramètre	Valeur
Nombre de nœuds	50, 100, 150
Vitesse	5, 10,20 m/s
Durée de simulation	300 s
Le porté radio	300 m
Taille de la zone	2000m x 2000m
Nombre d'essais	9
Modèle de mobilité	Random Trips (SUMO)
Protocole de la couche MAC	IEEE 802.11
Simulateur	Ns2

TABLE 4.1 – Les paramètres de simulation.

Tous les résultats de performance présentés sont des moyennes de 9 essais de simulation différents, les valeurs dans les cases de tables (tables 4.2, 4.3, 4.4, 4.5) sont des moyennes de 3 essais, la colonne « Moyenne » est le moyenne de ces moyennes. Exemple : Pour faire une simulation on doit considérer trois scénarios de nombre de nœuds (50, 100,200) : Simuler 3 fois avec la vitesse 5 m/s, avec 50 nœuds. Simuler 3 fois avec la vitesse 10 m/s avec un fichier de mobilité différent, avec 50 nœuds. Simuler 3 fois avec la vitesse 20 m/s avec un autre fichier de mobilité, avec 50 nœuds. Puis on doit calculer la moyenne de ces trois valeurs. Faire la même chose avec les densités 100 et 200.

Pour analyser la performance nous avons considéré les métriques suivantes :

4.4.1 Taux de réussite

Le taux de réussite(TR) est le pourcentage de requêtes qui ont été répondu avec succès par le service de localisation. Une requête est répondu avec succès, si le service de localisation peut déterminer la position du nœud cible avec une précision d'au moins de 250 m, soit en recherchant dans la mémoire cache locale ou par envoi d'une requête et la réception d'une réponse. Les figures Figure 4.2 et Figure 4.3 montrent le taux de réussite des requêtes pour des différentes vitesses et des différentes tailles de réseau.

4.4.2 Précision de la position

L'un des principaux paramètres qui montrent l'efficacité d'un service de localisation est la précision (exactitude) de l'information de position. Les figures Figure 4.4 et Figure 4.5 montrent les valeurs d'erreur de précision pour des différentes vitesses et différentes tailles de réseau.

4.4.3 Temps de séjour

Nous considérons ce paramètre pour évaluer le délai demande-réponse des protocoles. Il représente la durée totale mesurée à partir du moment où une demande est générée jusqu'à ce que la réponse est reçue par le nœud source. Les figures Figure 4.6 et Figure 4.7 montrent le temps de séjour de la requête pour des différentes vitesses et différentes tailles de réseau.

4.4.4 Surcharge des mises à jour

C'est le nombre des paquets de mises à jour envoyés. Les résultats présentés sont le nombre total de mises à jour générées par tous les nœuds pendant le temps de simulation divisé par le temps de simulation. Les figures Figure 4.8 et Figure 4.9 montrent le nombre de mises à jour générée par le service avec des différentes vitesses et différentes tailles du réseau.

4.5 Les résultats de simulation

Les tableaux suivants représentent les résultats de simulations obtenues : TR : Taux de réussite. EP : Erreur de précision. TS : Temps de séjour. SMAJ : Surcharge de mise à jour.

	50	100	200	Moyenne
5	TR=48.33% EP=4.2m TS=144.99ms SMAJ=8.94	TR=87% PP=22.14m TS =135ms SMAJ=17.27	TR=84% PP=4.6m TS=70ms SMAJ=33.18	TR=73.11% PP=10.31m TS=116.66ms SMAJ=19.79
10	TR=79.66% PP=21.56m TS=156ms SMAJ=12.7	TR=91.5% PP=24.14m TS=134ms SMAJ=20.2	TR=87.58% PP=7.16m TS=60ms SMAJ=35.8	TR=86.24% PP=17.62m TS=116.66ms SMAJ=22.9
20	TR=84% PP=20.7 TS=307ms SMAJ=15.1	TR=86.33% PP=33.21 TS=48ms SMAJ=25.3	TR=85.5 PP=35.6 TS=9ms SMAJ=39	TR=85.27% PP=29.83m TS=121.33ms SMAJ=26.46
Moyenne	TR=70.66% PP=15.48m TS=202.66ms SMAJ=12.24	TR=88.27% PP=26.49m TS=105.66ms SMAJ=20.92	TR=85.69% PP=15.78m TS=46.33ms SMAJ=35.99	

TABLE 4.2 – Les résultats de simulation de DMBLS.

