

Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

Zerouali imane

THEME :

**ETUDE DE LA STABILITÉ DES CLUSTERS A K-
SAUTS DANS UN RESEAU VANET**

Soutenu le : 18/09/2022

Devant le jury composé de :

Bentaouza chahinez	MCB	Université de Mostaganem	Président
Laredj Adnan	MCA	Université de Mostaganem	Examineur
Bahnes nacéra	MAA	Université de Mostaganem	Encadreur

Année Universitaire 2021-2022

Résumé

Les réseaux ad hoc de véhicules (VANET) sont un cas particulier des réseaux mobiles Ad-hoc caractérisés par une topologie dynamique due à la mobilité des véhicules. Cependant, le réseau ad hoc de véhicules souffre d'une connectivité discontinue et d'une capacité limitée. La hiérarchisation du VANET organise le réseau en groupes de nœuds (cluster) en désignant un cluster-head pour chaque cluster et apporte des solutions à ces problèmes si l'on tient compte des particularités des VANET.

A la base de l'algorithme de clusterisation (fitnessclustering) qui est fondé sur une fonction appelée *Fitness* (combinaison de plusieurs paramètres ayant un impact direct sur la stabilité des clusters), nous avons proposé une approche pour sélectionner le nœud qui effectue la diffusion des messages d'alerte en cas de collision entre les véhicules (signaler une situation d'urgence). Nous comparons les résultats de simulation de cette approche avec celles de l'algorithme WCA (Weight Clustering Algorithm).

Mots-clés: VANET, clustering, stability, WCA, Fitnessclustering .

Abstract

Vehicular ad hoc networks (VANETs) are a special case of mobile ad hoc networks characterized by a dynamic topology due to vehicle mobility. However, the vehicular ad hoc network suffers from discontinuous connectivity and limited capacity. VANET prioritization organizes the network into groups of nodes (clusters) by designating a cluster head for each cluster and provides solutions to these problems if the particularities of VANETs are considered.

Based on the fitnessclustering algorithm, which is related to a function called Fitness (a combination of several parameters that have a direct impact on the stability of clusters), we proposed an approach to select the node that performs the broadcasting of alert messages in case of collision between vehicles (signaling an emergency situation). We compare the simulation results of this approach with those of the weight-based clustering algorithm WCA (Weight Clustering Algorithm).

Keywords: VANET, clustering, stability , WCA , Fitnessclustering

ملخص

الشبكات المخصصة للمركبة (VANET) هي حالة خاصة من الشبكات المخصصة المتنقلة التي تتميز بطوبولوجيا ديناميكية بسبب تنقل المركبات. ومع ذلك ، فإن الشبكة المخصصة للمركبة تعاني من انقطاع الاتصال والقدرة المحدودة. ينظم التسلسل الهرمي للشبكة VANET في مجموعات من العقد (cluster) من خلال تعيين رأس مجموعة لكل مجموعة ويوفر حلاً لهذه المشكلات إذا أخذنا بعين الاعتبار خصوصيات VANET. استناداً إلى خوارزمية الكتلة العنقودية (مجموعة اللياقة Fitnessclustering) التي تستند إلى وظيفة تسمى اللياقة-Fitness (مجموعة من العديد من المعاملات التي لها تأثير مباشر على استقرار المجموعات) ، فقد اقترحنا نهجاً لتحديد العقدة التي تقوم بنشر رسائل التنبيه في حالة الاصطدام بين المركبات (الإبلاغ عن حالة طوارئ). قارنا نتائج المحاكاة لهذا النهج بنتائج WCA (خوارزمية تجميع الوزن Weight Clustering Algorithm)

كلمات مفتاحية: VANET, clustering, stability , WCA , Fitnessclustering

Dédicaces

*Je dédie ce modeste travail en signe de respect, reconnaissance et de remerciement
: A mes chers parents*

*A mes frères, mes sœurs et ceux qui ont partagé avec moi tous les moments
d'émotion de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé
tout au long de mon parcours.*

A ma famille, mes proches et a ceux qui me donnent de l'amour et vivacité

A mes amis qui m'ont toujours encouragé, et a qui je souhaite plus le succès.

Merci d'être toujours là pour moi.

Imane

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès et qui m'ont aidée lors de la rédaction de ce mémoire.

Nous tiendrons à exprimer toutes nos reconnaissances à notre directrice de mémoire, Madame BAHNES NACERA pour son encadrement ses conseils,

Son écoute, et son soutien moral et ses encouragements. Il m'a guidé dans mon travail et m'a aidé à trouver des solutions pour avancer.

Nous remercions nos très chers parents qui ont toujours été là pour nous.

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Réseau ad-hoc	6
Figure 2	Réseau VANET	8
Figure 3	Les modes de la communication VANET	10
Figure 4	Protocoles de routage dans les réseaux VANETs	12
Figure 5	Exemple of clusters structure	16
Figure 6	Classification des algorithmes de clustérisassions	17
Figure 7	Exemple de structures de l'ensemble des cluster-heads	19
Figure 8	Exemple de maintenance de la structure des clusters en cas d'apparition d'un nouveau lien	25
Figure 9	Exemples de maintenance de la structure des clusters lors de la perte de lien	26
Figure 10	schéma algorithme fitness-clustering	41
Figure 11	Diagramme d'use-case	42
Figure 12	Diagramme de classe	43
Figure 13	Diagramme de séquence	44
Figure 14	Types nœuds	50
Figure 15	Page d'accueil	51
Figure 16	Interface principale de notre simulateur	52
Figure 17	paramétrage de réseau et méthode WCA et fitness clustering	52
Figure 18	distribution des nœuds	53
Figure 19	Simulation des véhicules dans notre réseau	54
Figure 20	Courbe de mobilité relative (fitnessclustering)	54

Figure 21	Courbe de mobilité relative (WCA)	55
Figure 22	Energie consommée	55
Figure 23	Distance entre 2 nœuds	56
Figure 24	Path (chemin) entre 2 nœuds	56
Figure 25	La densité du véhicule	57
Figure 26	Caractéristiques de nœud	57
Figure 27	liste des véhicules au collets	58
Figure 28	La formation et maintenance de (fitnessclustering)	59
Figure 29	La formation et maintenance de (WCA)	60

Liste des tableaux

TableauN°	Titre du tableau	Page
Tableau1	Avantages et inconvénients d'un réseau Ad-hoc	6
Tableau 2	Etats de véhicule	32
Tableau 3	Paramètres de simulation	49

Liste des abréviations

Abréviation	Expression Complète	Page
STI	Systèmes de Transport Intelligents	3
VANET	Vehicular Ad-hoc Network	7
RSU	Road Side Unit	8
AC	Autorité Centrale	8
OBU	On Board Unit	9
V2V	Véhicule to véhicule	9
V2I	Véhicule to infrastructure	9
CH	Cluster-head	16
CM	Cluster member	16
CG	Cluster Gateway	16
IS	Independent Set	18
DS	Dominating Set	18
WCA	Wheight Clusering Algorithme	18

Table des matières

Introduction Générale	4
Chapitre 1 Réseau véhiculaire (VANET)	6
1.1 Introduction	6
1.2 Définitions	6
1.3 Réseau VANET	9
1.3.1 Entités de communication [9]	9
1.3.2 Modes de communication	10
1.3.3 Types de messages	11
1.3.4 Caractéristiques des réseaux véhiculaires	11
1.3.5 Application des réseaux VANET	12
1.3.6 Protocoles de routage dans les réseaux VANET	13
1.4 Conclusion	15
Chapitre 2 Clustering pour les réseaux VANET	16
2.1 Introduction	16
2.2 Définition de Clustering	16
2.3 Classification des algorithmes de clustérisassions	17
2.3.1 Métriques de sélection des Cluster-Head	18
2.3.2 Structure de l'ensemble des cluster-Head	19
2.3.3 Rayon du cluster	20
2.3.4 Métriques d'évaluation des performances des algorithmes de clustérisassions : ...	20
2.4 Les algorithmes de clustérisassions à 1 saut	22
2.4.1 Les algorithmes à métrique arbitraire	22
2.4.2 Les algorithmes à métrique liée à la topologie	22
2.4.3 Les algorithmes à métrique spécifique au nœud mobile	22
2.4.4 Les algorithmes à métriques combinées	23
2.4.5 Les algorithmes sans métrique	23

2.5	Les algorithmes de clustérisations à k-sauts	23
2.5.1	Les algorithmes à métrique arbitraire	23
2.5.2	Les algorithmes à métrique liée à la topologie	23
2.5.3	Les algorithmes à métrique spécifique au nœud mobile.....	24
2.5.4	Les algorithmes à métriques combinées	24
2.5.5	Les algorithmes sans métrique.....	24
2.6	La différence entre les algorithmes de clustrering a seul saut et a multi -sauts (k-saut) :.....	25
2.7	Stabilité :	25
2.8	Maintenance des clusters :.....	25
2.9	Etat de l'art.....	28
2.10	Conclusion	28
Chapitre 3 Contribution et conception.....		29
3.1	Introduction	29
3.2	Principe modèle proposé :.....	29
3.3	Les algorithmes :.....	31
3.3.1	Algorithme WCA.....	31
3.3.2	Algorithme Fitnessclustering :.....	34
3.4	Conception et modélisation :.....	40
3.4.1	Méthode de conception :.....	40
3.5	Les diagrammes UML :.....	40
3.5.1	Diagramme de use case :.....	40
3.5.2	Diagramme de classe :.....	41
3.5.3	Diagramme de séquence :.....	42
3.6	Conclusion :.....	42
Chapitre 4.....		43
4.1	Introduction :.....	43
4.2	Notre application (simulateur):	43
4.3	Simulation :.....	43

4.3.1	Environnement de travail :	44
4.4	Les résultats de simulation :	45
4.4.1	Approches implémentés	45
4.4.2	Les parametres de simulation :	46
4.4.3	Scénario :	46
4.5	L'implémentation :	47
4.6	Analyse des performances des algorithmes et comparaison des résultats :	55
4.7	Comparaison ente algorithme fitnessclustering et WCA :	57
4.8	Conclusion :	57
	Conclusion générale	59
	Bibliographie	60

Introduction Générale

Ces dernières années, le domaine des télécommunications sans fil a connu une évolution importante. Cette évolution est due essentiellement aux besoins actuels en termes de disponibilité et d'accès aux données à n'importe quel moment et depuis n'importe quel endroit. De nombreuses applications est développées afin d'améliorer notre vie quotidienne dans nos maisons, nos sociétés, nos voitures...etc.

Les systèmes de transport intelligents (STI) (en anglais : Intelligent Transportation System (ITS)) désignent les applications des nouvelles technologies de l'information et de la communication au domaine de transport.

Un réseau VANET (Vehicular Ad-hoc Network) est un élément clé des systèmes de transport intelligent. VANET est un réseau d'échanges de messages entre plusieurs nœuds mobiles (Voitures, Motocycles, Camions ou autres) afin de gérer et organiser les informations reçues des nœuds mobiles qui circulent sur les routes (rurales ou urbains) dont l'objectif est d'éviter de potentiels accidents ou congestions où assurer d'autre fonctionnalités. Actuellement, le réseau VANET est devenu un champ de recherche important en raison des perspectives d'application larges, comme le partage de l'information de la circulation, aide à la conduite, l'accès à l'Internet ...etc.

L'organisation des nœuds de réseau en clusters doit avoir la capacité de maintenir la formation de clusters existants après chaque changement topologique. La stabilité représente le défi essentiel pour les procédures de clustering dans des environnements dynamiques.

L'objectif de ce projet est d'étudier les techniques de formations des clusters et leur stabilité dans un réseau VANET. Ce rapport est organisé en 4 chapitres :

Dans le premier chapitre, nous allons présenter les réseaux véhiculaires (VANETs), leurs caractéristiques, les différents types de messages, les applications et les protocoles de routage dans ces réseaux.

Dans le deuxième chapitre, nous allons citer quelques algorithmes de formation et de maintenance des clusters dans les réseaux VANETs.

Dans le troisième chapitre, nous allons présenter la partie de contribution et la partie de conception.

Dans le quatrième chapitre, nous allons présenter la partie implémentation et simulation, plus les résultats d'évaluation de notre proposition.

Enfin, nous terminerons ce manuscrit par une conclusion générale dans laquelle nous allons résumer les résultats de notre travail et citer quelques perspectives.

Chapitre 1 Réseau véhiculaire (VANET)

1.1 Introduction

Les réseaux véhiculaires (VANET : Vehicular Ad-hoc NETwork) est un réseau Ad-hoc mobile pour fournir des communications : V2V (entre les véhicules voisins) et V2I (entre les véhicules et RSU (Road Side Unit)), habituellement décrit comme équipement de bord de la route. Le but des applications avec communication véhiculaire est d'améliorer la sécurité routière, réduire les collisions, et minimiser les dégâts en cas d'accident.

Dans ce chapitre, nous présentons quelques définitions, l'architecture de réseau VANET, les entités de communication et les modes de communication existants, et nous décrivons aussi les différentes caractéristiques des réseaux véhiculaires, les applications des réseaux VANET et les protocoles de routage.

1.2 Définitions

1) Réseaux ad-hoc [1]:

Un réseau sans fil Ad-Hoc est un type de réseau qui se caractérise par une communication point à point et sans infrastructure (point d'accès). La figure 1 schématise le mode d'échange en réseau Ad-hoc.

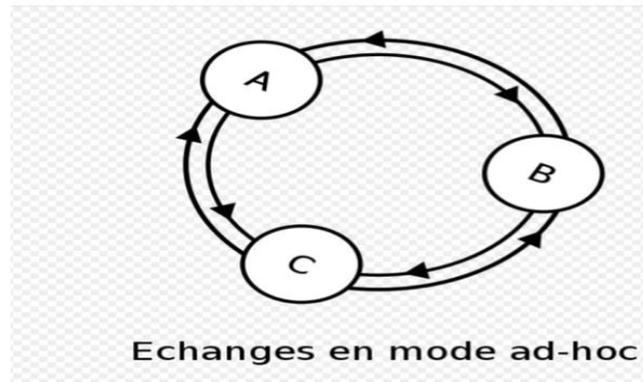


Figure 1 – Réseau Ad-hoc [2]

Le tableau suivant présente quelques avantages et inconvénients d'un réseau de type Ad-hoc.

Les avantages	Les inconvénients
<ul style="list-style-type: none"> • Les réseaux ad-hoc peuvent être déployés dans un environnement quelconque. • Le coût d'exploitation du réseau est faible : aucune infrastructure n'est à mettre en place initialement et surtout aucun entretien n'est à prévoir. • Le déploiement d'un réseau Ad-hoc est simple. 	<ul style="list-style-type: none"> • Les liens entre les stations ne sont pas isolés les uns des autres et polluent le voisinage, par diffusion, lors de chaque émission/réception de données • La sécurité dans les réseaux Ad-hoc est difficile.

Tableau1 : Avantages et inconvénients de réseau Ad-hoc [3]

2) Réseaux MANET :

Le réseau mobile Ad-hoc, appelé généralement MANET (Mobile Ad-hoc Network) est un système autonome se compose des nœuds mobiles dynamiques interconnecté par des liens sans fil sans l'utilisation de l'infrastructure fixe et sans gestion centralisée [16].

Les caractéristiques principales d'un réseau MANET sont :

- **Mobilité:** La mobilité des nœuds constitue à l'évidence une caractéristique très spécifique des réseaux Ad-hoc. Cette mobilité est intrinsèque au fonctionnement du réseau.

