

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي



جامعة سعيدة د. مولاي الطاهر
كلية التكنولوجيا
قسم: الإعلام الآلي

Mémoire de Master

Spécialité : Réseaux Informatiques et Systèmes.

Thème

ÉTUDE DE LA QUALITÉ DE SERVICE QoS DANS LES
RÉSEAUX VANET.
Optimisation du routage dans VANETs.

Présenté par :

Hennoune Aissa
Bouras Marouane

Dirigé par :

Mme. Derkaoui Orkia .



Promotion 2021 - 2022

Contents

1	Les réseaux véhiculaires:caractéristiques générales	10
1.1	Introduction	10
1.2	Caractéristiques des Réseaux de Véhicules	10
1.2.1	Forte Mobilité	10
1.2.2	Environnements de routage et modèles de mobilité	11
1.2.3	Capacité de traitement et puissance de la batterie	11
1.2.4	Échange fréquent d'informations	11
1.2.5	Communication sans fil	11
1.2.6	Temps critique	11
1.2.7	Mobilité prévisible	11
1.3	Défis de sécurité dans les réseaux de véhicules	11
1.4	Composants d'un réseau VANET	11
1.4.1	OBU (On Board Unit)	12
1.4.2	RSU (Road Side Unit)	12
1.4.3	TA (Trusted Authority)	12
1.5	Architecture des VANETs	12
1.5.1	Communication véhicule à infrastructure (V2I)	13
1.5.2	Communication véhicule à véhicule (V2V)	13
1.5.3	Communication hybride (HVC)	13
1.6	Technologies de communication véhiculaire	13
1.6.1	Technologie à courte portée	13
1.6.2	Technologies a portée moyenne	14
1.6.3	Technologies à longue portée	15
1.7	Applications des réseaux de véhicules	15
1.7.1	Applications de sécurité	16
1.7.2	Applications de confort	16
1.7.3	Applications de gestion du trafic routier	16
1.8	Le routage dans les réseaux de véhicules	16
1.8.1	Définition	16
1.8.2	Types de messages	16
1.9	Conclusion	17
2	Routage dans les réseaux vanets	18
2.1	Introduction	18
2.2	Concepts De Base Du Routage	18
2.2.1	Le routage	18
2.2.2	Définition D'un Protocole De Routage	18
2.2.3	Objectif DU Routage	18
2.3	Les Protocoles De Routage Dans Les Réseaux VANETS :	18
2.3.1	Classification des protocoles de routage	19

2.3.2	Routage basé sur la topologie	20
2.3.3	Routage basé sur la position	20
2.3.4	Routage basé sur les clusters	20
2.3.5	Routage basé sur la diffusion	21
2.3.6	Routage à base Geocast	21
2.4	Conclusion	21
3	Les algorithmes de la Clusterisation	22
3.1	Introduction	22
3.2	L' Algorithme:K-Medoids	22
3.2.1	Introduction	22
3.2.2	Definition	23
3.2.3	Description	23
3.2.4	Conclusion	23
3.3	l' algorithme:K-means	24
3.3.1	Introduction	24
3.3.2	Definition	24
3.3.3	Description	25
3.3.4	Conclusion	25
4	Application des algorithmes Clustering pour routage dans réseau vanet	26
4.1	Introduction	26
4.2	les Protocoles de routage dans Un réseaux vanets	26
4.2.1	Le Protocole DSDV	26
4.2.2	Le Protocole GPSR	27
4.3	Algorithme K-means	28
4.3.1	Introduction	28
4.3.2	Les entrées pour appliquer algorithme kmeans	28
4.3.3	Affichage Les Centres	29
4.3.4	Les Sortees De Appliquer Algorithme kmeans	30
4.4	Algorithme K-medoids	30
4.4.1	Introduction	30
4.4.2	Les Entrées Pour Appliquer Algorithme K-medoids	30
4.4.3	Affichage Les Centres	31
4.4.4	Les Sortees De Appliquer Algorithme k-medoids	32
4.5	Conclusion	32
5	Résultats et Interpretation	33
5.1	Introduction	33
5.2	Les outils de l'implimentation et la language de programmation	33
5.2.1	pycharm	33
5.2.2	Python	33
5.2.3	Les Bibliotheques	34
5.3	La fiche technique de la machine utilisée	35
5.3.1	Specifications de l'appareil	35
5.3.2	Specification de Windows	35
5.4	L' Application	36
5.4.1	Coté De Conception:Interface	36
5.4.2	Coté de programmation	37
5.4.3	Interprétations	52
5.4.4	Statistiques	52

5.5 Conclusion 55

List of Figures

1.1	Types De Communication Dans Les Réseaux véhiculaire[1].	13
2.1	Classification des protocoles de routage dans les VANETs[2].	19
3.1	l'implimentation le algorithme k medoid	22
3.2	les etapes de l'algorithme k-medoids [5].	23
3.3	l'implimentation l'algorithme k-means [7].	24
3.4	les étapes de l'algorithme k-means [8].	25
3.5	les étapes de l'algorithme k-means [8].	25
4.1	GPSR protocol [11].	28
4.2	initialiser le nombre de cluster	28
4.3	les centres	29
4.4	les centres	29
4.5	La Distance Euclidien	29
4.6	La Distance Manhattan	30
4.7	LL 'implimentation Les Algorithmes De La Clustérisation	30
4.8	Les Centres	31
4.9	Les Centres	31
4.10	L'implimentation Les Algorithmes De La Clustérisation	32
5.1	Spécification de L'Appareil	35
5.2	Spécification De Windows	35
5.3	page d'accueil de l'application	36
5.4	visulation global avant Capture	37
5.5	après Capture	37
5.6	réclustérisation t=1	38
5.7	réclustérisation t=1 Cluster2	38
5.8	réclustérisation t=1 Cluster3	38
5.9	réclustérisation t=1 Cluster4	39
5.10	réclustérisation t=1 Cluster5	39
5.11	réclustérisation t=1 Cluster6	39
5.12	Réclustering a structure de réseau dans un T1	40
5.13	Réclustering a structure de réseau dans un T2	40
5.14	Réclustering a structure de réseau dans un T3	41
5.15	Réclustering a structure de réseau dans un T4	41
5.16	Réclustering a structure de réseau dans un T5	42
5.17	Capturer la structure de réseau dans un T0	42
5.18	Cluster 1	43
5.19	Cluster2	43
5.20	Cluster3	43
5.21	Cluster4	44

5.22 Cluster5 44
5.23 Cluster6 44
5.24 Cluster6 44
5.25 Réclustering la structure de Réseau dans un T1 45
5.26 Réclustering la structure de Réseau dans un T2 45
5.27 Réclustering la structure de Réseau dans un T3 46
5.28 Réclustering la structure de Réseau dans un T4 46
5.29 Réclustering la structure de Réseau dans un T5 47
5.30 Algorithme k-medoid :les statistiques de routage 53
5.31 Algorithmes K-means et k-medoid :les statistiques de routage 54
5.32 Algorithme K-means :les statistiques de routage 55