	50	100	200	Moyenne
5	TR=47.9% PP=6m TS=200ms SMAJ=4.7	TR=85% EP=30m TS=180ms SMAJ=9.07	TR=79.9% EP=5.33m TS=85ms SMAJ=17.2	TR=70.93% EP=13.77m TS=155ms SMAJ=10.32
10	TR=76% EP=25.1m TS=202ms SMAJ=6.9	TR=88% EP=23.8m TS=145ms SMAJ=15.3	TR=85.1% EP=10.2m TS=85ms SMAJ=25	TR=83.03% EP=19.7m TS=144ms SMAJ=15.73
20	TR=70.1% EP=26 TS=307ms SMAJ=10.1	TR=85.4% EP=39.5 TS=48ms SMAJ=23.2	TR=80.2% EP=42.8 TS=21ms SMAJ=36.7	TR=78.56% EP=36.1m TS=125.33ms SMAJ=23.33
Moyenne	TR=64.66% EP=19.03m TS=236.33ms SMAJ=7.23	TR=86.13% EP=31.1m TS=124.33ms SMAJ=15.83	TR=81.7% EP=19.44m TS=80.33ms SMAJ=26.3	

TABLE 4.3 – Les résultats de simulation de DMBSX

	50	100	200	Moyenne
5	TR=43% EP=2.5m TS=400ms SMAJ=4.56	TR=84% EP=6m TS=320ms SMAJ=7.85	TR=76.5% EP=7m TS=180ms SMAJ=17	TR=67.83% EP=5.16m TS=300ms SMAJ=9.80
10	TR=63% EP=19m TS=350ms SMAJ=6.13	TR=85% EP=23.5m TS=280ms SMAJ=13.1	TR=70.1% EP=8m TS=110ms SMAJ=22.3	TR=72.7% EP=16.83m TS= 246.66ms SMAJ=13.84
20	TR=78.66% EP=26.23m TS=280ms SMAJ=9.2	TR=79% EP=25.21m TS=180ms SMAJ=19.5	TR=76.58% EP=10.19m TS=90ms SMAJ=33.5	TR=78.08% EP=20.54m TS=183.33ms SMAJ=20.73
Moyenne	TR=61.55% EP=15.91m TS=343.33ms SMAJ=6.63	TR=82.66% EP=18.23m TS=260ms SMAJ=13.48	TR=74.39% EP=8.39m TS=126.66ms SMAJ=24.26	

TABLE 4.4 – Les résultats de simulation de ILS.

	50	100	200	Moyenne
5	TR=48.33% EP=3.5m TS=100ms SMAJ=4.56	TR=83% EP=7m TS=80ms SMAJ=7.85	TR=80.1% EP=15m TS=50ms SMAJ=17	TR=70.47% EP=8.5m TS=76.66ms SMAJ=9.8
10	TR=80.5% EP=21.3m TS=70ms SMAJ=6.13	TR=87% EP=15m TS=50ms SMAJ=13.1	TR=88% EP=7.5m TS=45ms SMAJ=22.3	TR=85.16% EP=14.6m TS=55ms SMAJ=13.84
20	TR=88.33% EP=25.5m TS=65ms SMAJ=9.2	TR=84.66% EP=27.23m TS=20ms SMAJ=19.5	TR=86.5% EP=10.04m TS=3ms SMAJ=33.5	TR=86.49% EP=20.91m TS=29.33ms SMAJ=20.73
Moyenne	TR=72.38% EP=16.76m TS=78.33ms SMAJ=6.63	TR=84.88% EP=16.41m TS=50ms SMAJ=13.48	TR=84.86% EP=10.83m TS=32.66ms SMAJ=24.26	

TABLE 4.5 – Les résultats de simulation de CLS.