- **Équivalence des nœuds du réseau** : Dans un réseau classique, il existe une distinction entre les nœuds terminaux (stations, hôtes) qui supportent les applications et les nœuds internes (routeurs par exemple) du réseau, en charge de l'acheminement des données. Cette différence n'existe pas dans les réseaux Ad-hoc car tous les nœuds peuvent être amenés à assurer des fonctions de routage.
- **Liaisons sans fil** : Les technologies de communication sans fil sont indispensables à la mise en place d'un réseau Ad-hoc.
- **Capacité énergétique**: La consommation d'énergie constitue un problème important pour des équipements fonctionnant grâce à une alimentation électrique autonome. Ces équipements intègrent des modes de gestion d'énergie et il est important que les protocoles mis en place dans les réseaux ad-hoc prennent en compte ce problème.
- **La sécurité** : sécuriser les réseaux Ad-hoc est une tâche particulièrement difficile à cause de [5] :
 - ➔ Vulnérabilité des canaux: les messages peuvent être écoutés ou injectés dans le réseau.
 - ➔ La vulnérabilité des nœuds : les nœuds du réseau sont facilement accessibles par les intrus.
 - ➔ Le manque d'infrastructure: les solutions classiques basées sur la présence d'autorité de certification ou de serveurs centraux sont inapplicables ou difficilement applicables.
 - ➔ Le changement de topologie: La mobilité des nœuds entraîne un changement fréquent de topologie qui résulte en un changement de route, des partitionnements fréquents et des pertes de paquets possibles [6].

3) Réseau VANET

C'est un élément clé des systèmes de transport intelligent dans lequel les messages sont échangés entre les nœuds (mobiles ou fixes) pour la mise en place des applications de transport liées aux conducteurs et à la sûreté des routes (voir la figure 2). Ces dernières années, le VANET est devenu un champ de recherche intéressant surtout avec l'arrivée des

nouvelles applications, comme le partage de l'information de la circulation, aide à la conduite, l'accès à l'Internet...etc. [7].

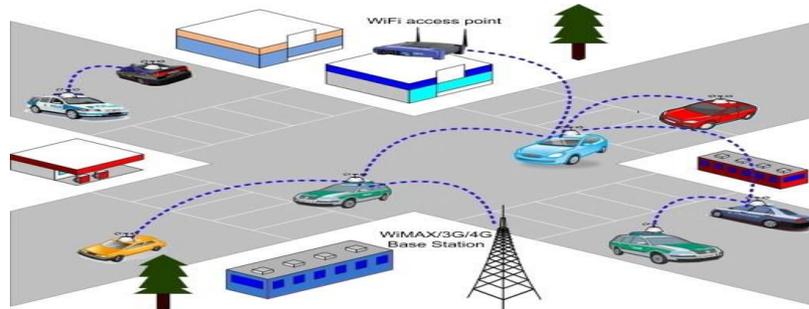


Figure 2 : Réseau VANET [8]

1.3 Réseau VANET

1.3.1 Entités de communication [9]

1) Road Side Unit

Les Road Side Unit (RSU) sont des entités situées et installées au bord de la route. Ces entités présentent des points d'accès ou de collecte d'informations. Ils sont déployés tout au long de la route. Chaque RSU a pour objectif de transmettre des messages aux véhicules qui se trouvent dans sa zone radio. Ces messages contiennent des informations sur les conditions météorologiques, ainsi que sur l'état de la route (vitesse maximale, autorisation de dépassement, etc.).

2) Autorité centrale

L'Autorité Centrale (AC) est un serveur de stockage et de transaction qui a la confiance de toutes les entités du réseau. Elle fournit des services et des applications à tous les utilisateurs, ainsi que les certificats, les clés ou pseudonymes de communication des véhicules.

3) On Board Unit

L'On Board Unit (OBU) est une unité embarquée dans les véhicules intelligents. Son rôle est de permettre aux véhicules de se localiser, calculer, enregistrer et envoyer des messages sur

une interface réseau. Dans le réseau VANET, le conducteur peut voir les pseudonymes des véhicules à proximité dans son OBU à l'aide des messages envoyés périodiquement, appelés « Beacon ». Ainsi, l'utilisateur peut choisir le véhicule avec lequel il veut communiquer [9].

1.3.2 Modes de communication

Dans les réseaux véhiculaires, il existe trois modes de communication (Figure 3): les communications Véhicule-à-Véhicule (V2V), les communications Véhicule-à-Infrastructure (V2I) et les communications hybride.

- **Communication en mode Ad-hoc (V2V) :** c'est une forme de communication permettant aux véhicules de communiquer directement entre eux. Elle ne nécessite aucune infrastructure. Cette communication est très efficace pour le transfert des informations concernant les services liés à la sécurité routière. En revanche, elle ne garantit pas une connectivité permanente entre les véhicules [10] [11].
- **Communication en mode infrastructure (V2I) :** c'est une communication réalisée entre les nœuds (Véhicules) et les entités fixes (RSU et CA). Elle permet aux véhicules d'accéder aux différents types d'applications (Sécurité, confort, gestion ...) et aux différents types d'informations (Etat du trafic, météo ...). Ce mode est inadéquat pour les applications liées à la sécurité routière car les réseaux à infrastructure ne sont pas performants qu'aux délais d'acheminement [12].
- **Communication hybride :** c'est une combinaison de deux types de communications (V2V et V2I) qui permet d'obtenir une communication hybride très intéressante. En effet, les portées des infrastructures (stations de base) étant limitées, mais avec l'utilisation des véhicules comme relais, ce problème peut être résolu en étendant cette distance [13].

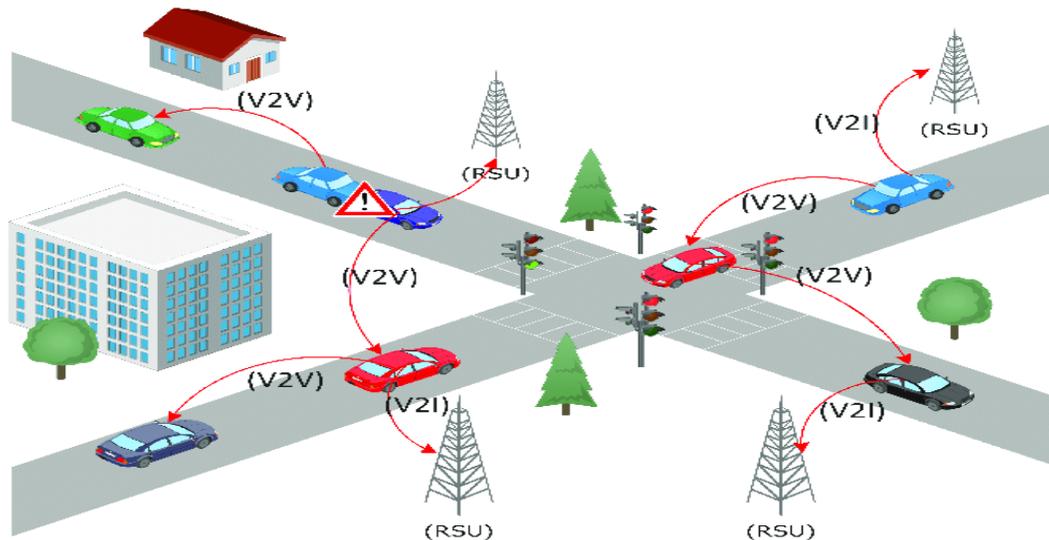


Figure 3 : Les modes de la communication VANET [14].

1.3.3 Types de messages

Message de contrôle : Les messages de contrôle sont des messages périodiques de type «Beacon ». Chaque véhicule émet un message de contrôle. Ce message contient des informations sur le véhicule tel que sa position, sa vitesse, sa direction. Ils donnent une vision locale aux véhicules voisins afin de prévoir la survenance d'accidents catastrophiques.

Message alerte : Ce type de message est envoyé lorsqu'un accident ou une situation dangereuse se produit pour avertir les autres véhicules (conducteurs) de diverses urgences et catastrophes (accident, embouteillage, informations météorologiques,.. etc.).

1.3.4 Caractéristiques des réseaux véhiculaires

Les réseaux véhiculaires ont des caractéristiques spécifiques qui les distinguent de réseaux Ad-hoc mobiles (MANET). Ces caractéristiques doivent être prises en compte lors de conception des protocoles pour les VANETs. Dans cette partie, nous présentons quelques propriétés et contraintes concernant ce type de réseau.

- **La capacité d'énergie et de stockage :** Contrairement au contexte des réseaux MANET où la contrainte d'énergie représente un défi pour les chercheurs, les éléments du réseau VANET

disposent suffisamment d'énergie qui peut alimenter les différents équipements électroniques d'une voiture intelligente (ou connecté). Alors, les nœuds sont censés avoir une grande capacité de traitement et de stockage de données [15].

- **La topologie et la connectivité :** Comme les réseaux Ad-hoc mobiles, les réseaux VANET sont caractérisés par une connectivité sporadique, car un véhicule peut rejoindre ou quitter un groupe de véhicules en un temps très court, ce qui nous mène ainsi à avoir une topologie très dynamique constituée de plusieurs groupes séparés [17][18].
- **Le modèle de mobilité :** Le modèle de mobilité des réseaux VANETs est lié à la diversité environnementale et les infrastructures routières [19].
- **La sécurité et l'anonymat :** L'importance des informations échangées via les communications véhiculaires rend l'opération de sécurisation de ces réseaux cruciale qui constitue un prérequis au déploiement des VANETs [20].

1.3.5 Application des réseaux VANET

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

- **Applications de la sécurité routière :** les véhicules peuvent s'échanger entre eux des messages d'alerte afin de prévenir les collisions, les travaux sur les routes et de détecter les obstacles. Par la suite, les interventions deviennent plus rapides et le nombre des accidents sera réduit [21].
- **Applications pour l'optimisation du trafic et aide à la conduite :** Les applications de gestion de trafic sont axées sur l'amélioration des conditions de circulation dont le but est de réduire les embouteillages et les risques d'accidents. Elles fournissent aux conducteurs un support technique leur permettant d'adapter leur parcours à la situation du trafic routier. Ces applications visent à équilibrer la circulation des véhicules sur les routes pour une utilisation efficace de la capacité des routes et des carrefours et à réduire par conséquent les pertes humaines, d'énergie, etc.
- **Applications de confort du conducteur et des passagers :** Les réseaux VANET ne se contenteront pas seulement à offrir des services liés à la sécurité des véhicules et leurs

occupants mais permettront aussi d'assurer le confort de ces derniers durant leurs voyages. Ces services comprennent entre autres la messagerie instantanée, les jeux en réseau, l'accès à Internet, les paiements automatiques et la diffusion des informations utiles sur la disponibilité de l'espace de stationnement dans les parkings en indiquant aux conducteurs les espaces libres [22].

1.3.6 Protocoles de routage dans les réseaux VANET

Le routage joue un rôle très important dans les VANETs puisque tous les services supportés, *unicast* ou *multicast*, se basent sur des communications multi-saut pour l'acheminement des données et les transferts de messages.

Le problème de routage dans les réseaux véhiculaires réside essentiellement dans l'instabilité des chemins causée par la forte mobilité des nœuds et les fragmentations fréquentes du réseau. Les protocoles dédiés aux VANETs peuvent être classés en deux classes : des protocoles basés sur la topologie et des protocoles basés sur la géographique (voir la figure 4). Les protocoles basés sur la topologie sont réactifs, proactifs, hybride. Ils utilisent les informations sur les liens qui existent entre les nœuds pour l'acheminement des paquets.

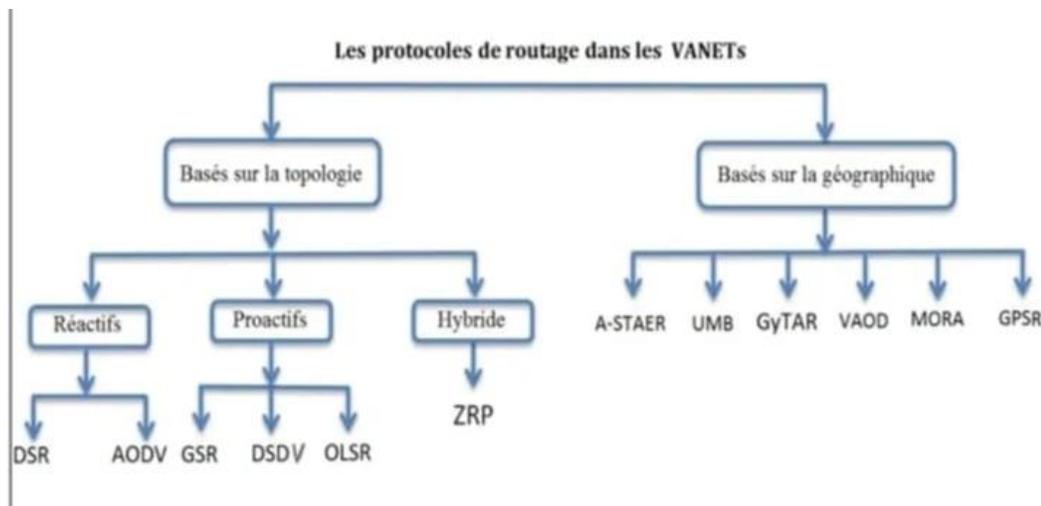


Figure4 : Protocoles de routage dans le réseau VANET [23]

- **Les protocoles proactifs**

Les protocoles proactifs dits « à *diffusion de table* » se caractérisent par l'établissement de route à l'avance, de façon que lorsqu'un nœud désire envoyer un paquet à un autre nœud, une route soit immédiatement connue, chaque nœud maintient une ou plusieurs tables qui contiennent des informations de routage. Cette catégorie de protocole requiert un échange périodique de messages de contrôle afin de mettre à jour les tables de routage dans chaque nœud. Ainsi, de nouvelles routes seront construites à partir des informations transportées par les trames de contrôle. Ce processus est déclenché à chaque changement de topologie pour reconstruire à nouveau les routes vers toutes les destinations possibles. Les protocoles basés sur ce principe sont : DSDV[24] et OLSR[25]. Un des avantages de ces protocoles est la disponibilité immédiate de la route lors du besoin. Cependant, la bande passante diminue à cause du trafic généré par l'échange de paquets de contrôle.

- **Les protocoles réactifs**

Le principe des protocoles réactifs également appelés « *protocoles de routage à la demande* » (On-Demand routing protocols). Le processus de recherche de routes est lancé uniquement en cas de besoin (à la demande). Ces protocoles se basent sur la découverte et le maintien des routes. Suite à un besoin, une procédure de découverte globale de route est lancée [25], en cherchant un chemin jusqu'à la destination, une fois ce chemin est trouvé, il est inscrit dans la table de routage et peut être utilisé tant que la destination est joignable ou jusqu'au moment où la route devient inutile. Ce type de routage minimise l'échange de messages de contrôle ce qui libère la bande passante « bande passante plus large » mais le délai d'établissement de la route est plus important en comparaison avec les protocoles proactifs. Actuellement le plus connu de ces protocoles est : DSR[26], AODV[27] et TORA[28].

- **Les protocoles hybrides**

Les protocoles hybrides combinent les deux approches précédentes. Pour bénéficier de leurs avantages, ils utilisent un protocole proactif pour connaître les voisins les plus proches dans le but de réduire le délai et un protocole réactif dans le but de réduire la charge des paquets de contrôles. Ces protocoles cumulent aussi les inconvénients des protocoles proactifs et réactifs à savoir, les paquets de contrôles périodiques et le délai de découvertes de routes [29]. Le protocole ZRP est parmi les protocoles hybrides les plus connus [30].

1.4 Conclusion

Dans ce chapitre, nous avons présenté les réseaux VANETs. Ce type de réseau consiste des communications V2V, V2I et hybride. Cette innovation dans la communication sans fil a été exploitée pour améliorer la sécurité routière dans plusieurs situations.

Dans le prochain chapitre, nous présenterons le mécanisme de « CLUSTERING » pour les réseaux VANET qui consiste à organiser le réseau en groupes de nœuds (**cluster**) et apporte alors des solutions à ces problèmes si l'on tient compte des particularités de **VANET**.

Chapitre 2 Clustering pour les réseaux VANET

2.1 Introduction

Ces dernières années, le Clustering est devenu l'un des mécanismes de contrôle les plus utilisés dans les VANET afin de surmonter leurs différents défis et d'améliorer leurs performances. Il est un concept important dans le réseau véhiculaire où l'ensemble des véhicules forme des groupes en se basant sur des critères prédéfinis ou des caractéristiques communes.