List of Tables

5.1	Cluster 1 EN T2	47
5.2	Cluster 1 EN T3	48
5.3	Mobilite Cluster 1 EN T2	48
5.4	Mobilite Cluster 1 EN T3	48
5.5	Avant Routage Les Message Dans Cluster 1 EN T2	49
5.6	Aprés Routage Les Message Dans Cluster 1 EN T2	49
5.7	Aprés Routage Les Message Dans Cluster 1 EN T2	49
5.8	Cluster 6 EN T2	50
5.9	Etat Routage De Cluster 6 EN T4	50
5.10	Mobilite Cluster 6 EN T4	50
5.11	Mobilite Cluster 6 EN T2	50
5.12	avant routage les message dans Cluster 6 EN T2	51
5.13	Aprés Routage Les Message Dans Cluster 6 EN T2	51
5.14	etat de messge Cluster 6 EN T4	52
5.15	avant routage :les message dans Cluster 6 EN T4	52

ملخص

(VANET شبكات Adhoc الخاصة بالمركبات) هي تقنية ناشئة جديدة تتكامل ميزات الجيل الجديد من الشبكات اللاسلكية للمركبات للهدف الرئيسي من هذا النوع من الشبكات هو تعميم المعلومات بأكثر الطرق فعالية بأكبر قدر ممكن من الكفاءة بين المركبات ؛ بين البنية التحتية والمركبات من أجل تحسين السلامة على الطريق وتوفير الراحة للسائقين والركاب ستند شبكات المركبات إلى البروتوكولات التي تضمن تبادل المعلومات بين المركبات التي تتميز بحركة عالية تحسين التواصل يتعلق الأمر بتحديد فعالية هذه البروتوكولات. في هذه المذكرة، نقدم تنفيذ خوارزميات التجميع عن طريق تحسين جودة الخدمة للتوجيه في الفانيهات. الكلمات الرئيسية: VANET التجميع ، التوجيه ، جودة الخدمة.

Abstract

VANET (Vehicular Adhoc Networks) is a new emerging technology that integrates features of the new generation of wireless networks for vehicles. The main objective of this type of network is to circulate information in the most efficient way, as efficiently as possible between vehicles; between infrastructure and vehicles in order to improve the road safety and provide comfort to drivers and passengers. Vehicular networks are based on protocols that ensure the exchange of information between vehicles that are characterized by high mobility; improving communication comes down to determining the effectiveness of these protocols.

In this memory, we present the implementation of clustering algorithms by improving the QoS of routing in vanets.

Keywords: VANET, clustering, routing, QoS.

Résumé

VANET (Vehicular Adhoc Networks) est une nouvelle technologie émergente qui intègre les fonctionnalités de la nouvelle génération de réseaux sans fil pour les véhicules. Le principal objectif de ce type de réseau est de faire circuler l'information de la manière la plus efficace possible entre les véhicules ; entre les infrastructures et les véhicules afin d'améliorer la sécurité routière et d'apporter le confort aux conducteurs et aux passagers.

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

Dans ce mémoire, nous présentons la mise en oeuvre des algorithmes de clustering par amélioration du QoS du routage dans les vanets.

Mots clés : VANET, clustering, routage, QoS

Introduction Generale

Les VANETs appartiennent à une classe des réseaux ad hoc mobiles (MANET) avec des caractéristiques spécifiques. Chaque véhicule est considéré comme un noeud équipé de capteurs et de caméras pour la collecte des données. A travers des logiciels dédiés, ces données seront traitées pour identifier les obstacles, les accidents, les stations- services, les signalisations routière, etc. Ensuite, ces informations extraites seront transmises d'un véhicule à un autre via des émetteurs sans fil pour former un réseau véhiculaire. Sachant que la portée de transmission de chaque véhicule est limitée à quelques centaines de mètres, une communication multi-sauts est nécessaire pour que les messages soient diffusés vers les zones éloignées.

Dans les VANETs, la grande mobilité des nœuds conduit à des changements fréquents dans la topologie du réseau. Cette mobilité est limitée par les types de routes et les règles de circulation. Sous ces conditions, les constructeurs automobiles et les chercheurs ont lancé plusieurs projets et recherches pour développer des systèmes de transport intelligents robustes et fiables. L'idée de base de ces solutions est de permettre aux véhicules l'échange des messages d'alerte sur les accidents et les conditions de circulation afin d'aider les conducteurs à prendre des décisions proactives. L'un des problèmes des VANET est que chaque véhicule communique avec tous ceux qui sont dans sa zone de couverture. Ceci entraîne une dégradation de la qualité de service (QoS) avec l'augmentation du nombre de véhicules. Un bon routage dans un réseau VANET fait partie de la QoS. Ce problème a fait l'objet de plusieurs études. Malgré la variété des approches proposées et le développement des technologies de communication, les algorithmes exacts ne peuvent pas résoudre le problème de routage dans les VANETs en un temps polynomial. Sur la base de cette remarque, nous proposons l'utilisation de métaheuristiques. Problématique : Le routage est l'opération consistant à diriger des données au sein d'un réseau. Le routage permet le transfert d'un message depuis l'émetteur vers le receveur. C'est une stratégie qui permet, à n'importe quel moment de trouver le meilleur chemin entre deux nœuds du réseau. Le Routage basé sur les clusters est une structure qui permet d'augmenter la QoS du Routage.

Dans le routage basé sur les clusters, les nœuds d'un réseau sont répartis en différents groupes géographiquement proches les uns des autres (Cluster). Dans chaque cluster, un nœud est élu pour assumer les fonctions essentielles dans le routage (tête de groupe). Il résume les informations de son cluster et les envoie aux têtes de cluster voisines à travers les passerelles. En effet, chaque cluster doit contenir trois types de nœuds "nœud ordinaire", " tête de cluster" et "passerelle". Dans notre projet de fin d'études, nous voulons proposer un mécanisme de routage optimal permettant d'acheminer un message de manière optimale entre une source et plusieurs zones de destination. On utilise des algorithmes de clustérisation « K-medoids» et « K-means» pour optimiser le problème de routage dans les réseaux véhiculaires.

Ce mémoire est structuré en cinq chapitres : Dans le premier chapitre, nous présentons une vue d'ensemble sur les réseaux véhiculaires Ad Hoc VANET, leurs caractéristiques et leurs applications. Dans le deuxième chapitre, nous donnons un état de l'art des protocoles qui ont été proposés pour assurer le routage dans les réseaux VANET. Nous décrivons leurs fonctionnements et les stratégies utilisées pour l'acheminement des données entre les différents nœuds mobiles. Le chapitre trois est consacré aux algorithmes utilisés pour réaliser un routage clusterisé. Le chapitre quatre présente la mise en œuvre des algorithmes sur un réseau VANET.. Le chapitre cinq présente l'interprétation et discussion des résultats obtenus et des comparaisons entre les algorithmes implémentés.. Nous terminons ce mémoire par une conclusion pour résumer les apports essentiels de notre travail et perspective de recherche.