Dans Figure 4.2 , on peut facilement remarquer que tous les protocoles ont un taux de réussite faible à 50 nœuds par rapport aux autres valeurs de densité, cette faiblesse est à cause de la fragmentation de la topologie. Le taux de réussite augmente avec la taille du réseau depuis plusieurs nœuds peuvent agir comme serveurs de localisation lorsque la densité de nœud est élevé. Le taux de réussite augmente aussi avec l'augmentation de vitesse (Figure 4.3), parce que quand les nœuds se déplacent plus vite, les nœuds vont avoir plus des voisins, mais à partir de 18 m/s, le taux de réussite va diminuer à cause de l'expiration rapide des entées de table de position. Dans les deux figures (Figure 4.2 et Figure 4.3), DBMLS surpasse les autres protocoles, c'est parce que est le seul protocole qui envoie les mises à jour à plusieurs places différent à la fois,

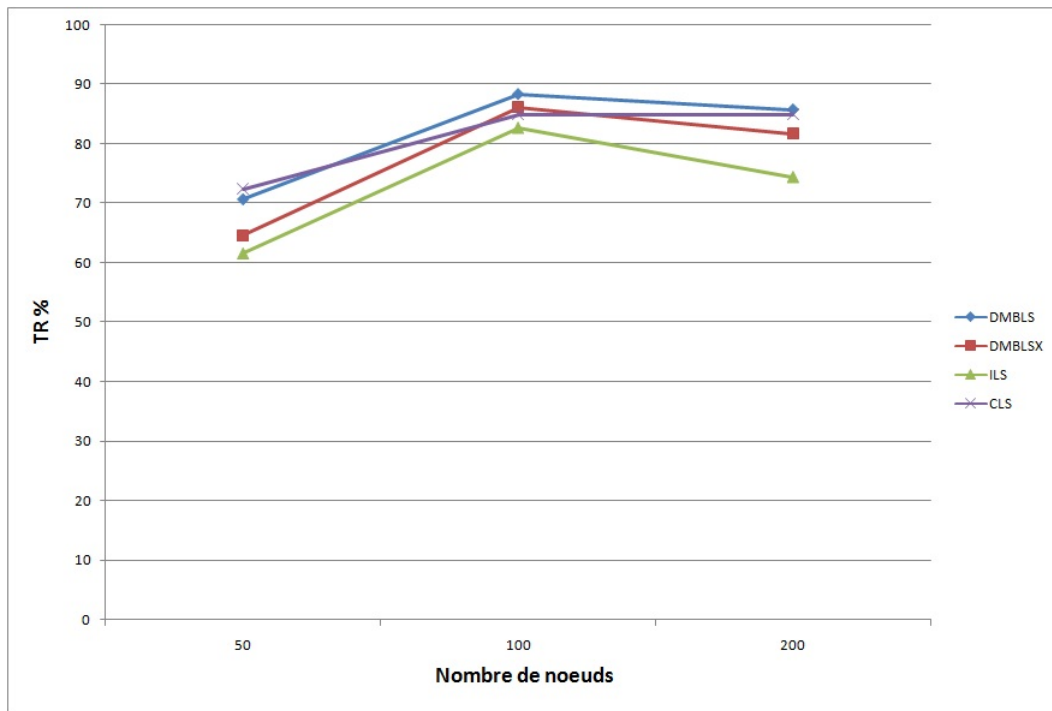


FIGURE 4.2 – TR en fonction de nombre de nœuds

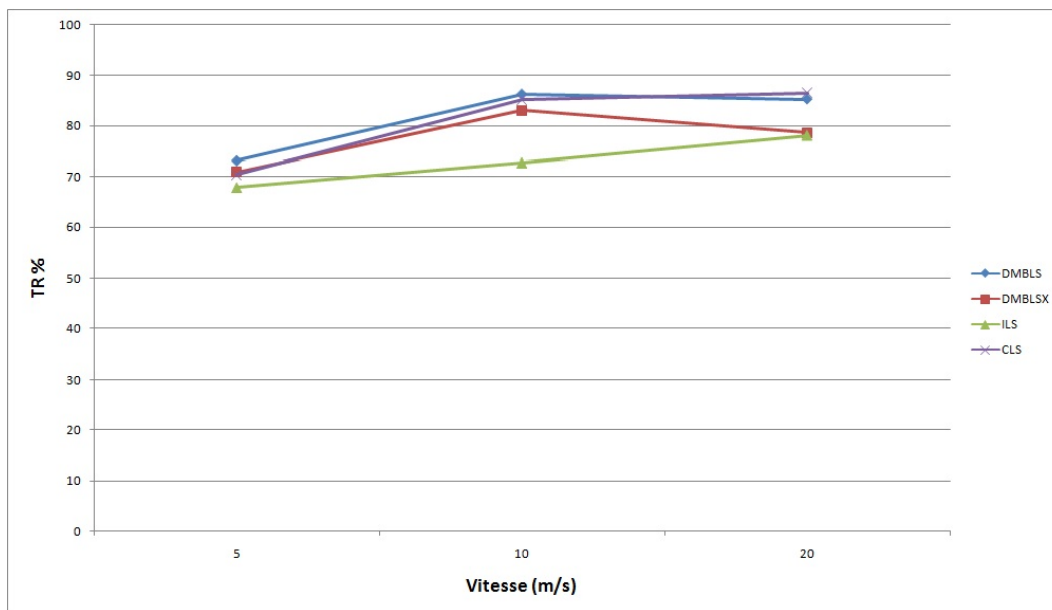


FIGURE 4.3 – TR en fonction de la vitesse.

ce qui conduit à mettre à jour un grand nombre de nœuds (les nœuds relais, et les nœuds serveurs).

Dans les deux figures (Figure 4.4 et Figure 4.5), on remarque que l'erreur de position dans DMBSX est plus grande par rapport aux autres protocoles, c'est parce que toutes les entrées dans la table des nœuds qui ne sont pas dans la même région avec le nœud source sont âgées, car les mises à jour sont envoyées seulement localement (la région la plus basse de la hiérarchie). On peut remarquer également que l'erreur de précision s'augmente avec l'augmentation de la vitesse dans tous les protocoles.