La clustérisation consiste à diviser le réseau en groupes de nœuds géographiquement proches. Elle présente une solution intéressante pour simplifier et optimiser les fonctions et les services du réseau.

2.2 Définition de Clustering

Bali & al [31] définissent le regroupement comme un mécanisme de regrouper les véhicules en petits groupes appelés « clusters » en fonction de métriques prédéfinies, telles que la densité de nœuds, la vitesse moyenne et l'emplacement du nœud. Ce regroupement est effectué en fonction des exigences de l'application cible pour fournir un réseau gérable.

Le clustering est un processus qui vise à regrouper les nœuds d'un réseau en un ensemble de groupes, appelés clusters, fournissant ainsi le réseau avec une structure hiérarchique. Cette structure répond à certaines exigences, telles que l'évolutivité, l'équilibrage de charge, la stabilité du réseau et la qualité de service (QoS). En règle générale, les nœuds géographiquement voisins ont une forte probabilité de rejoindre le même cluster selon

certaines règles et métriques. La structure du cluster comporte trois principaux types de nœuds (véhicules) : un chef de cluster (CH : Cluser Head), un membre de cluster (CM : Cluster Member) et une passerelle de cluster (CG : Cluster Gateway) (voir la figure 5). De plus, quelques études définissent d'autres types spéciaux, tels qu'un CH temporaire (TCH), un CM temporaire (TCM) et un CH secondaire(SCH) [31].

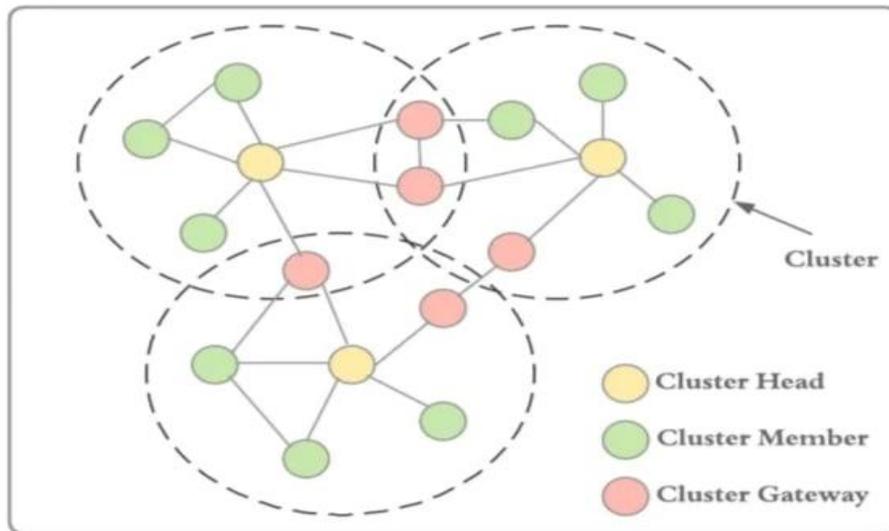


Figure 5 : Structure d'un cluster [32]

2.3 Classification des algorithmes de clustérisassions

La figure 6 illustre la classification des algorithmes de clustérisassions qui contient des catégories d'algorithmes: des algorithmes se basant sur la construction d'un ensemble dominant, des algorithmes à charge équilibrée, des algorithmes à faible coût de maintenance, des algorithmes basés sur la mobilité, des algorithmes d'économie d'énergie et des algorithmes à métriques combinées [33].



Figure 6 : Classification des algorithmes de clustérisassions [34].

2.3.1 Métriques de sélection des Cluster-Head

Plusieurs métriques ont été proposées dans la littérature pour désigner l'ensemble des cluster-Head. Suivant les métriques de sélection, les algorithmes de clustérisassions sont classés en cinq sous classes [34] :

- **Sans métrique:** elle regroupe les algorithmes qui déclarent les cluster-Head sans avoir recours à aucune métrique de sélection.

- **Métrique arbitraire:** elle représente une valeur choisie arbitrairement et généralement non significative. Dans cette classe, nous trouvons les algorithmes qui utilisent des valeurs aléatoires ou l'identifiant des nœuds.
- **Métriques liées à la topologie:** elle regroupe tous les algorithmes qui utilisent une métrique issue de la topologie du réseau. Parmi les métriques qui appartiennent à cette classe, nous citons le degré de connectivité, le k-degré de connectivité et la k-densité.
- **Métriques spécifiques au nœud mobile:** les algorithmes de cette classe utilisent des métriques très spécifiques aux nœuds mobiles. Dans le cas général, ces algorithmes sont basés sur certaines informations de chaque nœud tel que sa mobilité, la puissance des signaux reçus, le niveau d'énergie ou encore sa position géographique. Dans cette classe, nous avons distingué trois sous classes qui sont :
 1. Mobilité requise : elle regroupe les algorithmes qui supposent que les nœuds ont une connaissance de leur mobilité.
 2. Niveau d'énergie requis : elle regroupe tous les algorithmes qui utilisent la valeur de l'énergie restante dans les batteries des nœuds comme métrique de sélection des cluster-Head.
 3. Position géographique requise: cette classe d'algorithmes suppose que chaque nœud a la capacité de déterminer sa position géographique par exemple à travers un système GPS.
- **Métriques combinées:** plusieurs algorithmes combinent plusieurs métriques de différents types pour sélectionner l'ensemble de cluster-Head.

2.3.2 Structure de l'ensemble des cluster-Head

Les cluster-Heads élus forment des structures particulières parmi lesquelles nous citons [34]:

- **DS (Dominating Set)** : ensemble dominant ;
- **CDS (Connected Dominating Set)** : ensemble dominant connecté ;
- **WCDS (Weakly Connected Dominating Set)** : ensemble dominant faiblement connecté;
- **IS (Independant Set)** : ensemble indépendant ;
- **MIS (Maximal Independant Set)**: ensemble indépendant maximal.

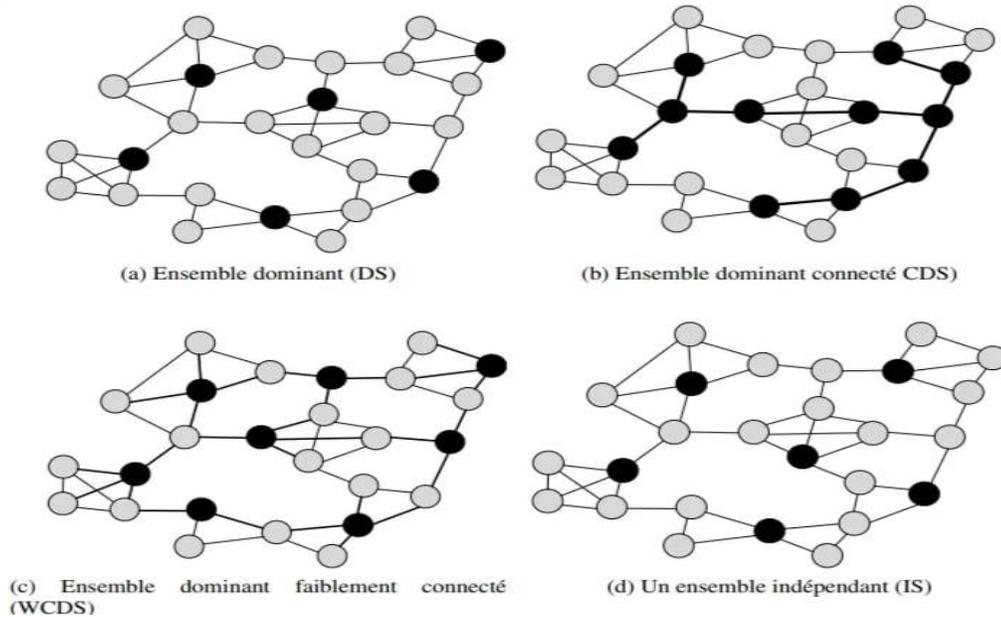


Figure 7 – Exemple de structures de l'ensemble des cluster-heads

2.3.3 Rayon du cluster

Le rayon du cluster est la valeur maximale de la distance séparant la tête du cluster de tous ses membres, et cette distance est exprimée en nombre de sauts. Suivant le rayon de cluster, trois classes d'algorithmes sont définies: les algorithmes à 1-saut, les algorithmes à k-saut et les algorithmes à rayon variable.

2.3.4 Métriques d'évaluation des performances des algorithmes de clustérisations :

Nous avons déduit les groupes de métriques suivants :

➤ *Messages explicites du processus de clustérisations*

- ➔ *Overhead* : l'overhead encouru durant le temps d'opération du réseau.
- ➔ *Complexité en message pour un changement de topologie* : le nombre de messages échangés entre les nœuds pour accomplir une réorganisation valide des clusters après un changement dans la topologie du réseau.

➤ ***Caractéristiques des clusters***

- ➔ *Taille moyenne d'un cluster* : le nombre moyen de nœuds gérés par un cluster-head.
- ➔ *Le nombre moyen de clusters* : le nombre moyen de clusters formés dans le réseau.
- ➔ *Distance totale aux cluster-heads* : la somme des distances entre les nœuds et leurs cluster-heads.

➤ ***Stabilité de la structure des clusters***

- ➔ *Durée moyenne de vie d'un cluster-head* : la durée moyenne pendant laquelle un nœud joue le rôle de cluster-head.
- ➔ *Durée moyenne de vie d'un membre* : la durée moyenne pendant laquelle un nœud est associé à son cluster-head.
- ➔ *Nombre de changements de cluster-head* : le nombre de fois qu'un cluster-head change de statut.
- ➔ *Nombre de mise à jour de l'ensemble dominant* : l'ensemble des cluster-heads est appelé un ensemble dominant si tous les nœuds dans le réseau sont dans cet ensemble ou sont voisins des nœuds de cet ensemble.
- ➔ *Nombre de ré-affiliation* : le compteur de ré-affiliation est incrémenté quand un nœud est dissocié de son cluster-head et se rattache comme membre à un autre cluster-head et à l'intérieur de l'ensemble dominant courant.

➤ ***Facteur d'équilibre de charge***

- ➔ *Durée moyenne de vie d'un cluster-head* : la durée moyenne pendant laquelle un nœud joue le rôle de cluster-head.
- ➔ *Variance de la durée de vie d'un cluster-head* : la variance de la durée pendant laquelle un nœud joue le rôle de cluster-head.

2.4 Les algorithmes de clustérisations à 1 saut

2.4.1 Les algorithmes à métrique arbitraire

L'un des algorithmes de clustérisations les plus anciens est LCA (Linked Cluster Algorithm) [35], connu sous le nom de l'algorithme du plus petit identifiant (ID) ou du plus petit ID. Chaque nœud se déclare cluster-head ou non en se basant sur son identifiant et ceux de ses voisins.

2.4.2 Les algorithmes à métrique liée à la topologie

L'algorithme HCC (Highest Connectivity Cluster) [34] choisit le nœud ayant le plus grand nombre de voisins (plus grand degré) comme cluster-head. Initialement, tous les nœuds sont non couverts. Chaque nœud diffuse la liste des nœuds qu'il peut détecter. La phase de formation des clusters applique les règles suivantes :

1. Un nœud est élu cluster-head s'il a le plus grand nombre de voisins. Dans le cas d'égalité, le nœud ayant le plus petit identifiant ID deviendra cluster-head.
2. Un nœud qui n'a pas encore choisi son cluster-head demeure dans son état non couvert, sinon il devient un nœud couvert.
3. Un cluster-head qui a élu un autre nœud comme son cluster-head, abandonne son rôle de cluster-head.
4. Les nœuds appartenant à plusieurs clusters jouent le rôle de passerelle.

2.4.3 Les algorithmes à métrique spécifique au nœud mobile

MOBIC (Lowest Relative Mobility Clustering Algorithm) [36] est un algorithme distribué qui applique le même algorithme que LCA et HCC mais implique une autre métrique qui est la mobilité relative des nœuds.

2.4.4 Les algorithmes à métriques combinées

Ces algorithmes de clustérisations utilisent une fonction qui combine plusieurs métriques. Cette fonction représente le poids d'un nœud. Cette catégorie d'algorithmes vise à élire un ensemble de cluster-heads qui sera le plus adapté à une topologie pour satisfaire un ou plusieurs besoins. Par exemple, dans un réseau de capteurs où l'énergie est un facteur important, l'énergie peut avoir un poids plus important dans la fonction combinée de la métrique résultante.

2.4.5 Les algorithmes sans métrique

Les algorithmes classiques de clustérisations requièrent que tous les nœuds envoient des messages explicites pour former et maintenir la structure des clusters ce qui laisse la clustérisation une des principales sources d'over-head.

2.5 Les algorithmes de clustérisations à k-sauts

2.5.1 Les algorithmes à métrique arbitraire

Chaque nœud doit connaître ses voisins à k-sauts. Tout nœud non affilié possédant le plus petit ID parmi ses k-voisins non encore affiliés devient cluster-head [37].

2.5.2 Les algorithmes à métrique liée à la topologie

Dans [38], les auteurs introduisent une métrique qu'ils appellent associativité qui représente la stabilité relative des nœuds dans leur voisinage. Pour chaque nœud, l'associativité comptabilise le temps que chacun des nœuds de son voisinage reste effectivement dans son voisinage et en fait la somme sur chaque voisin.

A chaque période de temps, un nœud u considère quels sont ses voisins actuels déjà présents lors de la période précédente et ajoute **1** à la valeur associée à chacun d'eux.

Si un voisin apparaît, il prend la valeur **1** et si un voisin a disparu, la valeur qui lui était associée prend **0**. A chaque période de temps, l'associativité de u est la somme des valeurs

associées à chacun de ses voisins. Cette valeur prend donc en compte la stabilité de u (si u est relativement stable dans son voisinage, il aura une forte associativité) et le degré des nœuds. La formation des clusters suit le procédé suivant: un nœud considère les nœuds de k -voisinage ayant un degré supérieur à une valeur seuil et élit parmi eux celui ayant la plus forte associativité.

2.5.3 Les algorithmes à métrique spécifique au nœud mobile

HEED[55] est un algorithme distribué de clustérisation proposé pour les réseaux de capteurs. HEED construit un graphe connecté de cluster-heads à plusieurs sauts. Les clusters formés sont disjoints. Les cluster-heads sont élus en se basant sur l'énergie résiduelle de chaque nœud.

2.5.4 Les algorithmes à métriques combinées

Mob-Dhop (Mobility-based D-hop) [37] est un algorithme distribué qui forme des clusters de diamètre variable avec la mobilité du nœud en supposant que chaque nœud peut mesurer la puissance du signal reçu.

2.5.5 Les algorithmes sans métrique

Lin et Chu [38] ont proposé un autre algorithme pour générer des k -clusters qui n'utilise aucune métrique pour le choix des cluster-heads et s'exécute comme suit :

- A l'arrivée d'un nœud, il passe à l'état "Initialisation". Puis, il demande l'état de ses voisins : sont-ils à l'état "Initialisation" ? ou sont-ils affiliés à des clusters ?, et dans ce dernier cas, à quelle distance se situe leur cluster-head correspondant ?
- Si tous les voisins d'un nœud sont à l'état "Initialisation", ce nœud se déclarera comme cluster-head et diffusera sa décision dans son voisinage. Tous les k -voisins de ce nœud qui n'est pas attachés à aucun cluster plus proche, s'attachent à ce nouveau cluster.
- Sinon, ce nœud joint le cluster de son voisin dont le cluster-head est le plus proche et qui se trouve à au plus k sauts de lui.

- Si tous les cluster-heads de ses voisins sont à k-sauts de ce nœud, ce dernier se déclarera cluster-head et invitera ses k-voisins qui ont des cluster-heads plus éloignés à le rejoindre.

2.6 La différence entre les algorithmes de clustering à seul saut et à multi -sauts (k-saut) :

La grande différence entre les algorithmes de clustering à un seul saut et à multi-sauts est la collecte et la diffusion d'informations entre les nœuds. La difficulté est d'organiser le réseau de manière distribuée et en se basant le plus possible sur des informations locales uniquement.