Chapter 1

Les réseaux véhiculaires: caractéristiques générales

1.1 Introduction

VANET est la forme abrégée de « Vehicular Adhoc NETwork ». C'est une sous-classe du réseau mobile MANET dont les nœuds ne sont pas soumis à des limitations de stockage d'informations et de puissance de calcul. C'est un réseau de communication sans fil autonome et auto-organisé, où les nœuds s'impliquent en tant que serveurs et / ou clients pour échanger et partager des informations . Cependant, les VANETs sont différents des autres réseaux ad-hoc en raison de la mobilité élevée des nœuds, de la densité variable et de l'environnement de communication imprévisible. Les véhicules et les routes sont équipés de dispositifs de communication sans fil tels que les unités embarquées (OBUs) dans les véhicules et les nœuds stationnaires appelés unités de bord (RSUs) déployées le long des routes. A partir de ces unités, il existe trois scénarios de communication : véhicule à véhicule (V2V) (c'est-à-dire entre les OBUs), véhicule à route (V2R) (entre OBU et RSU) et hybride (HVC) . Les VANET sont considérés comme une partie importante des Systèmes de Transport Intelligents (STIs) pour assurer la sécurité routière, le confort de la conduite et la diffusion d'informations actualisées sur les routes. Par exemple, en cas d'accident, les véhicules envoient des messages d'avertissement aux conducteurs pour prendre les bonnes décisions avant d'entrer dans la zone du crash. De même, les conditions de circulation peuvent être communiquées pour faciliter la conduite en empruntant de nouveaux chemins en cas de congestion. Le présent chapitre présente les concepts liés aux réseaux ad hoc de véhicules. Il met l'accent sur les techniques de propagation de l'information dans les STIs. Ce chapitre se compose de dix sections ; la première introduit les caractéristiques et les défis liés aux VANETs. La seconde section présente les différents composants ainsi que les architectures mises en place dans ce type de réseau. La troisième section détaille les technologies de communication véhiculaire et les normes de standardisations reliées. Dans les sections qui suivent, on traite le problème de routage, ainsi que les tendances futures. La dernière section présente les principaux projets lancés par les constructeurs automobiles pour développer des systèmes d'assistance aux conducteurs[1].

1.2 Caractéristiques des Réseaux de Véhicules

1.2.1 Forte Mobilité

La vitesse des nœuds dans un réseau ad hoc de véhicules peut atteindre 140 km/h. Cette mobilité affecte négativement la connectivité et diminue la durée de vie des liens établies. Par conséquent, n'importe quel véhicule peut entrer ou quitter le réseau dans un très court délai[1].

1.2.2 Environnements de routage et modèles de mobilité

En raison de la grande vitesse des véhicules, la transition d'un environnement à un autre est inattendue. Chaque environnement diffère des autres par sa densité et la vitesse de ses noeuds. Pour cela, il est très difficile de trouver un protocole pratique à toutes les situations possibles[1].

1.2.3 Capacité de traitement et puissance de la batterie

Contrairement aux autres types de réseaux qui ont des problèmes d'alimentation et de stockage, les noeuds dans VANET n'ont aucune limitation en termes d'énergie et de mémoire. En roulant chaque véhicule produit une énergie électrique continue permettant une capacité de traitement élevée[1].

1.2.4 Échange fréquent d'informations

La nature ad hoc des réseaux véhiculaires motive les noeuds pour recueillir des informations auprès des unités de bord de la route et des autres véhicules. Ainsi l'échange d'informations entre les noeuds devient très fréquent[1].

1.2.5 Communication sans fil

Dans les VANETs, l'échange des données entre les noeuds est sans fil. Par conséquent, la sécurité des messages transmis est un défi majeur qui doit être pris en compte lors de la conception des protocoles de communication[1].

1.2.6 Temps critique

Les informations circulant dans un réseau véhiculaire doivent être transmises avec une limite de temps afin que les noeuds puissent prendre les décisions proactives et effectuent l'action en conséquence[1].

1.2.7 Mobilité prévisible

Contrairement aux autres réseaux mobiles, les véhicules suivent des mouvements restreints aux voies de circulation. Les informations routières sont souvent disponibles à partir d'une carte géographique et d'un système de positionnement comme le GPS (Global Positioning System)[1]. En effet, compte tenu de la direction, de la trajectoire de route et de la vitesse moyenne, on peut prédire la position future d'un véhicule[1].

1.3 Défis de sécurité dans les réseaux de véhicules

Les réseaux véhiculaires sont vulnérables à de nombreuses menaces. Le but de ces attaques est de créer un problème d'accès au système ou d'hameçonnage de certaines informations. En raison du caractère hautement sensible des informations diffusées, toutes les applications destinées au réseau véhiculaire doivent être protégées contre les manipulations malveillantes[1].

1.4 Composants d'un réseau VANET

Un réseau ad hoc de véhicules est constitué de trois composants de bases décrits ci-dessous[1] :

1.4.1 OBU (On Board Unit)

Ce sont des unités montées sur les véhicules intelligents, utilisées pour communiquer avec les unités latérales de la route ou avec d'autres véhicules. En plus d'un processeur de commande de ressource, chaque OBU contient des composants logiciels et matériels de hautes technologies (radar, GPS, capteurs, caméras et autres). Ces composants permettent le routage géographique, le contrôle de la congestion et la diffusion des informations sur le réseau. Ce sont des émetteurs-récepteurs responsables de la connexion du véhicule au réseau[1].

1.4.2 RSU (Road Side Unit)

Installées au bord des routes, elles peuvent être principalement des lampadaires, des feux de signalisation ou autres. Chaque RSU est équipée d'un périphérique réseau pour une communication à courte portée. Les principales fonctions des RSUs sont les suivantes :

- Étendre la couverture du réseau Ad Hoc pour permettre l'échange d'informations avec les OBUs communicants.

- Collecter et diffuser les informations.

- Aider les OBUs à se connecter à Internet[1].

1.4.3 TA (Trusted Authority)

Appelée autorité de confiance, c'est la composante sensée de la sécurité. Elle est responsable de l'enregistrement et de la gestion de toutes les entités du réseau (OBU et RSU). En effet, la mission de la TA est de connaître les vraies identités des véhicules et de les divulguer vers les entités du réseau. En entrant dans le réseau chaque véhicule doit posséder une copie des informations d'identification des voisins sous la forme d'un certificat. L'autorité de certification (AC) fournit des véhicules avec des certificats signés universellement. Elle permet aussi en cas de besoin de retirer les certificats précédemment signés. En cas de révocation, l'AC ajoute l'identification du certificat révoqué à une liste de révocation de certificats (LRC). Ensuite, l'autorité de certification annonce cette liste à tous les véhicules, en leur demandant de ne pas approuver le certificat révoqué. Les RSUs sont utilisées pour diffuser cette liste à toutes les unités mobiles du réseau[1].

1.5 Architecture des VANETs

VANET est un réseau de communication sans fil dans lequel tous les noeuds sont des véhicules qui se déplacent avec une grande vitesse. Généralement, cette communication est de trois types[1]

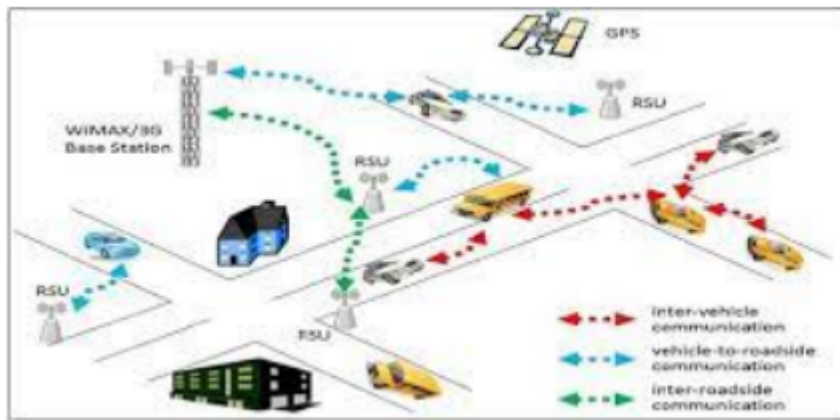


Figure 1.1: Types De Communication Dans Les Réseaux véhiculaire[1].