Dans les figures Figure 4.6 et Figure 4.7 ci-dessus on remarque que ILS a le

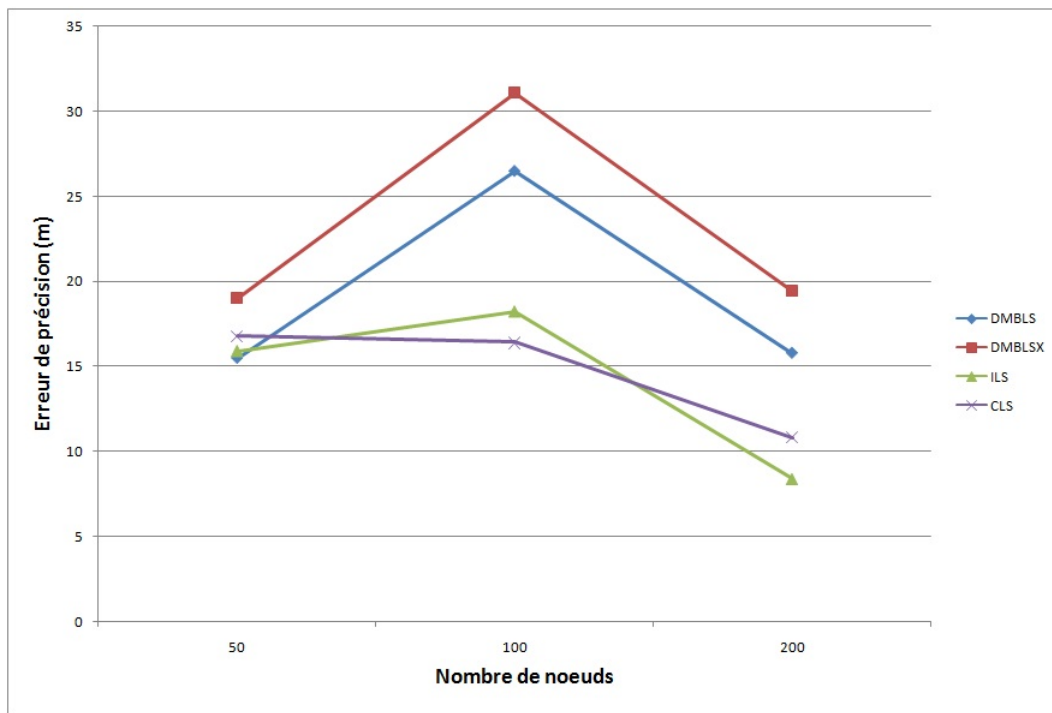


FIGURE 4.4 – EP en fonction de nombre de nœuds

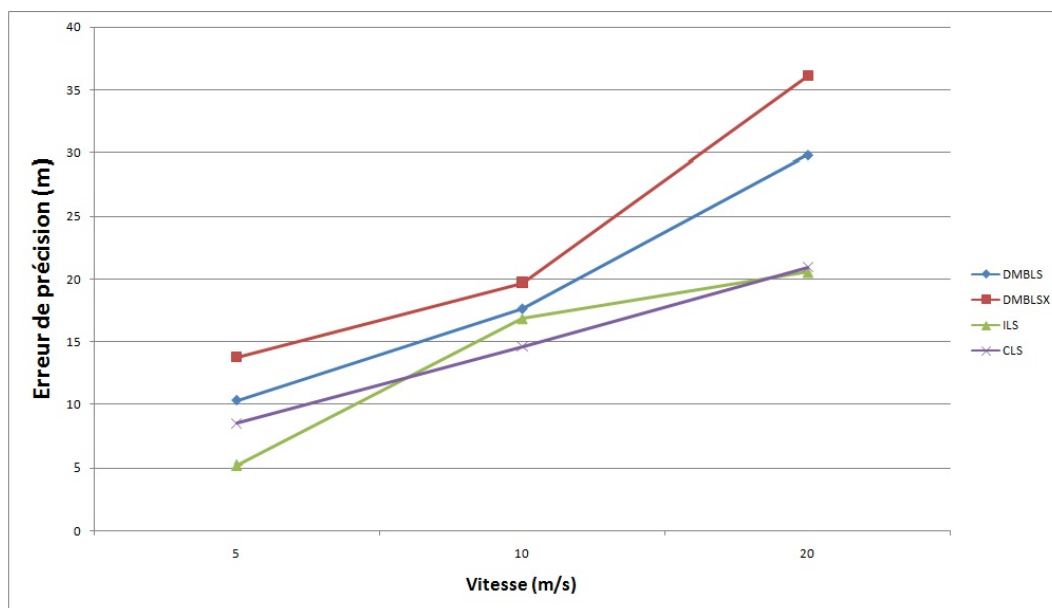


FIGURE 4.5 – EP en fonction de la vitesse.

plus grand temps de séjour, c'est parce que aucune hiérarchie est défini donc la requête va visiter directement la seule intersection qui sert comme serveur de localisation du nœud cherché, par contre dans DMBLS et DBMLSX la requête va visiter d'abord le serveur local, cette approche va diminuer le temps de séjour de commutation local (dans la région de la hiérarchie correspondant). CLS surpasse les autres protocoles en raison du fait que le serveur est situé dans le centre de la carte, cela empêche la requête de parcourir de longues distances, et donc il nécessite moins de temps pour trouver le serveur de localisation le plus proche et de recevoir la réponse.