2.7 Stabilité :

Lorsque la structure de clustering subit des changements (comme variation du nombre de clusters et/ou l'évolution de l'adhésion des nœuds aux clusters), un compromis entre ces changements et la stabilité des clusters doit être établi afin de conserver la structure aussi stable que possible. Autrement, elle est supposée fixe telle que chaque nœud soit propre à son cluster ainsi que le nombre de clusters dans le réseau soit fixe tout au long de sa durée de vie [4].

2.8 Maintenance des clusters

Définition :

Dans de nombreux algorithmes, la maintenance est effectuée par re-clustering. D'autres algorithmes proposent d'éliminer l'effet de cascade introduit par le re-clustering. Ils proposent quelques règles et mesures de maintenance qui fonctionnent localement et limitent le diamètre de la modification de la structure du cluster.

Le processus de maintenance des clusters essaye d'adapter la structure des clusters à tous les changements de la topologie qui peuvent avoir lieu. Dans α -SSCA[52], la maintenance des clusters est inspirée de l'algorithme LCC[51]. Elle a comme objectif d'améliorer la stabilité de la structure des clusters.

La figure 8 présente un exemple de maintenance des clusters lorsque deux cluster-heads se trouvent dans la même portée de transmission. Elle montre l'effet de l'abandon d'un cluster-head sur la stabilité de la structure des clusters. En effet, quand les deux cluster-heads (1 et 6) deviennent voisins, les algorithmes LCC-hc et 3HBAC imposent que le cluster-head numéro 1 avec le plus faible degré de connectivité abandonne son rôle. Cette figure montre aussi la nouvelle structure des clusters maintenue après le renoncement du cluster-head « 1 ». En effet, cet action de maintenance va causer la suppression d'un cluster et l'apparition de trois nouveaux clusters ce qui entraîne au total quatre changements d'état.

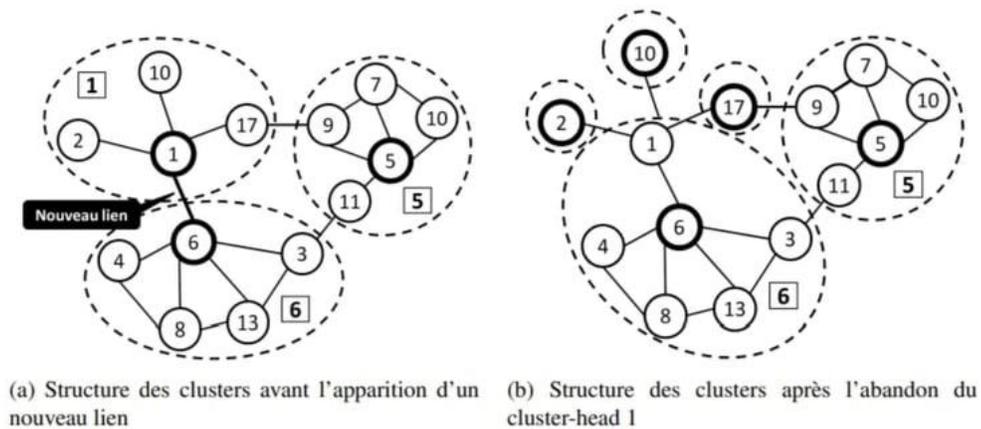


Figure 8 : Exemple de maintenance de la structure des clusters en cas d'apparition d'un nouveau lien

La figure 9 : illustre la maintenance de la structure des clusters pour plusieurs cas de figure de perte de lien. Cette figure présente quatre cas différents de perte de lien selon l'état des deux nœuds liés.

Cas 1 : il présente la perte d'un lien entre les nœuds 2 et 11 appartenant à deux clusters différents respectivement (1 et 6). Dans ce cas, aucune modification ne sera apportée à la structure des clusters puisqu'elle reste valide. En effet, cette perte de lien ne fait que diminuer l'inter-connectivité entre les clusters voisins.

Cas 2 : il présente la perte d'un lien entre deux nœuds membres 4 et 16 du même cluster (cluster 6). Dans ce cas, la structure des clusters n'est pas aussi modifiée. Et la perte de ce lien a comme effet de diminuer l'intra-connectivité dans un même cluster.

Cas 3 : il expose la perte de lien entre un nœud membre (nœud 12) et son cluster-head (nœud 6). Cette perte de lien viole la contrainte que chaque nœud membre est à un saut de son cluster-head. Par la suite, le nœud 12 quitte le cluster et revient à l'état non décidé. Pour rejoindre la structure des clusters, il applique les mêmes règles de la phase formation. Puisqu'il est voisin du cluster-head 5, il devient alors membre du cluster 5

Cas 4 : il illustre aussi le cas de perte de lien entre un nœud membre et son clusterhead. Dans ce cas le nœud 7 membre du cluster 5 quitte son cluster. Puisque qu'il n'est voisin à aucun cluster-head, et devient nœud cluster

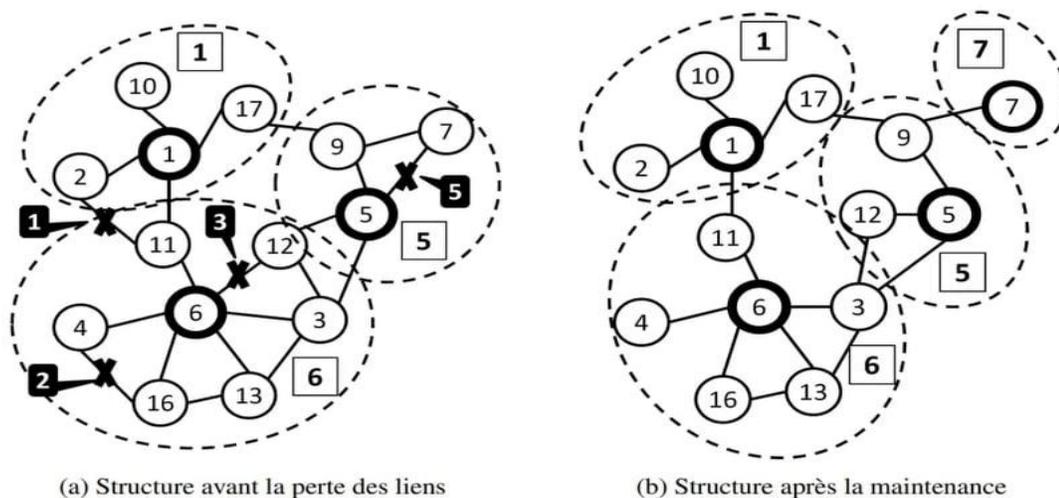


Figure 9 : Exemple de maintenance de la structure des clusters lors de la perte de lien

2.9 Etat de l'art

Plusieurs propositions de clustering auto-stabilisantes et déterministes ont été faites dans la littérature [40, 43, 45–48]. Les approches auto-stabilisantes [43, 45–47] construisent des clusters à 1 saut.

Mitton et al. [47] utilisent comme métrique la densité afin de minimiser la reconstruction de la structure en cas de faible changement de topologie. Chaque nœud calcule sa densité et diffuse l'information à ses voisins situés à distance k .

Flauzac et al. [43] ont proposé un algorithme auto-stabilisant qui combine la découverte de topologie et de clustering en une seule phase. Elle ne nécessite qu'un seul type message échangé entre nœuds voisins. Un nœud devient cluster-head s'il possède la plus grande identité parmi tous ses voisins. Johnen et al. [45] ont proposé un algorithme auto-stabilisant basé sur un modèle à état qui construit des clusters de taille fixe. Ils attribuent un poids à chaque nœud et fixe un paramètre « SizeBound » qui représente le nombre maximal de nœuds dans un cluster. Un nœud ayant le poids le plus élevé devient cluster-head et collecte dans son cluster jusqu'à « SizeBound » nœuds. Les approches auto-stabilisantes [40, 48] construisent des clusters à k sauts.

Dans [54], ils proposent un algorithme appelé algorithme de clustering pondéré (WCA) qui combine efficacement chacun des paramètres système ci-dessus avec certains facteurs de pondération choisis en fonction des besoins du système. Benkerdagh [53] a proposé un algorithme de clustering Fitness Clustering basé sur une fonction appelée « Fitness » qui combine plusieurs paramètres ayant un impact direct sur la stabilité des clusters.

2.10 Conclusion

Dans ce chapitre, nous avons étudié les techniques de clustérisations dans les réseaux VANET. Nous avons présenté une classification des algorithmes de clustérisations et les principales caractéristiques des approches proposées.

Le chapitre suivant présente notre proposition pour la stabilité des clusters.

Chapitre 3 Contribution et conception

3.1 Introduction

La première partie de ce chapitre est consacré à présentation de l’algorithme proposé. Cet algorithme a comme objectif de former les clusters et d’assurer leur stabilité. Dans la deuxième partie, nous allons présenter notre conception du projet à travers les diagrammes UML (Unified Modeling Language) tel que : Diagramme de cas d’utilisation, diagramme de séquence et diagramme de classe.

3.2 Principe modèle proposé :

A la base de algorithme de clustering (**fitnessclustering**) basé sur une fonction appeler fitness [53], nous avons proposé une approche pour sélectionner le nœud qui effectue la diffusion des messages d’alerte en cas de collisions multiples impliquant des voitures (signaler un état d’urgence) et nous comparons les résultats de simulation avec les résultats de l’algorithme **WCA** (Weight Clustering Algorithme).

- ❖ **Algorithme FitnessClustering** : est basé sur une fonction appelée *Fitness*, qui combine plusieurs paramètres ayant un impact direct sur la stabilité des clusters. C’est une fonction multi-objectif qui permet de garantir un meilleur compromis entre la stabilité, la surcharge du réseau et le taux de livraison pour les clusters produit. L’algorithme Fitness Clustering produit des clusters de différentes tailles, dont certains sont à **deux sauts**, ce qui a réduit le nombre de clusters formés.

❖ **Algorithme WCA (Weight Clustering Algorithm)** [54] est basé sur l'utilisation d'un poids combiné métrique prenant en compte plusieurs paramètres. Il utilise le degré d'un nœud, sa puissance d'émission, sa mobilité et son batterie pour calculer le poids combiné. Bien que WCA ait prouvé de meilleures performances algorithmes, il manque à connaître les poids de tous les nœuds avant de démarrer le processus de clustering et de vidanger rapidement les CH. Cet algorithme considère quatre métriques dans le calcul du poids d'un nœud: le degré D , la somme des distances avec ses voisins P , la mobilité relative moyenne M et le temps de service au tant que cluster-head T , tel que:

$$\text{Poids} = \alpha D + \beta P + \gamma M + \delta T \quad \text{avec: } \alpha + \beta + \gamma + \delta = 1$$

- Le degré $D(u)$ est la différence entre le degré du nœud u et une valeur m qui représente le nombre de nœuds qu'un cluster-head peut servir. Les auteurs n'explicitent pas le moyen de déterminer la valeur de m .
- Les distances $P(u)$ entre les nœuds sont calculées.
- La mobilité relative $M(u)$ d'un nœud représente le rapport des niveaux de puissance des transmissions successives reçues par un nœud de ses voisins.

❖ **Etat d'un véhicule** : Dans l'approche proposée, les véhicules peuvent être dans un des 5 états suivants : Nœud indéfini (UN), Chef de cluster (CH), Nœud Membre (NM) et chef de groupe secondaire(SCH) et Second-Membre (SM). Ces états sont définis dans le tableau suivant.

Etat	Description
UN	Etat initial d'un véhicule, il n'appartient à aucun cluster et il n'a aucun rôle
CH	Cluster –Head : véhicule chargé de la coordination entre les membres du cluster et assure la communication intra et inter cluster
SCH	Second-Cluster-Head : Le véhicule qui remplacera le CH principal, au cas où il deviendrait indisponible ou quitte le cluster.
NM	Noeud-Membre : Un véhicule dans un cluster mais qui n'est pas un CH ou un SCH.
SM	Second-Membre : Le véhicule qui remplacera le SM principal, au cas où il deviendrait indisponible ou quitte le cluster.

Tableau 2 : états d'un véhicule.

3.3 Les algorithmes :

3.3.1 Algorithme WCA

Algorithme de formation des clusters et processus de sélection de CH

Entrées : nombre de nœuds voisins, position, vitesse, identifiant de route, direction, message d'accueil, identifiant de nœud

Sortie : Weight value

1 : Broadcast. xi

2 : Si xi reçoit Cjrm Then // Cluster déjà existant à proximité

3 : Si plusieurs trames balises sont reçues Alors // Déterminer le SCH le mieux adapté à rejoindre

4 : Si (pos.SCHi > pos.xi) & (rel_sp.NM < rel_sp.SCHj) Alors // Comparer la position et la vitesse relative

5 : Rejoindre Clusteri // Rejoindre clusteri et s'affilier à NM i

6 : xi.status = NM // Le véhicule accuse réception de la demande et rejoint le cluster

7 : Sinon

8 : Rejoindre le cluster j // Rejoindre le cluster j et s'affilier à SCHj

```

9 : Fin si
10 : Fin si
11 : Fin si
12 : Passez à l'étape 35
13 : Si  $x_i \in x_L$  Alors //où  $x_L$  est l'ensemble de tous les véhicules dans Trange
14 : Adéquation.  $x_i()$  // calcule la valeur d'adéquation d'un véhicule
15 :  $x_i$ .Broadcast hello_message//informations sur l'état de la diffusion du véhicule avec valeur d'adéquation
16 : Tx.  $x_i \leftarrow = = = x_i.Tx()$  //calculer le temps d'attente d'une réponse cluster head
17 : Pendant que Tx.  $x_i > 0$  faire
18 : Si Initiate Cluster (CHid) est reçu Alors
19: Stop Competition () //CH trouvé
20 : Proces sInitiate Cluster (CHid) // traite le message reçu
14 : Sinon
21 : Décrémenter le temps d'attente Tx.  $x_i$ 
22 : Fin si
22 : Fin si
23 : Si poids_valeur.  $x_i = =$  poids_valeur. $x_j$  Alors
24: Si ( $N_i < N_j$ ) & ( $nodeid.x_i > nodeid.x_j$ ) Alors //déterminer le véhicule avec plus de voisins
25 :  $x_j.status \leftarrow = = = SCH$  // le véhicule j change son statut en SCH
26 :  $x_j.id \leftarrow = = = CHid$  // l'identifiant du véhicule est défini comme un identifiant de cluster
27 : Sinon
28 :  $x_i.status \leftarrow = = = SCH$  // véhicule i change son statut en SCH
29 :  $x_i.id \leftarrow = = = CHid$  // l'identifiant du véhicule est défini comme un identifiant de cluster
30 : Fin si
31 : Fin si
32 : Fin pendant
33 : SendInitiateCluster(CHid) // CH diffuse son identifiant à tous les véhicules voisins
34 : Fin si
35 : Fin //Fin de la formation du cluster et du processus d'élection

```

Cluster –Head →CH

Cluster initiate message →Cini

Cluster join message → Cjrm

Neud –membre → NM

Highest_xi → Valeur de pondération la plus élevée parmi les membres du cluster

Pos → Position d'un véhicule

Rel_sp → Vitesse relative

SCH → second- cluster-head

Tx → Temps d'attente d'une réponse du chef de cluster

xi Véhicule actuel

xj → Véhicule voisin à portée de transmission

xL → Ensemble de véhicules voisins à portée de transmission

3.3.2 Algorithme Fitnessclustering

Algorithme 1 calcul de fitness

Algorithme1:Calcul de *Fitness*.

```
1:Diffuser les messages "Hello"  
3:  $CT \leftarrow TS_{(Hello)}$   
4:  
5: tantque  $CT - TS_{(Hello)} < AVI_{(Hello)}$  faire  
6:   Attendre la réception d'un message "Hello"  
7:   Enregistrer Mac Address dans  $NV []$   
8:   Calculer la distance ( $Dist_j$ )  
9:    $CT \leftarrow Current\ Time$   
10:fin tantque  
11  
12:  $CD \leftarrow NV[].length$   
13:  
14:pour chaque  $v_j \in NV []$  faire  
15:   Calculer la période de transmission  $TRP_j$   
16:   Calculer la vitesse relative  $Vr_j$   
17:   Calculer la durée de vie des liens  $LV_j$   
18: fin pour  
19:
```

Pour calculer la valeur « Fitness » d'un véhicule v_i ($Fitness_i$) (cf. Équation 3.5). La fonction objective vise à minimiser la valeur de la « Fitness » afin de choisir un chef de cluster (CH) pour chaque groupe de véhicules (cf. Équation 3.6)

$$Fitness_i = \alpha \cdot \sum_{i=0}^{CD} \frac{TRP}{n} \beta \cdot CD + \gamma \cdot \sum_{i=0}^{CD} \frac{Vr_j}{n} - \sigma \cdot \sum_{i=0}^{CD} \frac{LV_j}{n} \quad (3.5)$$

$$CH_j \leftarrow \text{argh}(\min(Fitness_i)) \quad (3.6)$$

Où :

- α , β , γ , et σ sont des facteurs positifs.
- T_{RPj} fait référence à la période de transmission entre le nœud actuel et son voisin j (V et V_j), exprimée par la moyenne de la différence entre le temps de réception de la réponse T_{RCj}
- n : nombre de nœuds dans une zone
- CD est le degré de connectivité
- La mobilité relative V_{rj}
- La durée de vie du lien LV_j entre le véhicule actuel et son voisin j

Lorsque le temps de transmission entre les nœuds (T_{RPj}) et leur mobilité relative (V_{rj}) sont inférieurs, le degré de connectivité (CD) et la durée de vie des liens entre les véhicules (LV_j) sont plus élevés, le nœud est plus susceptible d'être un CH (cf. Equation 3.5, 3.6). Dans les sections suivantes, cet algorithme est désigné par le nom Fitness Clustering.