1.5.1 Communication véhicule à infrastructure (V2I)

L'échange de messages entre véhicules et infrastructures peut offrir aux usagers de nombreux services supplémentaires sur de grandes surfaces (accès Internet, informations de stationnement, localisation des restaurants, etc.). Dans ce mode de communication, bien que la connectivité soit améliorée même dans un environnement à faible densité comme l'autoroute, l'installation et l'entretien des infrastructures le long des routes est une tâche coûteuse qui prend beaucoup de temps[1].

1.5.2 Communication véhicule à véhicule (V2V)

Dans ce cas, il n'y a pas besoin d'infrastructure routière. La communication a lieu directement entre deux voitures appartenant à la même zone radio, ou indirectement via des relais qui transmettent les messages en utilisant un protocole multi-sauts. La communication V2V est adaptée aux applications nécessitant des communications à courte distance. Elle est liée à la conduite coopérative et à la sécurité routière. Ce type de communication convient aux environnements fortement connectés avec une grande densité[1].

1.5.3 Communication hybride (HVC)

La combinaison des deux types cités ci-dessus (V2V et V2I) aboutit à un modèle hybride. Dans ce cas, les véhicules communiquent entre eux pour échanger des informations de trafic et avec la station de base pour demander des données spécifiques[1].

1.6 Technologies de communication véhiculaire

Dans un réseau de véhicules, les noeuds utilisent différentes technologies d'accès sans fil pour améliorer la gestion et la surveillance du trafic. Ces technologies permettent aux passagers et aux conducteurs d'accéder aux services de divertissement et aux informations de sécurité. Elles peuvent être classées en trois groupes en fonction de leurs couvertures[1].

1.6.1 Technologie à courte portée

a. Infrarouge

C'est une technique d'accès sans fil qui utilise une lumière invisible pour transférer des données. Elle fonctionne dans une bande de fréquence de 300 GHz à 400 THz. Ce spectre est divisé en trois

sous-bandes : l'infrarouge lointain, le mi- infrarouge et le proche infrarouge. Il peut diffuser des données de 115 kbps à 4 Mbps. Parce que les signaux infrarouges sont affectés par des obstacles, ils sont utilisés pour les communications à courte distance. Cette technique est capable de transmettre des informations vocales, des données et des vidéos de manière plus sécurisée.

b. ZigBee

ZigBee (IEEE 802.15.4) fonctionne à des fréquences de 868 MHz, 900 MHz et 2,4 GHz. Cette technologie utilise des signaux radio de faible puissance pour transférer des données jusqu'à 100 mètre en utilisant la modulation par changement de phase en quadrature (QPSK). Comme ZigBee utilise une faible puissance et une faible latence, elle est largement utilisée dans les télécommandes, les PAN (personal area network), les réseaux de capteurs et la surveillance du transport. Cette technique est moins consommatrice d'énergie que les normes Bluetooth et Wi-Fi. Elle peut être utilisée pour les applications V2I et V2V. Cependant, elle ne convient pas aux applications d'info divertissement à bande passante intensive telles que la navigation sur le Web ou la VoIP.

c. Bluetooth

Le Bluetooth (IEEE 802.15.1) est basé sur les bandes radio ISM (Industrial Scientific and Medical). Il est utilisé pour transférer les données à un débit allant de 1 Mbps à 4 Mbps sur une distance de 10 mètre. Bluetooth fonctionne à une fréquence de 2,4 GHz et utilise la technique du spectre d'étalement de sauts de fréquence pour surmonter les interférences de signaux. Bien que Bluetooth 3 puisse fonctionner sur une bande de 6 GHz à 9 GHz, il utilise une bande de 2,4 GHz pour communiquer avec d'autres appareils. Aussi, le Bluetooth 4.0 utilise peu d'énergie lors d'une transmission, mais il n'est pas compatible avec les versions antérieures. Outre la création des réseaux personnels (PAN), Bluetooth peut être utilisé pour prendre en charge les applications V2V et V2I. Par rapport aux autres normes Wi-Fi, il nécessite une faible puissance, mais il ne peut pas être utilisé pour construire des applications de sécurité en raison du taux de transfert lent, de la zone de couverture courte et de la vulnérabilité aux interférences d'air. Il ne convient pas aux informations de divertissement telles que la VoIP et la navigation Web.

d. UWB (Ultra Wide Band)

Ultra large bande est une technologie sans fil utilisée pour transmettre de grandes quantités de données numériques avec une très faible puissance sur une courte distance. Cette technologie peut non seulement transporter une grande quantité de données sur une distance allant jusqu'à 100 mètre à très faible puissance (moins de 0,5 milliwatts), mais elle est aussi capable de transporter des signaux à travers des obstacles tels que les portes. Dans les VANETs, elle peut être utilisée pour éviter les collisions. L'UWB diffuse des impulsions numériques bien chronométrées sur un signal porteur dans un très large spectre en même temps. L'émetteur et le récepteur doivent bien coordonner pour envoyer et recevoir des impulsions avec une haute précision. L'ultra large bande a deux principaux types d'application : 1) Applications impliquant un radar, dans lesquelles le signal pénètre les surfaces proches mais réfléchit les surfaces les plus éloignées, ce qui permet de détecter les objets derrière les murs ou d'autres revêtements. 2) Transmission de voix et de données à l'aide d'impulsions numériques, permettant à un signal à très faible puissance et à coût relativement faible de transporter des informations à des débits très élevés dans une plage restreinte[1].

1.6.2 Technologies a portée moyenne

a. DSRC / WAVE

Le DSRC (Dedicated Short Range Communication) est une norme de communications de courte à moyenne portée conçue spécialement pour les communications V2V et V2I. DSRC fournit des transferts élevés de données avec une faible latence de communication dans les petites zones de communication. En 1999, la commission fédérale des communications du gouvernement des États-Unis a attribué un spectre de 75 MHz à 5,9 GHz à utiliser par le DSRC. Il a été approuvé pour être utilisé sur la base de la couche physique IEEE 802.11a et la couche MAC 802.11. Le spectre de DSRC est

divisé en sept canaux avec une largeur de 10 MHz chacun. Le premier est conçu pour diffuser les données de sécurité. Deux canaux sont utilisés pour une utilisation future. Les canaux restants sont utilisés pour les services de divertissement ou les applications de sécurité.

b. CALM Le concept CALM (Communication Architecture for Land Mobile) utilise un ensemble de protocoles, de procédures et d'actions de gestion. C'est une solution en couche développée pour fournir des communications de diffusion, de monodiffusion et de multidiffusion. Elle utilise multiples supports de télécommunications sans fil tel que 802.11, 802.11p, 802.15, 802.16e, 802.20, réseaux cellulaires 2G/3G/4G.

c. Normes WLAN / Wi-Fi Wi-Fi désigne certains types de réseaux locaux sans fil qui utilisent les spécifications de la famille 802.11 (802.11a, 802.11ac, 802.11b, 802.11e, 802.11g et 802.11n). Ces normes sont utilisées pour créer des réseaux ad-hoc en raison de leurs faibles coûts, et de leurs taux élevés de transfert de données. Un réseau Wi-Fi utilise des ondes radio pour transmettre sans fil des informations sur un réseau local, dont la portée peut être étendue par un prolongateur d'autonomie Wi-Fi. Les normes Wi-Fi sont utilisées pour créer un ensemble de services de base indépendant dans VANET. Généralement, le Wi-Fi nécessite plus de points d'accès et fournit des communications moins sécurisées et non fiables pour les VANETs que les systèmes cellulaires. Wi-Fi est largement utilisé dans les agences, les entreprises, les foyers et les écoles comme une alternative à un réseau local câblé. De nombreux hôtels, établissements et aéroports offrent un accès public aux réseaux Wi-Fi. Ces emplacements sont connus sous le nom de hots spots[1].