Dans les figures Figure 4.8 et Figure 4.9 on remarque que DMBLS est le pire

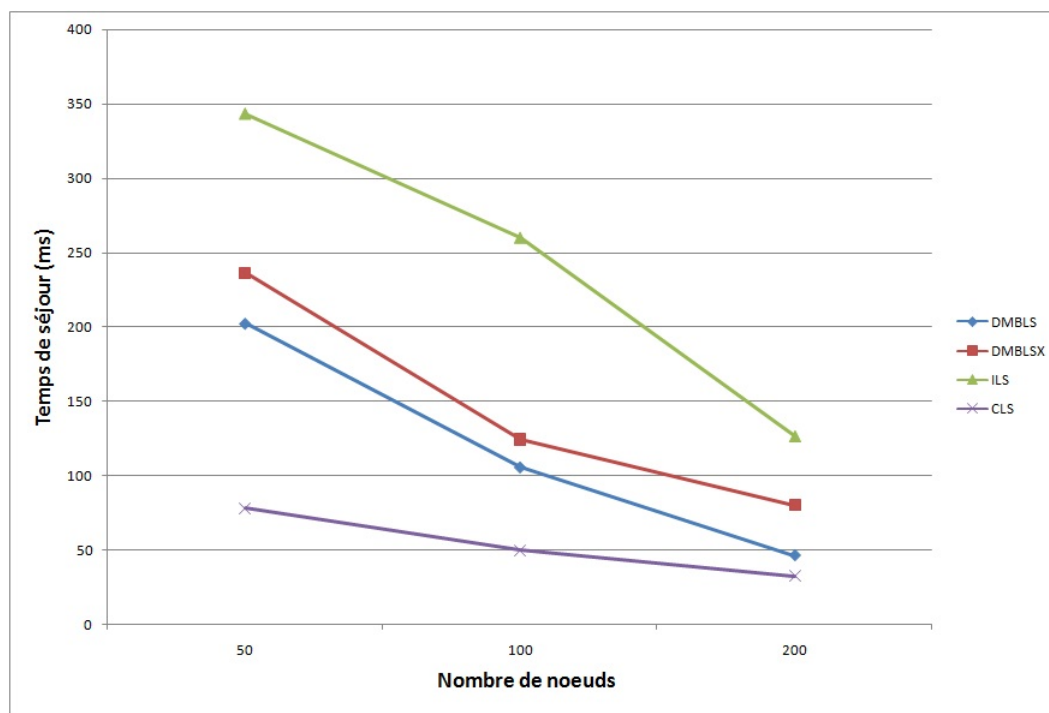


FIGURE 4.6 – TS en fonction de nombre de nœuds

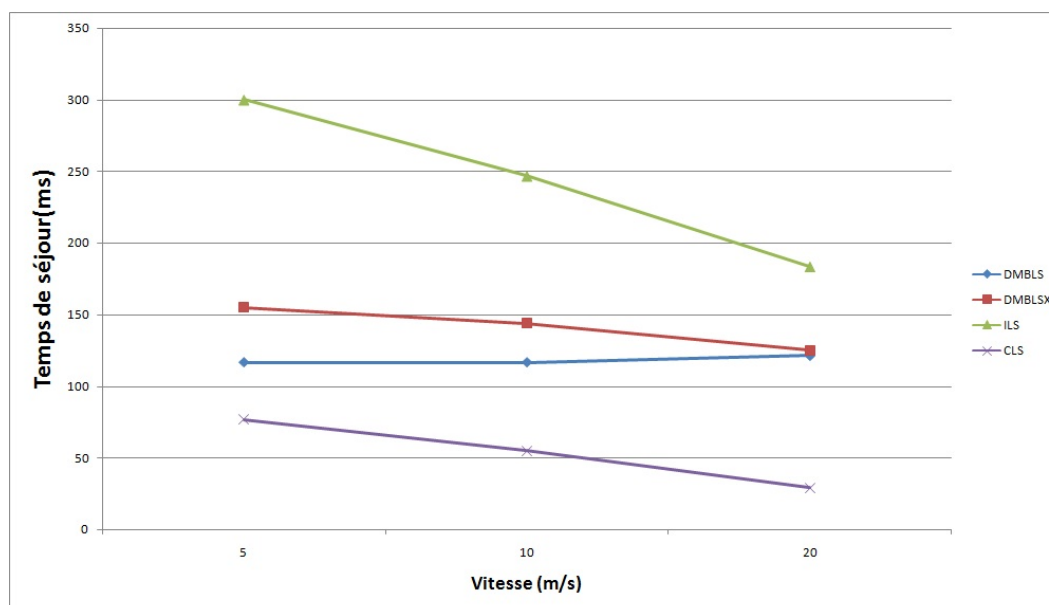


FIGURE 4.7 – TS en fonction de la vitesse.

de ces protocoles, puisque il envoie les mises à jour à tous les serveurs de différents niveaux d'hierarchie à chaque période de mise à jour. Par contre DMBLSX envoie seulement au serveur de niveau le plus bas d'hierarchie (sauf si le nœud traverse la frontière d'une nouvelle région).

4.6 Analyse de résultats

Après l'analyse de ces graphes, les résultats sont absolument inattendus. De manière générale, le protocoles le plus naïf qu'on a proposé (CLS) surpasse