Rfitness est un paramètre important de la fonction Fitness. Cela conduit à catégoriser cet algorithme en tant que protocole de Clustering basé sur la mobilité. En plus de ces paramètres, la période de transmission (TRP) et la durée de vie des liens (LV) permettent de prédire l'état futur du réseau, ce qui permet de classer l'approche comme étant une méthode de classification prédictive.

La mobilité relative V_{rj} est exprimée par la différence entre la vitesse du véhicule actuel V et la vitesse du véhicule voisin V_j . La formule de la mobilité relative est exprimé comme suit : $V_{rj} = |V - V_j|$.

Algorithm 2: algorithme fitnessclustering pour la selection de cluster head

Algorithme2: Élection du CH et délégation de SCH pour chaque véhicule.

```
1:Initialiser:RecordFitness[]=NULL,Response=NULL,Demand=0,minFit=0,RelayNode=NU
LL,my_CH=NULL,my_SCH=NULL
2:Envoyer Fitness à tous les véhicules voisins
3:Attendre un temps t

4:Enregistrer la Fitness reçue dans RecordFitness[]
5:
6:si (CD=0) ou (RecordFitness[]=NULL) alors
7:    my_CH←-myMacAdress
8: sinon
9:    my_fit←-- Min(RecordFitness [])
10:   my_CH←MacAddress(Node minFit)
11:   Envoyer_CH_Demand à my_CH
12:   Demand←Demand +1
13:   Attendre un temps t
14:
15:   si (Response = CH_ACCEPTED) alors
16:     Envoyer my_CH et my_SCH à tous les voisins
17:   sinon
18:
19:     si (Demand=1) alors
20:       Relay nodes←my_CH
21:       minFit← Min(RecordFitness[] - {my_CH})
22:       Retour à laligne 10
23:     sinon
24:       my_SCH←RelayNode
25:       my_State←SM
26:
27:       Envoyer my_CH et SCH à tous les voisins
28:     fin si
29:
```

```
30:   fin si
31:
32:   si (my MacAddress  $\neq$  Designated SCH) and (my_state  $\neq$  SM) alors
33:     my_State  $\leftarrow$  MN
34:   fin si
35:
36: fin si
```

Le processus général d'élection des CH est expliqué dans l'algorithme 2.

Le processus de diffusion des messages "Hello" et de calcul de la Fitness est résumé dans l'algorithme 1, où :

— CT fait référence au temps actuel.

— T S est l'estampille temporelle du message "Hello".

— AV I représente l'intervalle de validité absolue du message "Hello"

Le CH est choisi en fonction de la Fitness du véhicule et de ses voisins. Chaque nœud échange sa Fitness avec ses voisins NV et doit attendre un temps t pour enregistrer toutes les Fitness des véhicules dans la table RecordFitness.

Le CH est initialisé au premier voisin MacAddress. Après avoir comparé toutes les valeurs de Fitness, la plus petite valeur sera sélectionnée et comparée avec la Fitness locale. Le minimum sera donc sélectionné. Chaque voisin qui ne répond pas sera supprimé de NV. Après ces étapes, un CH_DEMAND sera envoyé au nœud sélectionné my_CH. Ce dernier envoie un CH_ACCEPTED ou un CH_REFUSED basé sur son adhésion à un groupe et la différence de Fitness.

Un CH_ACCEPTED sera envoyé si my_CH n'appartient pas à un autre cluster ou s'il s'agit d'un membre et que la différence des ratios de Fitness des membres demandant le même

nœud pour être leur CH est inférieure à la différence des ratios de Fitness avec les autres voisins ayant le même CH que le nœud concerné.

Algorithme 3 algorithme de maintenance fitnessclustering

Maintenance des clusters.

```

2:   Envoyer CH_Checking à my_CH
   Envoyer SCH_Checking à my_SCH
3:   Envoyer NeighbourCH_Checking à  $CH \in NV[]$ 
4:   Attendre un temps t
   Si CH_Confirmed=NULL alors
5:     Si SCH_Confirmed≠NULL alors
6:       my_CH← SCH
   sinon
7:     Si NeighboursCH_Confirmed≠ NULL alors
8:       my_CH← argmin{FitnessNeighboursCH}
   sinon
9:     Si  $v_i$  est en communication avec  $v_k$  et  $v_k$  est en communication avec
   my_CH et /ou my_SCH et/ou Neighbours CH alors
14:      Calculer la RFitnessCHi
15:      Calculer la RFitnessmy_SCHi
16:      Calculer la RFitnessNeighboursCH
17:      my_CH← argmin (RFitCHi, RFitSCHi, RFitNeighboursCH)
18:    sinon
19:      my_CH ← my MacAddress
20:    finsi
21:  finsi
22: finsi
23: finsi
24: fin pour

```

La maintenance de ce dernier sera exécutée pour assurer la restructuration des clusters et la fiabilité des processus exécutés. Dans cette phase, une métrique appelée Fitness relative.

RFitness est utilisée pour évaluer les liens (ou les communications). Un nœud intermédiaire v_k est utilisé dans le cas où un nœud perd sa connexion directe avec son CH ou SCH. Les Fitness relatives pour CH_i , SCH_i , CH_j sont calculées respectivement selon les équations 3.7,

3.8 et 3.9. Ces paramètres font référence aux Fitness calculées indirectement lorsque la connexion est perdue entre un membre et son CH ou SCH.

$$RFitCH_i = RTRP_{k,CH_i} + RVRP_{k,CH_i} - RVRP_{k,CH_i} \quad (3.7)$$

Où :

- RT RP, RV r, RLV sont des paramètres qui font référence à la période de transmission relative, à la mobilité relative et à la durée de validité relative du lien, respectivement, entre deux nœuds connectés indirectement.

$$RTRP_{k,CH_i} = \frac{TRP_k + TRP_{k,CH_i}}{2}$$

$$RVR_{k,CH_i} = \frac{V_{rk} + V_{rk,CH_i}}{2}$$

$$RLV_{k,CH_i} = \frac{LV_k + LV_{k,CH_i}}{2}$$

$$RFitSH_i = RTRP_{k,SCH_i} + RVRP_{k,SCH_i} - RVRP_{k,SCH_i} \quad (3.8)$$

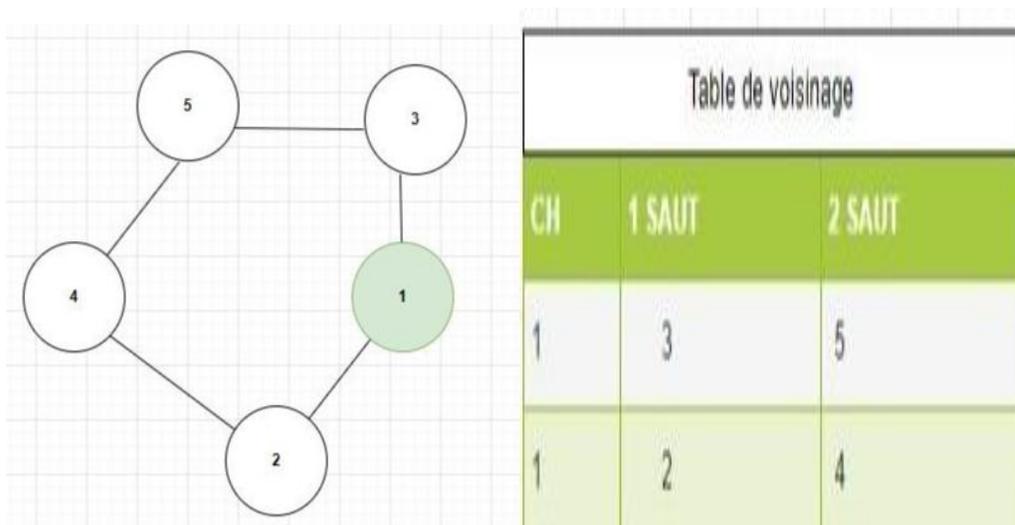


Figure 10 : schéma de l'algorithme FitnessClustering.

3.4 Conception:

3.4.1 Méthode de conception :

Définition UML :

C'est l'acronyme anglais pour « Unified Modeling Language » (en français : Langage de modélisation unifié). La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. Le langage UML ne préconise aucune démarche, ce n'est pas une méthode [50].. Chacun est libre d'utiliser les types de diagramme qu'il souhaite, dans l'ordre qu'il veut. Il suffit que les diagrammes réalisés soient cohérents entre eux, avant de passer à la réalisation du simulateur.

3.5 Diagrammes UML :

3.5.1 Diagramme de use case :

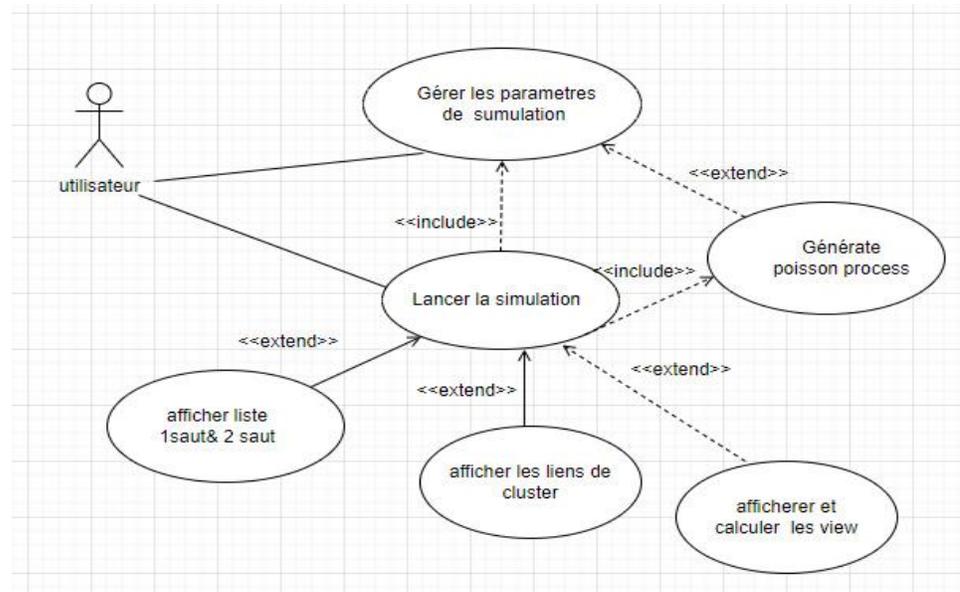


Figure 11 : Diagramme de use case

3.5.2 Diagramme de classe :

Nous présentons les classes utilisées pour implémenter l’algorithme proposé. La classe principale est la classe nodes . La classe véhicule, classe mobilitéetc.

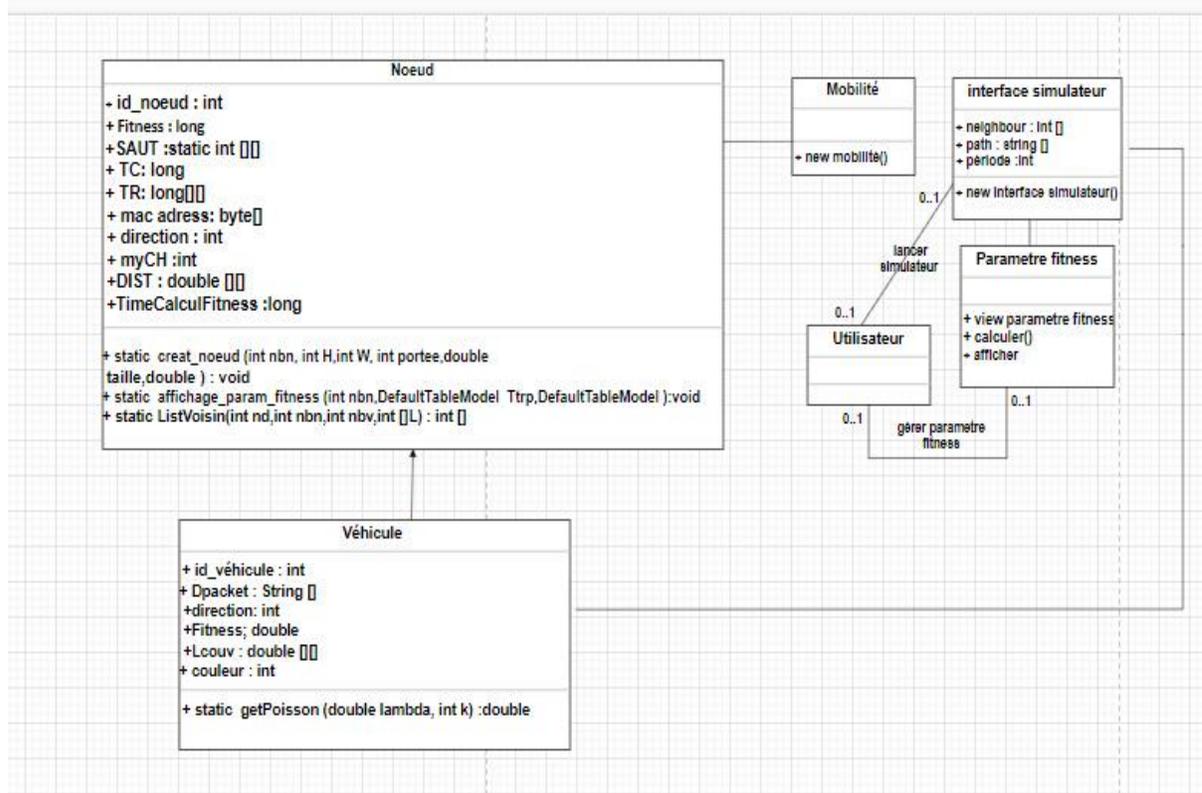


Figure 12 : Diagramme de classe

3.5.3 Diagramme de séquence :

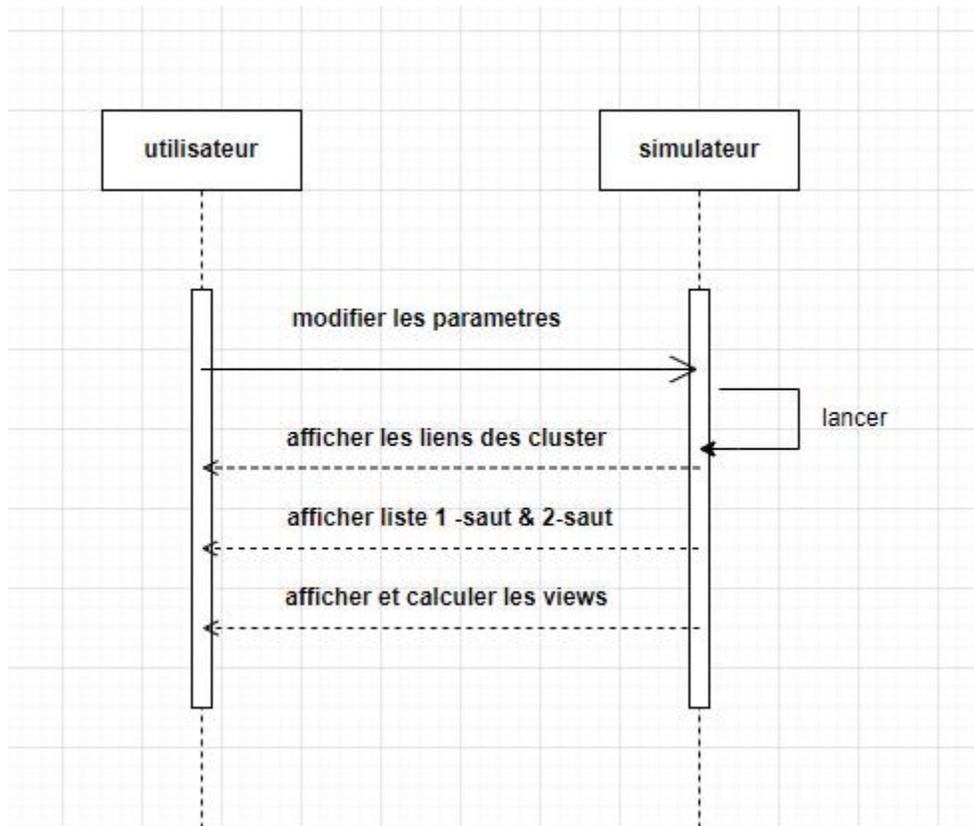


Figure 13 : Diagramme de séquence

3.6 Conclusion :

Dans ce chapitre, nous présentons un algorithme WCA et Fitness clustering pour sélectionner le nœud qui effectue la diffusion des messages d’alerte en cas de collisions multiples impliquant des voitures pour les réseaux véhiculaires basés sur le mode de communication V2V. Et nous présentons notre modélisation du simulateur à travers le langage UML, puis nous détaillons les 2 algorithmes que nous utiliserons dans l’étape simulation et l’implémentation.