1.6.3 Technologies à longue portée

a. WiMax

WiMax (Worldwide Interoperability for Microwave Access) : c'est un standard de communication sans fil utilisé comme système de transmission et d'accès à Internet à haut débit. Il couvre une distance allant jusqu'à 50 km avec une vitesse de 70 Mbps. Il peut offrir des débits de données en liaison descendante allant jusqu'à 63 Mbps et des débits de données en liaison montante de 28 Mbps.

b. MBWA

L'IEEE 802.20 ou MBWA (Mobile Broadband Wireless Access) conçu pour fournir un accès Internet sans fil à des noeuds fortement mobiles. Il fournit une faible latence avec un débit de données pouvant atteindre 4,5 Mbps. Ce standard fonctionne sous licence 3.5 GHz et est optimisé pour prendre en charge la transmission rapide des paquets IP.

c. Microwave

Cette norme utilise une fréquence de 300 MHz à 300 GHz. Elle transmet jusqu'à 16 Gbps de données sur de longues distances. Elle fournit une large bande passante avec un taux de transmission élevé utilisé dans les fours à micro-ondes, les RADARs et les communications par satellite. En outre, elle est utilisée pour construire un réseau local sans fil qui couvre plusieurs villes. La principale limitation des micro-ondes est qu'elle nécessite une communication de ligne de vue (LoS).

d. Systèmes cellulaires (2 / 2,5 / 2,75 / 3G/4G)

La génération sans fil cellulaire (G) se réfère généralement à un changement de la nature du système, de la vitesse, de la technologie et de la fréquence. Chaque génération a des normes, des capacités, des techniques et de nouvelles caractéristiques qui la différencie de la précédente. Les réseaux cellulaires sans fil sont développés depuis les premiers systèmes de téléphonie mobile. Depuis l'introduction du premier système 1G en 1981, les connexions mobiles ont atteint des milliards, avec une nouvelle génération mobile apparaissant tous les dix ans environ[1].

1.7 Applications des réseaux de véhicules

Les applications véhiculaires peuvent être classées en trois catégories principales[1].

1.7.1 Applications de sécurité

Elles sont conçues pour avertir les passagers des routes de toute situation dangereuse. Par diffusion de messages d'alerte, les conducteurs auront une connaissance de l'état de la route et des véhicules voisins pour qu'ils prennent les mesures nécessaires pour éviter toute perte matérielle, humaine ou morale. Parmi ces applications nous citons les avertissements sur les conditions de la route, l'avertissement des collisions, l'assistance de changement de voie et de dépassement, etc[1].

1.7.2 Applications de confort

Désigne les applications de confort des conducteurs tel que l'accès à l'Internet, le chat inter-véhicules, le téléchargement des vidéos et les jeux en réseaux. Il y a aussi d'autres services d'assistance au conducteur comme la localisation des stations de service à proximité, des parkings, des restaurants, etc[1].

1.7.3 Applications de gestion du trafic routier

Ces applications sont axées sur l'amélioration du flux de circulation des véhicules et l'assistance routière pour améliorer les conditions de circulation. En équilibrant la circulation sur les routes, ces applications permettent de réduire les embouteillages et les risques d'accidents[1].

1.8 Le routage dans les réseaux de véhicules

Les réseaux véhiculaires vont être déployés dans les années à venir pour devenir la forme la plus pertinente de réseaux mobiles ad hoc. Ils fournissent une communication sans fil entre les véhicules et les équipements installés au bord des routes. En effet, les protocoles de routage MANET ne permettent pas de gérer efficacement cet environnement véhiculaire[1].

1.8.1 Définition

Le routage permet le transfert d'un message depuis l'émetteur vers le receveur. C'est une stratégie qui permet, à n'importe quel moment de trouver le meilleur chemin entre deux noeuds du réseau[1].

1.8.2 Types de messages

Quatre types de messages peuvent circuler dans un réseau véhiculaire.

Messages d'avertissement ou d'urgence : ils sont diffusés lorsqu'un évènement critique est détecté comme les accidents, le brouillard, le passage des véhicules d'ambulance, de la police ou d'incendie. Ces messages contiennent les paramètres des zones de retransmission et les coordonnées du lieu de l'accident. Les messages d'avertissements concernent les virages, la réparation des routes, la congestion du trafic, ...

Messages interpersonnels : ces messages concernent les profils des conducteurs et des passagers de véhicule tel que les messages d'envoi des e-mails ou de transaction financière.

Messages de sécurité et de routage : ce sont les messages informatifs, généralement générés à intervalle régulier. Ces messages nommés « beacons », contiennent la vitesse, la position, la direction ainsi que l'itinéraire du véhicule émetteur. Grâce à ces informations, les véhicules auront une vue de leurs voisinages. Ils peuvent donc prédire les accidents possibles et anticiper les congestions de la route pour avoir une communication sûre.

Messages d'information et de divertissement : ces messages sont liés aux services répartis sur les bords des routes, tel que les messages indiquant les stations d'essence, les points d'alimentation, les restaurants, les points d'intérêt à proximité...[1]

1.9 Conclusion

Une idée détaillée de la communication des véhicules a été donnée et d'une certaine manière si elle décrit la structure du réseau de véhicules et le mouvement à l'intérieur de celui-ci. Comprendre ce qui précède nous donne une vue plus précise d'un aspect important comme la direction dans ce type de réseau. Nous sommes amenés à rechercher une solution optimale, cela amène les spécialistes à réfléchir à des emplois afin d'obtenir. Sur cet idéal, on les appelle des méthodes approchées appelées métaheuristique.

Chapter 2

Routage dans les réseaux vanets

2.1 Introduction

VANET (Vehicular Ad hoc Network) est un réseau sans infrastructure fixe, constitué d'un ensemble de véhicules dénommés nœuds mobiles. Ces nœuds établissent entre eux une communication dite à un-saut lorsqu'ils sont à portée radio l'un de l'autre. Une communication dite multi-sauts peut également être établie entre deux nœuds distants (nœuds n'étant pas dans une même zone de transmission), réalisable grâce à la mise en place d'un chemin de routage reliant le nœud source au nœud destinataire et impliquant un ou des nœuds intermédiaires, dits aussi nœuds relais[2].