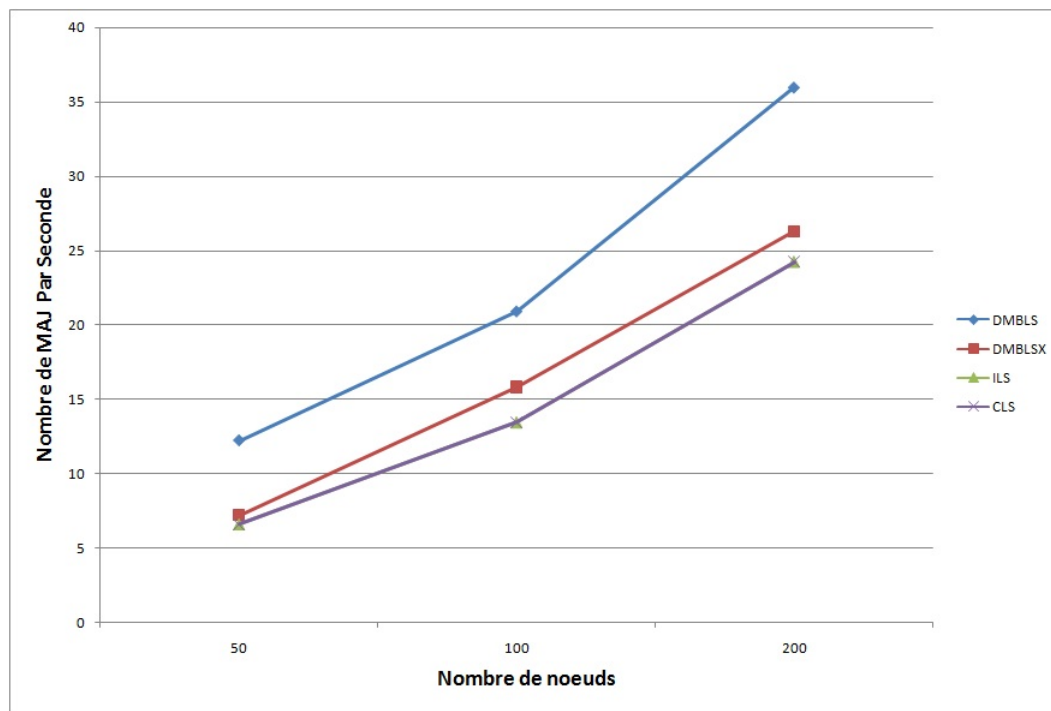


FIGURE 4.8 – Nombre de MAJ en fonction de nombre de nœuds

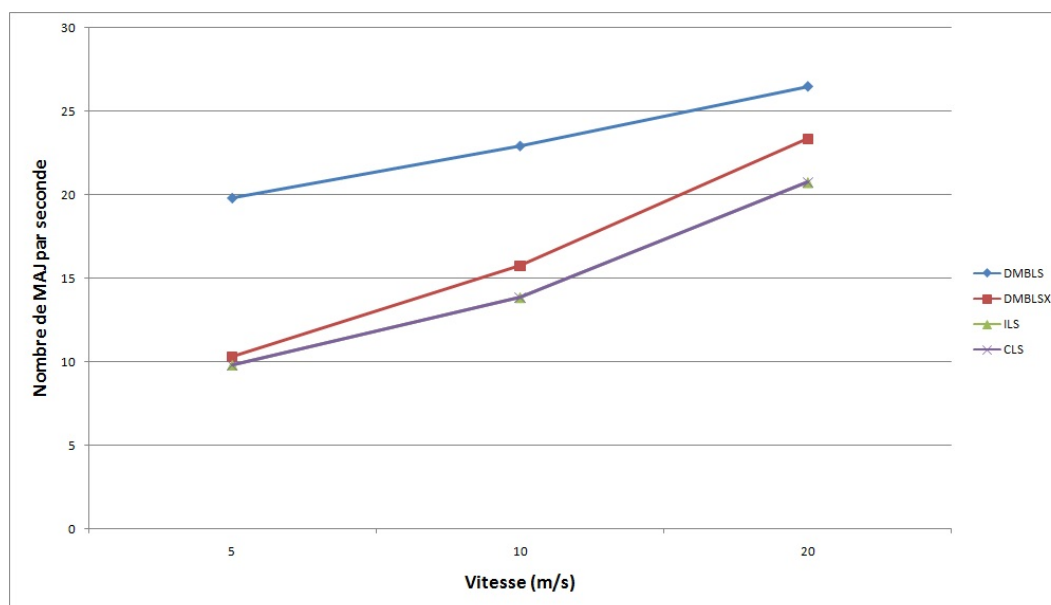


FIGURE 4.9 – Nombre de MAJ en fonction de la vitesse.

tous les autres protocoles, parce que ces protocoles ont été conçus pour un environnement urbain comme Houston ou New York, où la densité des nœuds est distribuée sur toute la carte, ce qui n'est pas le cas dans les villes algériennes, parce que, dans nos villes, il est souvent une seule grande route qui contiennent La partie majeure des nœuds du réseau. Les changements que nous avons fait dans DMBS, donnent une meilleure performance en terme de stockage et surcharge de réseau (le coût de mise à jour est 64 octets), et performe presque comme DMBS dans les autres métriques (TR, EP, TS).

4.7 Conclusion

Dans ce chapitre nous avons présenté et analysé les résultats de simulation de protocoles (DMBLS, DMBSX, ILS, CLS). Nous avons prouvé que le changement que nous avons fait donne une meilleure performance et terme de terme de stockage et surcharge de réseau.

Conclusion générale

Dans ce mémoire , nous avons étudié les protocoles de service de localisation dans les réseaux véhiculaires. Dans le cadre de cette étude, nous avons fait une étude comparative entre plusieurs protocoles et nous avons simulé certain protocoles et nous avons amélioré la performance d'un de ces protocoles.

La simulation a été faite en utilisant ns2, qui était un très mauvais choix, puisque c'est un simulateur très limité, et n'est pas scalable, en effet il ne peut pas simuler plus de quelques centaines de nœuds. Alors nous proposons comme perspective de faire la simulation avec un simulateur performant comme SWANS [20] par exemple, qui peut simuler plus de 1.000.000 nœuds.

Bibliographie