L'implémentation et simulation

4.1 Introduction :

Après avoir présenté une description détaillée de notre contribution qui consiste à assurer la stabilité, nous allons présentons la partie de simulation et l'implémentation.

4.2 Notre application (simulateur):

L'objectif principal de notre application est l'étude des performances des protocoles appliqués aux réseaux VANETs par le simulateur pour assurer la stabilité entre les clusters.

4.3 Simulation :

Nous avons développé une application pour simuler un réseau VANeT, ce dernier ne permet d'évalué certain algorithmes de clustérisassions comme **WCA** et **FinessClustering**, nous avons procéder comme suit :

1. Paramétrage de simulation :

Nombre de nœud (véhicules), vitesse de déplacement, les positions et la portée de chaque véhicule.

2. **Création une zone de simulation** sous forme d'autoroute de 6 voie, les véhicules sont affectés et positionnés sur les voies selon une loi de distribution de poisson. Une fois les véhicules prennent leurs positions on lance la simulation.

La simulation est divisée en 3 périodes répétitives :

a. Premier période (Sélection de Cluster-Head) :

Après avoir collecté les informations partagées entre les véhicules voisins. Chaque véhicule examine les informations obtenues pour trouver le nœud compétent et le plus approprié qui mérite de devenir un nœud leader (CH). Les règles de choix du CH varient selon certaines métriques déterminées. Lorsqu'un nœud se déclare candidat pour présider un cluster, il peut être élu par ses nœuds voisins ou s'auto-élire, en mettant à jour son état à CH. Une fois que ce dernier est élu car il est le mieux adapté pour être le responsable, il passe à la phase d'annonce.

b. Deuxième période (Formation des clusters) : un cluster est composé de 4 types de nœuds.

Type 1: Cluster-Head (CH)

Type 2: Sous-Cluster-Head (SCH)

Type 3 : Nœud Normal (Nm)

Type 4 : Nœud Second (Ns)

Ces 4 types prennent différents couleur dans le simulateur.

c. Troisième période (Maintenance des clusters) : Après la formation de cluster et avec la mobilisation d'un CH, l'un de ces membres devient un cluster-head.

4.3.1 Environnement de travail :

Nous avons développé application (simulateur) dans IDE **Apache NetBeans IDE**

Apache NetBeans IDE:

C'est un environnement de développement intégré (EDI) qui est placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C , le C++ , le JavaScript , le XML , le Groovy , le PHP et le HTML , ou d'autres

(dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Compilé en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)).

Comment installer Apache NetBeans ?

1. Une fois l'exécutable téléchargé, double cliquez dessus afin de lancer l'**installation**.
2. La fenêtre de lancement ci-dessous apparaît. ...
3. Cliquez sur le bouton Next.
4. Cochez l'acceptation des termes du contrat puis cliquez sur le bouton Next.
5. Cochez l'acceptation des termes du contrat puis cliquez sur le bouton **Next**
6. Sélectionnez l'emplacement d'installation de NetBeans, par exemple C:\NetBeans-12.0.
7. Sélectionnez le **JDK** qui va être utilisé par l'IDE. Par défaut, l'installateur a détecté la présence de la version 14 du kit, précédemment installée.

4.4 Les résultats de simulation :

4.4.1 Approches implémentés

On a effectué la simulation de 2 algorithmes :

- **WCA** (weight clustering algorithm)
- **Fitness clustering**

4.4.2 Les paramètres de simulation :

Les paramètres de simulation sont résumés dans le tableau suivant :

Paramètre	Valeur
Durée de simulation	300 s
Zone de simulation	50000*300 pixel
Vitesse	10..100km/h
Range de transmission	150 km
Le temps de transmission des messages périodiques	1 s
Nombres des voies	6
Taille d'un Packet envoyée (taille message hello)	64 octet
Nombre maximal de sauts	2 sauts
Temps de formation	19 ms
Temps de maintenance	3.16 ms après la formation du cluster

Tableau 03: Paramètres de simulation

4.4.3 Scénario :

Nous supposant que les véhicules circulent sur l'autoroute de six voies. Au début de simulation, les véhicules prennent la couleur « vert ». Présente un aperçu de simulation où

les véhicules ont des couleurs différents (CH prend la couleur rouge et les véhicules avec l'état Sch prennent la couleur bleu, et véhicules avec l'état SM prennent la couleur jaune). Les nœuds SCH ont pour rôle de gérer les clusters lorsque le CH est déconnecté. La vitesse maximale sur une voie influence inversement la durée de vie d'un SCH, car dans un environnement très mobile, les nœuds CH ont tendance à se déconnecter plus fréquemment de leurs clusters. Donc, les SCH ont pour rôle de gérer les clusters jusqu'au prochain cycle de Clustering.

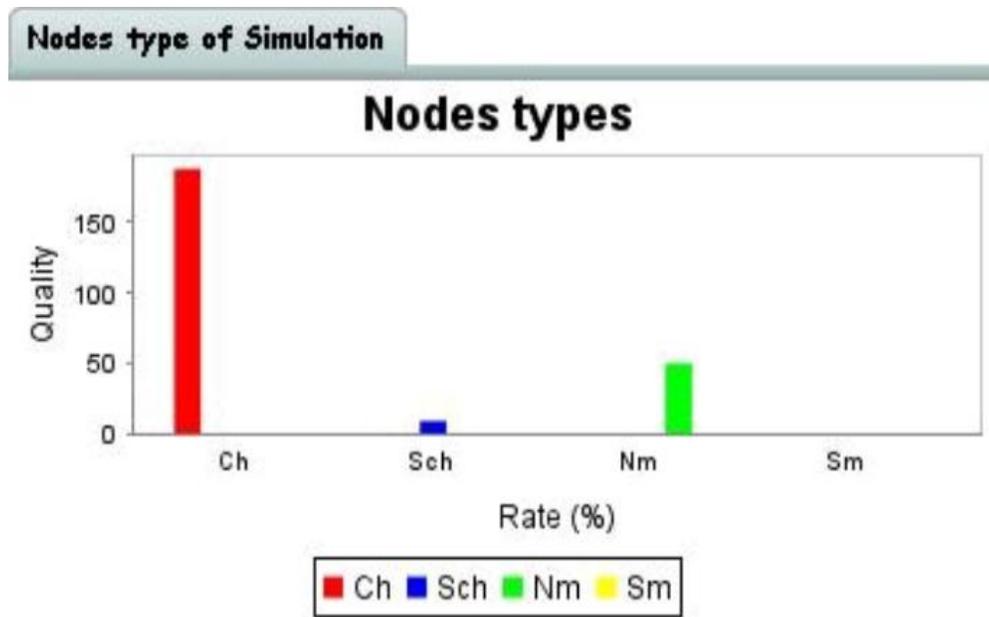


Figure 14 : Types nœuds

4.5 L'implémentation :

Réalisation et Interprétations :

Une fois notre programme est lancé, la fenêtre décrite par la (Figure15) est affichée. C'est la page d'accueil et elle représente la première rencontre avec la plateforme de simulation dans

laquelle nous avons introduit brièvement le cadre et le groupe de travail. À partir de cette fenêtre, nous pouvons accéder aux autres interfaces pour commencer les simulations.

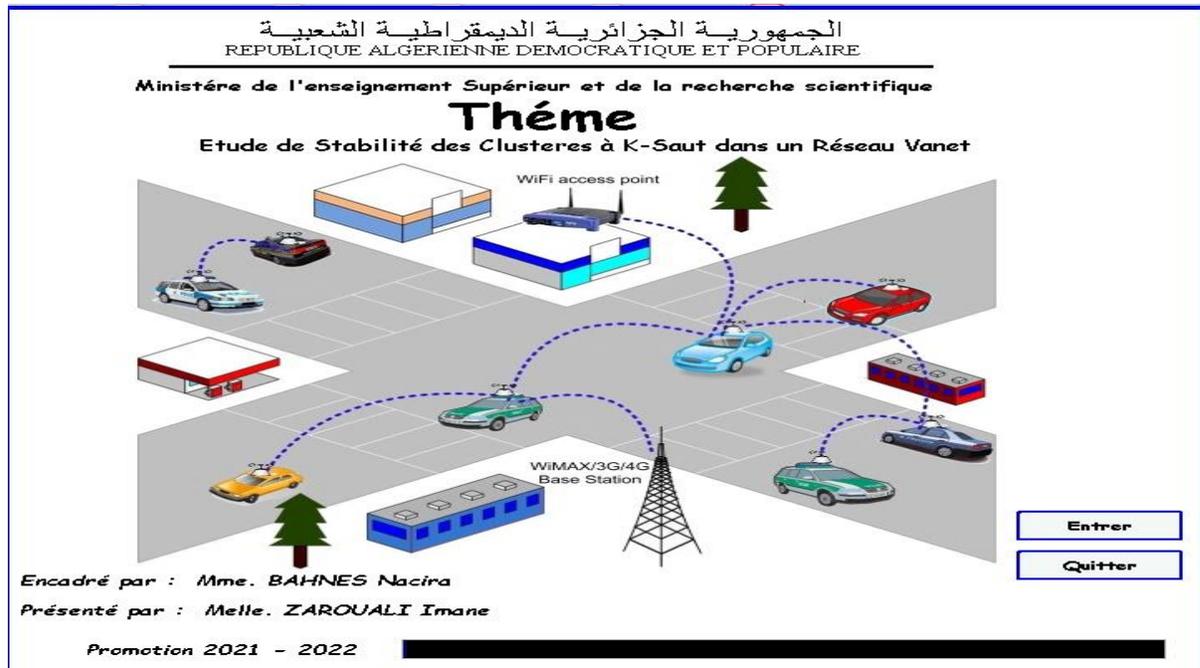


Figure 15 : Page d'accueil

La figure 16 représente le menu principal de notre simulateur, il compose de 4 parties :

- La partie en haut : représente paramétrage de notre simulation.
- La partie gauche : représente affectation des nœuds et leurs types
- La partie droite : représente les voies dans l'autoroute dans simulateur
- La partie en bas : caractéristiques de chaque nœud, énergie...etc.

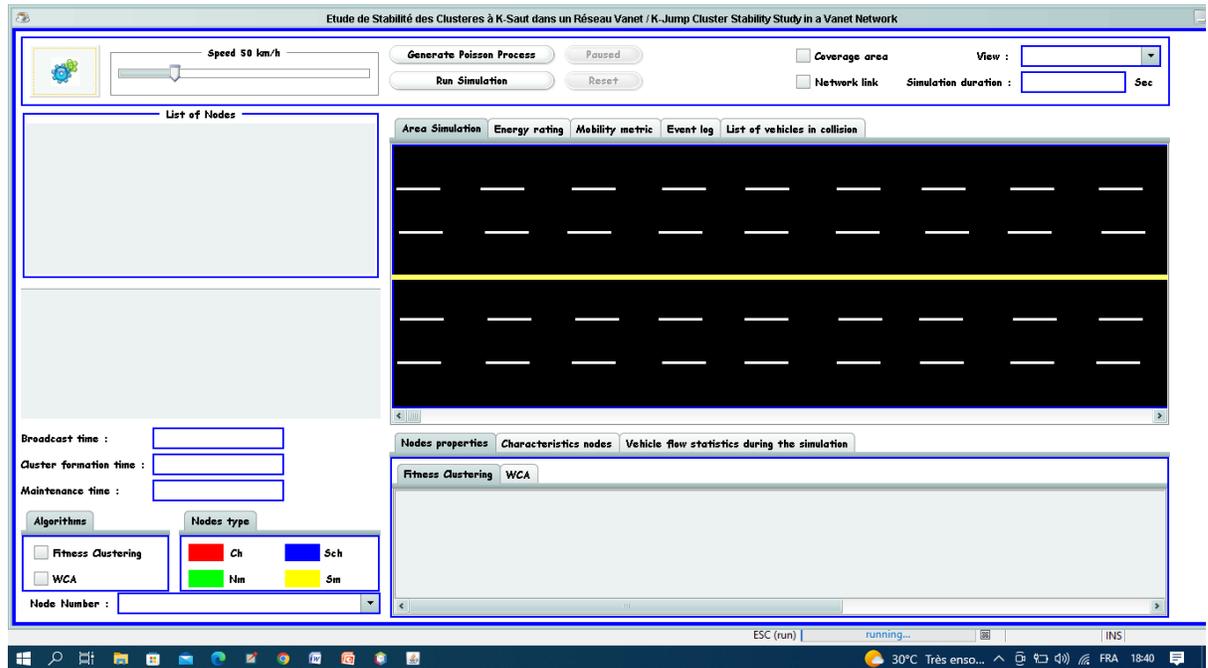


Figure 16 : l'interface de menu principale de notre simulateur

La figure 17 illustre le paramétrage de réseau pour les deux méthodes (WCA et Fitnessclustering)

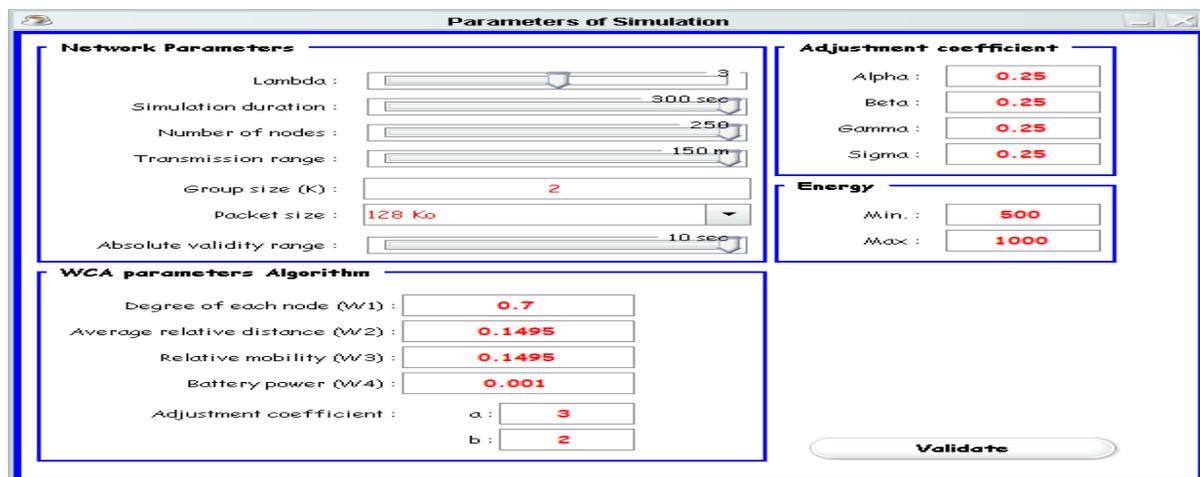


Figure 17 : Paramétrage de réseau et méthode WCA et fitness clustering

La Figure 18 représente la partie de distribution des nœuds dans des voies en utilisant la loi de poisson.