2.2 Concepts De Base Du Routage

2.2.1 Le routage

La fonction de routage est l'une des plus importantes dans les réseaux de communication, que ce soit dans les réseaux filaires ou dans les réseaux sans fil. Le routage sert à faire communiquer des nœuds distants. C'est un mécanisme qui permet l'établissement et le maintien de routes optimales entre les nœuds d'un réseau[2].

2.2.2 Définition D'un Protocole De Routage

Est un ensemble de règles de communication respectées par tous les systèmes interconnectés afin de permettre la liaison entre systèmes émetteurs et systèmes récepteurs[2].

2.2.3 Objectif DU Routage

Le routage dans les réseaux VANET est un problème très difficile à gérer et un axe de recherche pour beaucoup de chercheurs. Pour que les véhicules puissent communiquer entre eux, un protocole de routage doit être défini. En effet quand les terminaux ne sont pas à une portée de transmission radio directe, le routage est exigé pour établir la communication entre les véhicules[2].

2.3 Les Protocoles De Routage Dans Les Réseaux VANETS :

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

de routage se doit d'assurer un temps de connexion minimal tout en utilisant de façon optimale les ressources du réseau. Afin de présenter les principaux protocoles de routages dans les VANETs, nous avons choisi de commencer par faire une classification des différents protocoles existants[2].

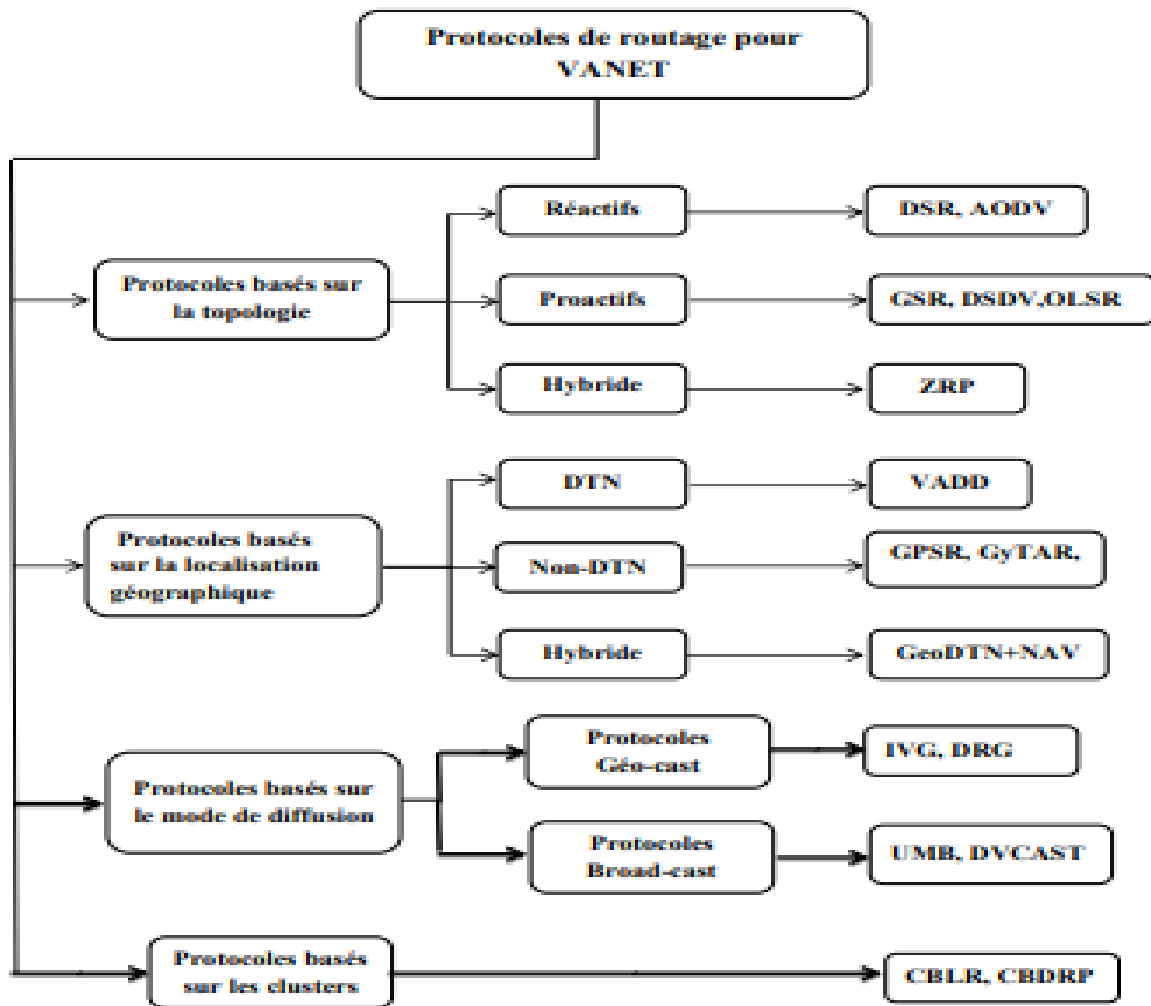


Figure 2.1: Classification des protocoles de routage dans les VANETs[2].

2.3.1 Classification des protocoles de routage

Comme écrit précédemment, le réseau VANET est un sous-groupe des réseaux MANET. Néanmoins, pour appliquer n'importe quel protocole de routage MANET à VANET, des modifications substantielles doivent être effectuées pour surmonter les problèmes d'inondation et de la grande mobilité des véhicules. Les protocoles spécifiques aux VANETs peuvent être classés de différentes manières suivant plusieurs critères. Par exemple, en fonction du mode de création et de maintenance des chemins, cinq classes de protocoles de routage sont représentées. Du fait de la grande mobilité des nœuds et de la diversité des environnements de routage (urbain, résidentiel, autoroute), les différents protocoles développés jusqu'à présent n'offrent pas une solution optimale en temps polynomial. Par conséquent, des efforts importants ont été faits pour trouver un système intelligent fiable qui assure un routage efficace des messages de la source vers la destination avec moins de problèmes de transmission[1].

2.3.2 Routage basé sur la topologie

Dans ce type de protocole, les noeuds ne disposent pas d'informations de localisation géographique concernant leurs positions ou la position de leurs voisins. La découverte et l'entretien des routes s'effectuent par des échanges périodiques de messages entre voisins. Cette classe englobe les protocoles réactifs, proactifs et hybrides[1].

a. Protocoles réactifs (ou à la demande)

Dans ce type de protocoles, une route est recherchée si un noeud a l'intention de transmettre une information à un autre. Le noeud source, qui cherche un chemin vers la destination, utilise l'inondation pour diffuser une requête dans le réseau. A la réception de cette requête, les noeuds intermédiaires envoient le chemin vers le noeud source, et sauvegardent la route dans la table envoyée. Une fois la destination atteinte, cette dernière utilise le chemin inverse pour envoyer la réponse. Par conséquent, un chemin full duplex va être établi entre la source et la destination. Lorsqu'un noeud intermédiaire possède déjà un chemin vers la destination, le temps de construction de route est considérablement réduit[1].

b. Protocoles proactifs

Les protocoles proactifs (appelés aussi approches pilotées par table) visent à maintenir à jour les informations de routage circulant dans un réseau. Chaque noeud garde une table contenant des informations de routage. Périodiquement, même s'il n'y a pas de données de trafic, tous les noeuds envoient des messages de routage pour créer ou mettre à jour les tables de routage. L'avantage majeur de ce type de protocoles est la réduction significative du temps de retard moyen par paquet. Les informations d'acheminement présentes dans la table de routage sont accessibles chaque fois que cela est nécessaire. Néanmoins, en maintenant ces informations à jour, les protocoles proactifs consomment une partie importante de la capacité du réseau. Cela rend de tels protocoles invalides aux réseaux ad hoc de forte mobilité. D'autre part, un autre gaspillage de la capacité du réseau se produit car la majorité des informations stockées ne seront pas utilisées avec la variation rapide de la topologie des réseaux. Ces protocoles nécessitent des messages de contrôle excessifs et consomment plus de bande passante. Dans les réseaux véhiculaires très dynamiques, les deux protocoles manquent d'efficacité, car les coûts de construction et de maintenance des routes augmentent lorsque la mobilité du réseau augmente. Plusieurs améliorations doivent être proposées pour ces protocoles afin de supporter les réseaux véhicules à forte mobilité[1].

c. Protocoles hybrides

Ils sont créés à partir de la combinaison des protocoles de routage proactifs et réactifs. Ils héritent des avantages de ces deux protocoles. En effet, le temps de retard lié au protocole de routage à la demande et l'énorme surcharge de routage de l'approche de routage proactif sont réduits[1].

2.3.3 Routage basé sur la position

Dans ce type de protocole, chaque noeud possède des informations concernant sa position géographique ainsi que celle de la destination. Les coordonnées des noeuds peuvent être obtenues en utilisant un système de géolocalisation tel que le GPS[1].

2.3.4 Routage basé sur les clusters

Dans ce cas, les noeuds d'un réseau sont répartis en différents groupes géographiquement proches les uns des autres (Cluster). Dans chaque cluster, un noeud est élu pour assumer les fonctions essentielles dans le routage (tête de groupe). Il résume les informations de son cluster et les envoie aux têtes de cluster voisines à travers les passerelles. En effet, chaque cluster doit contenir trois types de noeuds "noeud ordinaire", "tête de cluster" et "passerelle"[1].

2.3.5 Routage basé sur la diffusion

Dans ce type de protocole, la construction d'un chemin entre deux noeuds se fait en diffusant un message sur l'ensemble du réseau. Le message reçu pour la première fois par un noeud sera répété à tous les voisins, et ces voisins feront de même plus tard[1].

2.3.6 Routage à base Geocast

Dans ce type de protocole, la source délivre les messages uniquement à une zone géographique spécifique ; cette zone visée doit contenir la destination[1].

2.4 Conclusion

Nous avons choisi le Routage sur la base de clusters, car il donne une excellente structuration au réseau et flexibilité, lorsque les voitures movement, et c'est ce qui donne le dynamisme du réseau.

Chapter 3

Les algorithmes de la Clusterisation

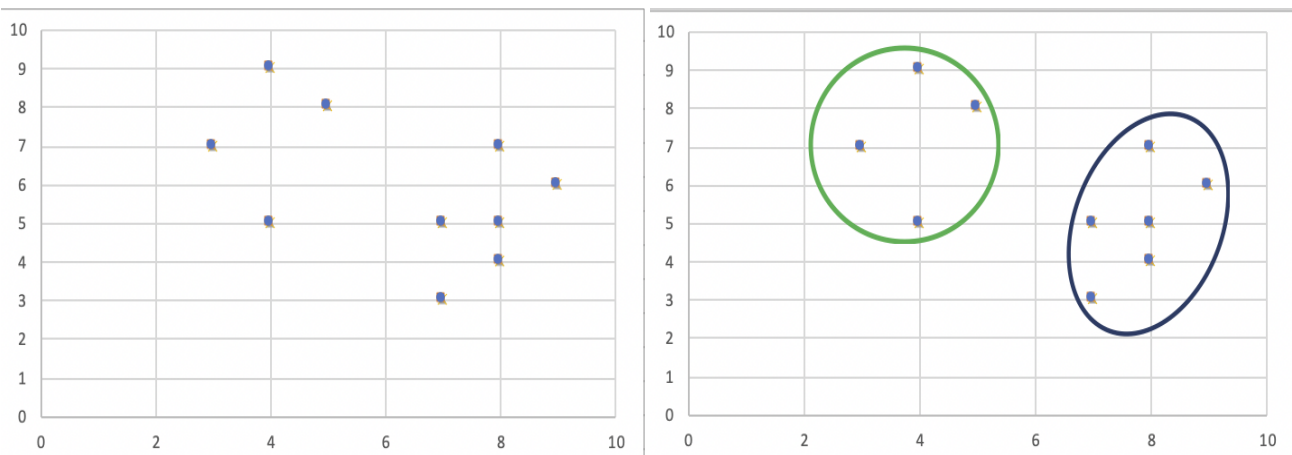
3.1 Introduction

Le clustering est une technique non supervisée dans laquelle l'ensemble de points de données similaires est regroupé pour former un cluster. Un cluster est dit bon si la similarité intra-cluster (les points de données dans le même cluster) est élevée et la similarité inter-cluster (les points de données à l'extérieur du cluster) est faible. Le clustering peut également être considéré comme une technique de compression de données dans laquelle les points de données d'un cluster peuvent être traités comme un groupe. Le clustering est également appelé segmentation des données car il partitionne les données de telle sorte qu'un groupe de points de données similaires forme un cluster.

3.2 L' Algorithme:K-Medoids

3.2.1 Introduction

k-medoids est un autre type d'algorithme de clustering qui peut être utilisé pour trouver des regroupements naturels dans un ensemble de données. Le clustering k-medoids est très similaire au clustering k-means, à quelques différences près. L'algorithme de clustering k-medoids a une fonction d'optimisation légèrement différente de celle des k-means. Dans cette section, nous allons étudier le clustering des k-médoïdes.



(a) avant l'implémentation de l'algorithme k-medoid [3]. (b) après l'implémentation de l'algorithme k-medoid [3].

Figure 3.1: l'implémentation de l'algorithme k-medoid

3.2.2 Définition

est lié à l'algorithme KMeans. Tandis que KMeans tente de minimiser la somme des carrés à l'intérieur du cluster, KMedoids tente de minimiser la somme des distances entre chaque point et le médoïde de son cluster. Le médoïde est un point de données (contrairement au centroïde) qui a la distance totale la plus faible par rapport aux autres membres de son cluster. L'utilisation d'un point de données pour représenter le centre de chaque cluster permet l'utilisation de n'importe quelle métrique de distance pour le clustering. Cela peut aussi être un avantage pratique, par exemple des algorithmes K-Medoids ont été utilisés pour la reconnaissance faciale pour lesquels le medoid est une photo typique de la personne à reconnaître alors que K-Means aurait obtenu une image floue qui mélangeait plusieurs photos de la personne reconnaître.

KMedoids peut être plus robuste au bruit et aux valeurs aberrantes que KMeans car il choisira l'un des membres du cluster comme médoïde tout KMeansen déplaçant le centre du cluster vers la valeur aberrante qui pourrait à son tour éloigner d'autres points du centre du cluster.

KMedoids est également différent de K-Médians, qui est analogue à KMeans sauf que la médiane de Manhattan est utilisée pour chaque centre de cluster au lieu du centroïde. K-Médians est robuste aux valeurs aberrantes, mais il est limité à la métrique Manhattan Distance et, comme KMeans, il ne garantit pas que le centre de chaque cluster sera membre de l'ensemble de données d'origine[4].