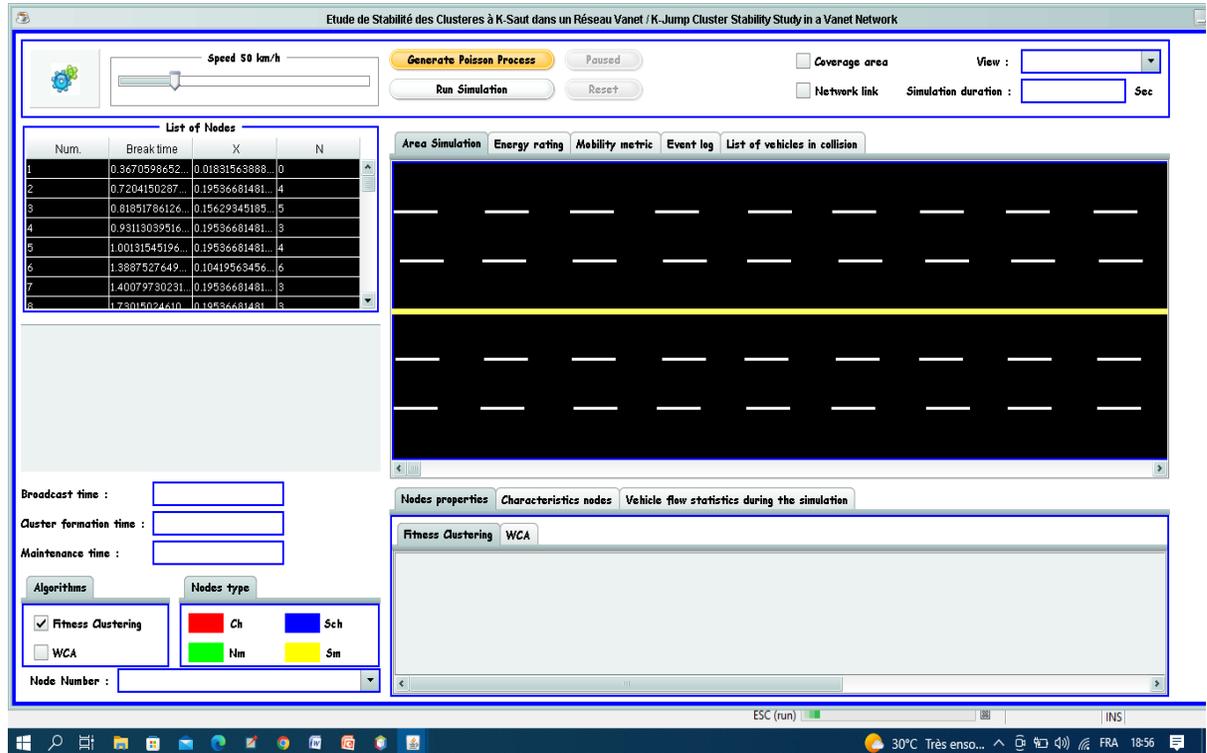


Figure 18 : représente la partie de distribution des nœuds

La figure 19 représente le lancement dans notre simulateur, on voit la circulation imulation des véhicules sur les routes ainsi que leur type de nœud (CH, SCH, SM, Nm).

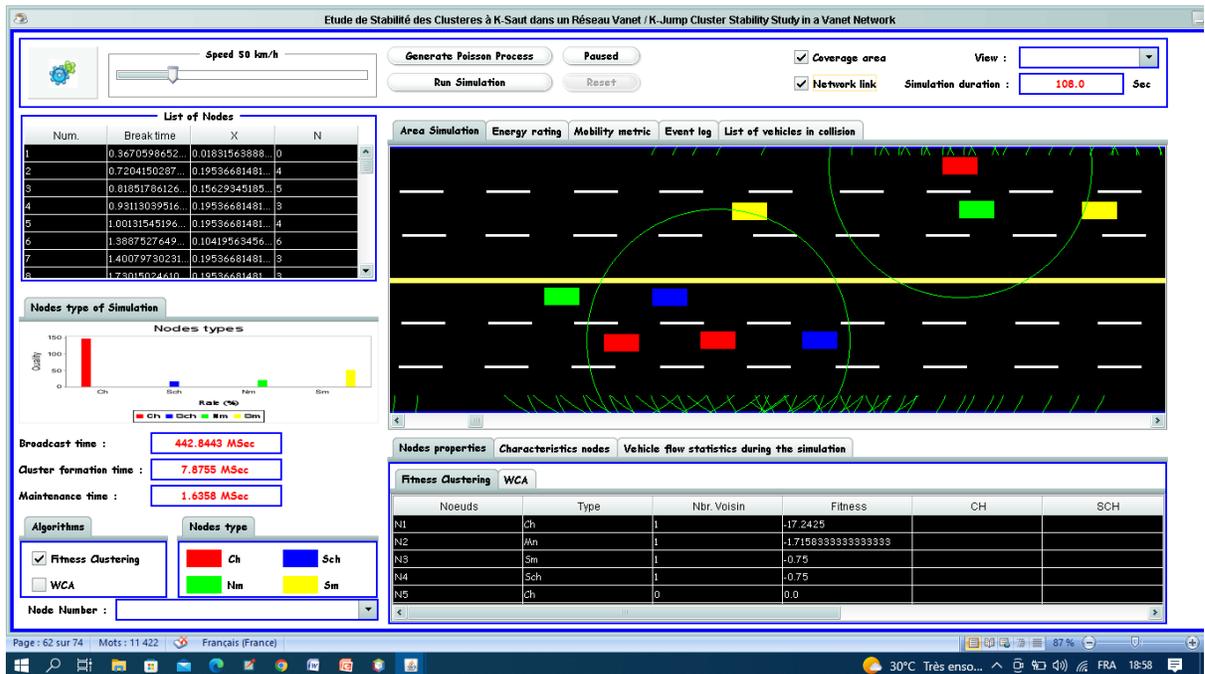


Figure 19 : Aperçu de simulation des véhicules dans notre réseau

La figure 20 représente la courbe de mobilité relative d’algorithme fitness clustering en fonction de temps.



Figure 20 : Mobilité relative (fitnessclustering)

La figure 21 représente la mobilité relative d’algorithme WCA en fonction de temps.

	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15	N16	N17	N18	N19	N20	
N1																					
N2																					
N3																					
N4																					
N5																					
N6									83.1												
N7																					
N8																					
N9					83.1																
N10																					
N11																					
N12																					
N13																					
N14																					
N15																					
N16																					
N17																					
N18																					
N19																					
N20																					
N21																					

Figure 23 : Distance entre 2 nœuds

la figure 24 représente le path (chemin) entre chaque paire de nœuds (ni,nj).

	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12
N0	N1,	N1,	N1,	N1,	N1,	N1,	N1,	N1,	N1,	N1,	N1,	N1,	N1,
N1	N2,	N2,	N2,	N2,	N2,	N2,	N2,	N2,	N2,	N2,	N2,	N2,	N2,
N2	N3,	N3,	N3,	N3,	N3,	N3,	N3,	N3,	N3,	N3,	N3,	N3,	N3,
N3	N4,	N4,	N4,	N4,	N4,	N4,	N4,	N4,	N4,	N4,	N4,	N4,	N4,
N4	N5,	N5,	N5,	N5,	N5,	N5,	N5,	N5,	N5,	N5,	N5,	N5,	N5,
N5	N6,	N6,	N6,	N6,	N6,	N6,	N6,	N6,	N6, N9,	N6,	N6,	N6,	N6,
N6	N7,	N7,	N7,	N7,	N7,	N7,	N7,	N7,	N7,	N7,	N7,	N7,	N7,
N7	N8,	N8,	N8,	N8,	N8,	N8,	N8,	N8,	N8,	N8,	N8,	N8,	N8,
N8	N9,	N9,	N9,	N9,	N9,	N9, N6,	N9,	N9,	N9,	N9,	N9,	N9,	N9,
N9	N10,	N10,	N10,	N10,	N10,	N10,	N10,	N10,	N10,	N10,	N10,	N10,	N10,
N10	N11,	N11,	N11,	N11,	N11,	N11,	N11,	N11,	N11,	N11,	N11,	N11,	N11,
N11	N12,	N12,	N12,	N12,	N12,	N12,	N12,	N12,	N12,	N12,	N12,	N12,	N12,
N12	N13,	N13,	N13,	N13,	N13,	N13,	N13,	N13,	N13,	N13,	N13,	N13,	N13,
N13	N14,	N14,	N14,	N14,	N14,	N14,	N14,	N14,	N14,	N14,	N14,	N14,	N14,
N14	N15,	N15,	N15,	N15,	N15,	N15,	N15,	N15,	N15,	N15,	N15,	N15,	N15,
N15	N16,	N16,	N16,	N16,	N16,	N16,	N16,	N16,	N16,	N16,	N16,	N16,	N16,
N16	N17,	N17,	N17,	N17,	N17,	N17,	N17,	N17,	N17,	N17,	N17,	N17,	N17,
N17	N18,	N18,	N18,	N18,	N18,	N18,	N18,	N18,	N18,	N18,	N18,	N18,	N18,
N18	N19,	N19,	N19,	N19,	N19,	N19,	N19,	N19,	N19,	N19,	N19,	N19,	N19,
N19	N20,	N20,	N20,	N20,	N20,	N20,	N20,	N20,	N20,	N20,	N20,	N20,	N20,
N20	N21,	N21,	N21,	N21,	N21,	N21,	N21,	N21,	N21,	N21,	N21,	N21,	N21,

Figure 24 : Path (chemin) entre 2 nœuds

La figure 25 représente la densité du véhicule pour chaque instant donnée

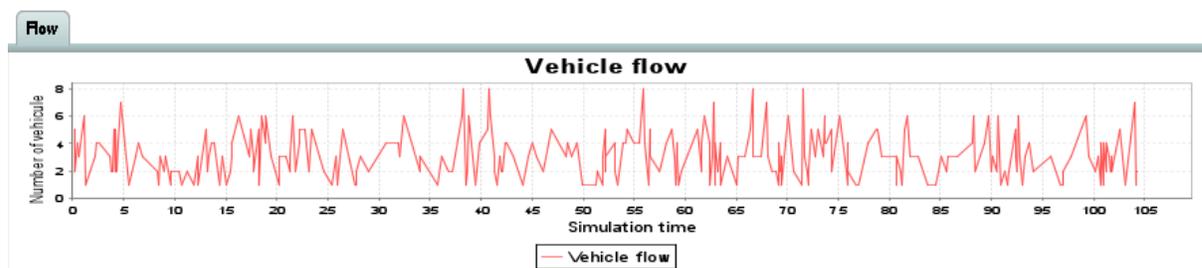


Figure 25 : Densité du véhicule

La figure 26 illustre les caractéristiques d'un nœud sélectionné (type, nb voisins, liste des voisins a 1-saut et 2 –saut).

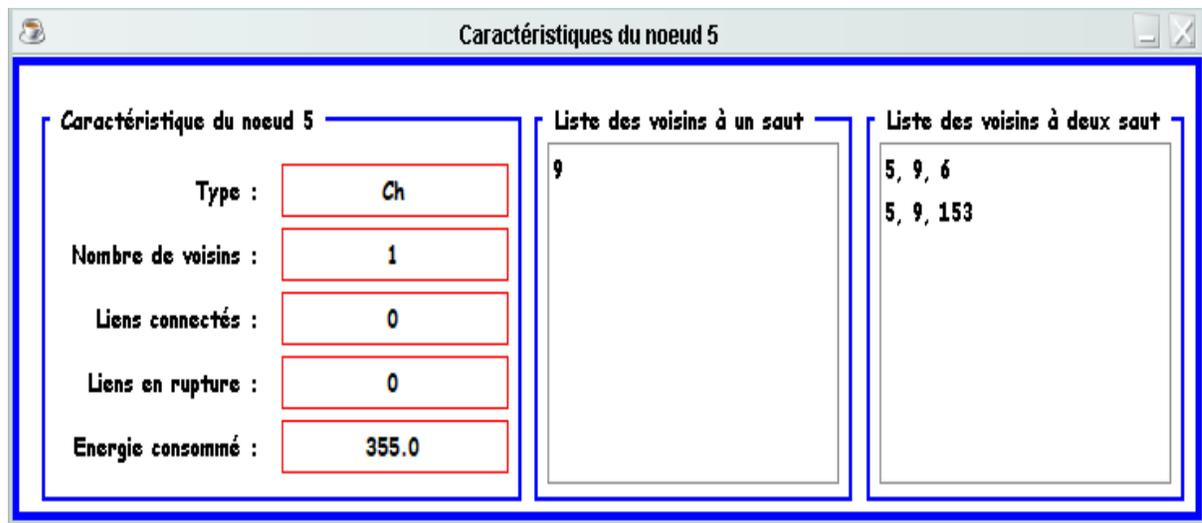


Figure 26 : Caractéristiques du nœud

La Figure 27 représente la liste des véhicules à la collision.

Collision entre véhicule 272 est 19 à l'instant 0.0
Collision entre véhicule 321 est 234 à l'instant 0.0
Collision entre véhicule 224 est 194 à l'instant 7.0
Collision entre véhicule 11 est 1 à l'instant 11.0
Collision entre véhicule 137 est 68 à l'instant 24.0
Collision entre véhicule 224 est 194 à l'instant 32.0
Collision entre véhicule 206 est 186 à l'instant 54.0
Collision entre véhicule 206 est 186 à l'instant 55.0
Collision entre véhicule 321 est 234 à l'instant 66.0
Collision entre véhicule 272 est 19 à l'instant 70.0
Collision entre véhicule 94 est 293 à l'instant 80.0
Collision entre véhicule 112 est 55 à l'instant 81.0
Collision entre véhicule 112 est 230 à l'instant 81.0
Collision entre véhicule 112 est 55 à l'instant 82.0
Collision entre véhicule 112 est 230 à l'instant 82.0
Collision entre véhicule 112 est 55 à l'instant 83.0

Figure 27 : Liste des véhicules en collision

4.6 Analyse des performances des algorithmes et comparaison des résultats :

Pour évaluer les performances de routage des approches **FitnessClustering** et **WCA** , la simulation a été lancée avec différentes d'événements : (broadcast, formation, maintenance).

Pour fitness clustering:

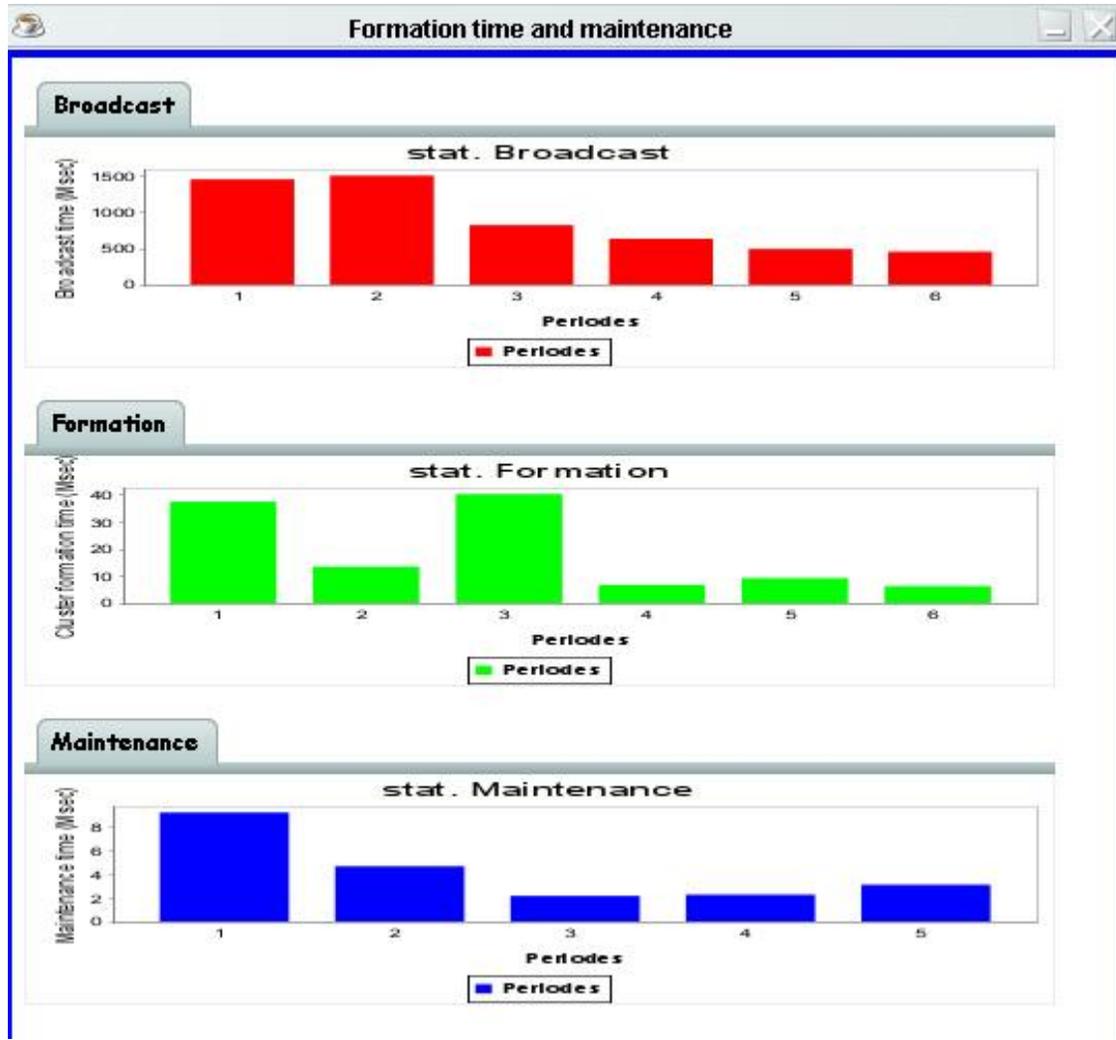


Figure 28: Formation et maintenance des clusters

Pour WCA:



Figure 29 :
Formation et
maintenance
avec WCA

4.7 Comparaison entre les deux algorithmes fitnessclustering et WCA :

Après les simulation , nous constater que le fonctionnement d’algorithme fitness clustering est plus rapide et que WCA et que FitnessClustering présentent de meilleurs résultats que la algorithme WCA

4.8 Conclusion :

Dans ce chapitre, nous avons donné un aperçu général de notre application en utilisant la simulation, ensuite l’environnement de simulation. Nous avons aussi présenté la partie de simulation, l’implémentation, analyse des performances des algorithmes et comparaison des résultats.

Conclusion générale

Les réseaux véhiculaires constituent un nouveau type des réseaux issu des réseaux ad-hoc mobiles, leur particularité provient des communications qui peuvent s'instaurer entre véhicules ou bien avec une infrastructure (station de base).

Le développement des nouvelles technologies a favorisé une formidable évolution des réseaux véhiculaires. Cette évolution vise à rendre les réseaux plus efficaces, plus sûrs et plus écologiques aussi bien du point de vue de l'industrie automobile que des opérateurs de réseaux et de services.

Il y a beaucoup de défis qui doivent être abordés lors de la création d'un réseau véhiculaire. L'un des défis auxquels font face les réseaux Ad-hoc est la topologie du réseau qui change rapidement. Les véhicules dans un VANET ont un degré élevé de mobilité.

Dans cette étude, nous avons proposé un schéma de clustering qui assure la formation et la maintenance des clusters.

Perspectives :

Le travail présenté ouvre des voies d'améliorations que l'on peut donner comme suit :

- Tester notre proposition sous le simulateur NS3.
- Généraliser l'approche proposée de 3-sauts à k-sauts.

Bibliographie

- [1] <https://www.informatique-mania.com/linternet/ad-hoc-reseau/> **consulter le 23/01/2022**
- [2]:https://www.google.com/search?q=reseau+ad+hoc&rlz=1C1XXVF_frDZ985DZ985&sxsrf=APq-WBscKT9DNDKK57MbeMQnUC_hzXMM1w:1646398889305&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiAkdLnwaz2AhXwi_0HHZRBC3YQ_AUoAXoECAEQAw&biw=1366&bih=635&dpr=1#imgrc=UUYZ5VXzq-C3sM&imgdii=9FCBPfPyRFrDxM
consulter le 5/02/2022
- [3]:<https://docplayer.fr/11708658-Memoire-presenter-par-tahar-abbes-mounir-pour-obtenir-le-diplome-de-doctorat-specialite-informatique-intitule.html> **consulter le 7/02/2022**
- [4] :<https://www.ummo.dz/dspace/bitstream/handle/ummo/12331/ChemounKarima.pdf?sequence=1> **consulter le 20/06/2022**
- [5] ManojPandey and Daniel Zappala, The Effects of Mobility on Multicast Routing in Ad Hoc Networks. March, 2004 (Technical Report, UO-CIS-TR-2004-2).
- [6] Y. Amir, Y. Kim, C. Nita-Rotaru, G. Tsudik, On the performance of Group Key Agreement Protocols, ACM Transactions on Information and System Security (TISSEC), Volume 7 , Août 2004, pp. 457 – 488
- [7] Song, Tao, et al. "A cluster-based directional routing protocol in VANET."Communication Technology (ICCT), 2010 12th IEEE International Conference on.IEEE, 2010
- [8] <https://www.sciencedirect.com/topics/computer-science/vehicular-ad-hoc-network>
consulter le 28/02/2022
- [9] ADETUNDJA ADIGUN «gestion de l'anonymat et de la traçabilité dans les réseaux véhiculaires sans fil », mémoire de maîtrise, Département de mathématiques et informatique appliquées. Université du Québec à Trois-Rivières, Octobre 2013.
- [10] Q. Xu and D. Jiang, « Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum », Vehicular Technology

- Conference, 2003.VTC 2003-Spring. The 57th IEEE Semiannual, vol. 4, pp. 2451-2455, Apr. 2003.
- [11] J. Santa, A. F. Gómez-Skarmeta, and M. Sánchez-Artigas, « Architecture and Evaluation of a Unified V2V and V2I Communication System based on Cellular Networks », Computer Communications, vol. 31, no. 12, pp. 2850-2861, Jul. 2008.
- [12] M. JERBI, « Protocoles pour les Communications dans les Réseaux de Véhicules en environnement Urbain : Routage et GeoCast basés sur les intersections », Université D'EVRY VAL D'ESSONNE, thèse de doctorat, 2008.
- [13] M. BURMESTER, E. MAGKOS et V. CHRISSI. Strengthening Privacy Protection in VANETs. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications(WIMOB), 2008
- [14] https://www.researchgate.net/figure/Modes-of-VANET-communication-system_fig1_334535599
- [15] Sharef, Baraa T., Raed A. Alsaqour, and Mahamod Ismail. « Vehicular communication ad hoc routing protocols : A survey. » Journal of Network and Computer Applications 40 (2014) : 363-396
- [16] Macker Joseph, Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations, 1999.
- [17] M. JERBI, "Protocoles pour les communications dans les réseaux de véhicules en environnement urbain : Routage et GeoCast basés sur les intersections," UNIVERSITE D'EVRY VAL D'ESSONNE thèse de doctorat, 2008.
- [18] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehicular Mobility Simulation for VANETs," in Proceedings of the 40th Annual Simulation Symposium, Norfolk, VA , 2007, pp. 301-309
- [19] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehicular Mobility Simulation for VANETs," in Proceedings of the 40th Annual Simulation Symposium, Norfolk, VA , 2007, pp. 301-309. Print ISBN: 0-7695-2814-7
- [20] S. N. Pathak and U. Shrawankar, "Secured Communication in Real Time VANET," in Proceedings of the 2009 Second International Conference on Emerging Trends in Engineering & Technology, Nagpur , 2009, pp. 1151-1155.
- [21] J. CHOI, S. JUNG, Y. KIM et M. YOO. A Fast and Efficient Handover Authentication Achieving Conditional Privacy in V2I Networks. Proceedings of the 9th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking and Second Conference on Smart Spaces St. Petersburg, Russia, pages 291–300, 2009.

- [22] Q. XU et D. JIANG. Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum. In VehicularTechnologyConference, volume 4, pages 2451–2455, 2003.
- [23] Macker Joseph, Mobile ad hoc networking (MANET): Routingprotocol performance issues and evaluationconsiderations, 1999.)
- [24] Perkins Charles E. and BhagwatPravin, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, ACM SIGCOMM computer communication review, Vol. 24, No. 4, pp. 234-244, 1994.
- [25] Clausen Thomas and Jacquet Philippe, Optimized link state routing protocol (OLSR), No. RFC3626, 2003.
- [26] Sedrati M., Aouragh L., Guettala L. and Bilami A., Etude des Performances des Protocoles de Routagedans les Réseaux Mobiles Ad-Hoc, 4 thInternational Conference on Computer integrated manufacturing CIP, 2007.
- [27] Johnson David B., Maltz David A. and Hu Yih-Chun, The dynamic source routing protocol for mobile ad hoc networks, IETF MANET Working Group Internet Draft, 2004.
- [28] Perkins Charles, Belding-Royer Elizabeth and Das Samir, Ad hoc on-demand distance vector (AODV) routing, 2003. [20] Park Vincent D. and Corson M., A performance comparison of the temporallyordered routing algorithm and ideal link-state routing, Computers and Communications, ISCC'98, Proceedings Third IEEE Symposium on, IEEE, pp. 592-598, 1998
- [29] Wang Lan and Olariu Stephan, A two-zone hybrid routing protocol for mobile ad hoc networks, Parallel and Distributed Systems, IEEE Transactions on, Vol. 15, No 12, pp. 1105-1116, 2004.
- [30] Haas Zygmunt J., Pearlman Marc R. and Samar Prince, The zone routing protocol (ZRP) for ad hoc networks, draft-ietf-manet-zone-zrp-04. txt, 2002.
- [31] Sood M, Kanwar S. Clustering in MANET and VANET: a survey. In: 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA). Mumbai, India; 2014:375-380.
- [32] : IEEE 1609.4 SWG, “IEEE Standard for Wireless Access in Vehicular Environments (WAVE)”, Multichannel Operation. IEEE Computer Society,2010
- [33] Agerwal and MrMotwani. Survey of Clustering Algorithms for MANET. International Journal on Computer Science and Engineering, 1(2) :98–104, 2009
- [34] Elizabeth M. Belding-Royer. Hierarchical Routing in Ad Hoc Mobile Networks. Wireless Communications and Mobile Computing, 2(5) :515–32, 2002.

- [35] M. Re, L. Khoukhi, H. Labiod, J. Zhang and V. Veque, “A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs)”, IEEE/IFIP NOMS Workshop: International Workshop on Urban Mobility & Intelligent Transportation Systems (UMITS), 2016.
- [36] Anthony Ephremides, Jeffrey E. Wieselthier, and D.J. Baker. « A Design Concept for Reliable Mobile Radio Networks With Frequency Hopping Signaling ». Proceedings of the IEEE, 75(1) :56 – 73, January 1987. ISSN 0018-9219.
- [37] ChingChuan Chiang, Hsiao-Kuang Wu, Winston Liu, and Mario Gerla. « Routing In Clustered Multihop, Mobile Wireless Networks With Fading Channel », In Proceedings of the IEEE Singapore International Conference on Networks (SICON’97), pages 197–212, Singapore, August 19 1997. IEEE.
- [38] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit Disk Graphs. Discrete Mathematics, 86 :165–177, January 1991. ISSN 0012-365X. URL [http://dx.doi.org/10.1016/0012-365X\(90\)90358-O](http://dx.doi.org/10.1016/0012-365X(90)90358-O).
- [39] L. Blin, M. G. Potop-Butucaru, et S. Rovedakis. Self-stabilizing minimum degree spanning tree within one from the optimal degree. Journal of Parallel and Distributed Computing, pages 438 – 449, 2011.
- [40] E. Caron, A. Datta, B. Depardon, et L. Larmore. A self-stabilizing k-clustering algorithm for weighted graphs. J. Parallel Distrib. Comput., pages 1159–1173, 2010.
- [41] A. Datta, L. Larmore, et P. Vemula. Self-stabilizing leader election in optimal space. In Stabilization, Safety, and Security of Distributed Systems, pages 109–123. 2008
- [42] E. W Dijkstra. Self-stabilizing systems in spite of distributed control. Commun. ACM, pages 643–644, 1974
- [43] O. Flauzac, B. S. Hagggar, et F. Nolot. Self-stabilizing clustering algorithm for ad hoc networks. International Conference on Wireless and Mobile Communications, pages 24–29, 2009.
- [44] C. Johnenet L. H. Nguyen. Self-stabilizing weight-based clustering algorithm for ad hoc sensor networks. In Algorithmic Aspects of Wireless Sensor Networks, pages 83–94. 2006.
- [45] C. Johnenet L. H. Nguyen. Self-stabilizing construction of bounded size clusters. International Symposium on Parallel and Distributed Processing with Applications, pages 43–50, 2008
- [46] C. Johnen et L. H. Nguyen. Robust self-stabilizing weight-based clustering algorithm. Theoretical Computer Science, pages 581 – 594, 2009.

- [47] N. Mitton, A. Busson, et E. Fleury. Self-organization in large scale ad hoc networks.2004.
- [48] N. Mitton, E. Fleury, I. Guerin L., et S. Tixeuil. Self-stabilization in self-organized multihop wireless networks. In Proceedings of the Second International Workshop on Wireless Ad Hoc Networking - Volume 09, ICDCSW '05, pages 909–915, 2005
- [49] Korichi Ahmed. (2009). TCAO et Simulation : Vers une plate-forme d’analyse et de conception de systèmes de production orientée groupe. Thèse de Doctorat de L’Université de Batna. 15 Janvier 2009.
- [50]https://www.memoireonline.com/04/17/9779/m_Mise-en-place-dun-systeme-informatise-de-transfert-dargent-cas-de-go-sar110.html#:~:text=UML%2C%20c'est%20l',diff%C3%A9rente%20du%20projet%20%C3%A0%20traiter.
- [51] Ephremides (A.), Wieselthier (J.) et Baker (D.). – A design concept for reliable mobile radio networks with frequency hopping signaling. Proceedings of the IEEE, vol. 75, n1, janvier 1987, pp. 56–73
- [52] Guizani, B., Ayeb, B., Koukam, A. (2011), “Impact of Stability in Cluster Based Link State Routing Protocol for Self-Organizing Networks,” 7th ICWMC
- [53] Thèse de Doctorat de l’Université de Mostaganem présenté par Mme Saliha BENKERDAGH le 09 Décembre 2019.
- [54] M. Chatterjee, S. K. Das, and D. Turgut, “WCA: A weighted clustering algorithm for mobile ad hoc networks,” Journal of Cluster Computing, No. 5, 2002, pp. 193-204.
- [55] YOUNIS, Ossama et FAHMY, Sonia. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*, 2004, vol. 3, no 4, p. 366-379.