3.2.3 Description

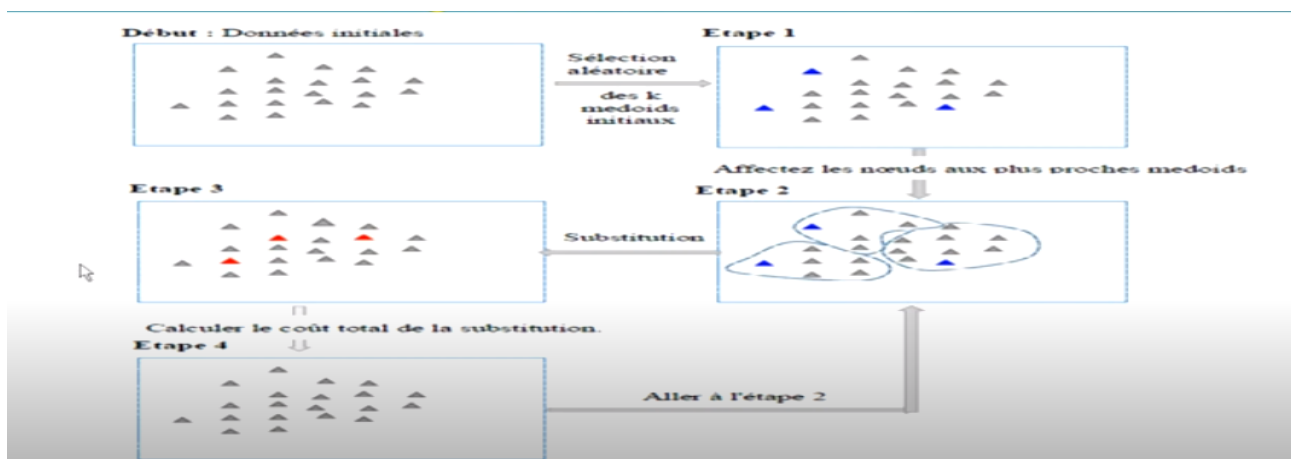


Figure 3.2: les etapes de l'algorithme k-medoids [5].

Initialiser : sélectionnez n clusters dans l'ensemble de données les médoïdes à l'aide d'une approche heuristique, aléatoire ou k-médoïdes.

Étape d'affectation : attribuez chaque élément du jeu de données au médoïde le plus proche.

Étape de mise à jour : identifiez le nouveau médoïde de chaque cluster.

Répétez l'étape d'affectation et de mise à jour pendant que les médoïdes continuent de changer ou que le nombre maximal d'itérations max-iterest atteint.

La complexité d'exécution de l'algorithme PAM original par itération de $O(k*(n-k)**2)$ [6].

k:nb de clusters

n:nb de objet

3.2.4 Conclusion

k medoids C'est l'un des meilleurs algorithmes de clustering et l'un des plus intégrés et utilisés, il donne des résultats idéaux au niveau de similarité des données, et c'est dû, mais il répète les calculs au sein de boucles pour donner des résultats idéaux (groupes).

3.3 l'algorithme:K-means

3.3.1 Introduction

Le clustering K-means est l'un des algorithmes de machine learning non supervisés les plus facile à comprendre et à utiliser. Aujourd'hui cet algorithme est l'un des plus répandus. D'habitude, les algorithmes non supervisés font des inférences à partir de jeux de données en se servant uniquement des vecteurs d'entrée sans faire référence à des résultats connus ou étiquetés.

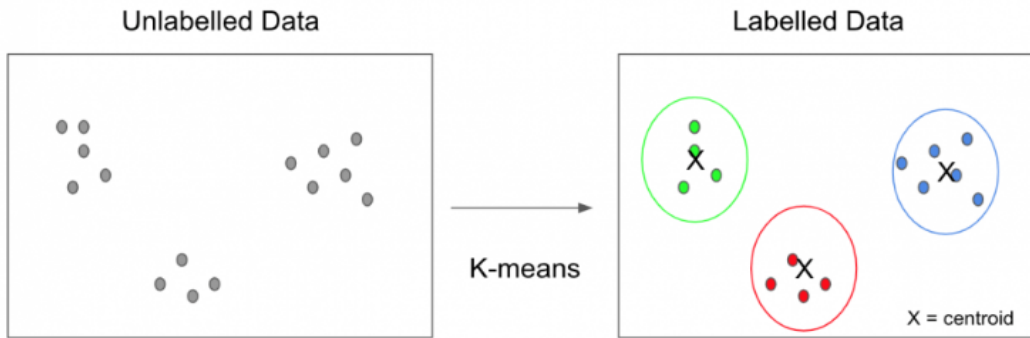


Figure 3.3: l'implimentation l'algorithme k-means [7].

3.3.2 Definition

Le clustering K-means est l'un des algorithmes d'apprentissage automatique non supervisé les plus simples et les plus populaires. En règle générale, les algorithmes non supervisés font des inférences à partir d'ensembles de données en utilisant uniquement des vecteurs d'entrée sans faire référence à des résultats connus ou étiquetés. AndreyBu, qui a plus de 5 ans d'expérience en apprentissage automatique et enseigne actuellement ses compétences aux gens, déclare que "l'objectif de K-means est simple : regrouper des points de données similaires et découvrir des modèles sous-jacents. Pour atteindre cet objectif, K-means recherche un nombre fixe (k) de clusters dans un ensemble de données. Un cluster fait référence à une collection de points de données agrégés en raison de certaines similitudes. Vous allez définir un nombre cible k , qui fait référence au nombre de centroïdes dont vous avez besoin dans l'ensemble de données. Un centroïde est l'emplacement imaginaire ou réel représentant le centre du cluster. Chaque point de données est attribué à chacun des clusters en réduisant la somme des carrés dans le cluster. En d'autres termes, l'algorithme K-means identifie k nombre de centroïdes, puis alloue chaque point de données au cluster le plus proche, tout en gardant les centroïdes aussi petits que possible. La « moyenne » dans les K-moyennes fait référence à la moyenne des données ; c'est-à-dire trouver le centroïde.

Principe algorithmique

Algorithme K-means

Entrée :

- K le nombre de cluster à former
- Le Training Set (matrice de données)

DEBUT

Choisir aléatoirement K points (une ligne de la matrice de données). Ces points sont les centres des clusters (nommé centroïde).

REPETER

Affecter chaque point (élément de la matrice de donnée) au groupe dont il est le plus proche au son centre

Recalculer le centre de chaque cluster et modifier le centroïde

JUSQU'À CONVERGENCE

OU (stabilisation de l'inertie totale de la population)

FIN ALGORITHME

Figure 3.4: les étapes de l'algorithme k-means [8].

3.3.3 Description

Pour traiter les données d'apprentissage, l'algorithme K-means dans l'exploration de données commence par un premier groupe de centroïdes sélectionnés au hasard, qui sont utilisés comme points de départ pour chaque cluster, puis effectue des calculs itératifs (répétitifs) pour optimiser les positions des centroïdes. Il interrompt la création et l'optimisation des clusters lorsque : Les centroïdes se sont stabilisés - il n'y a pas de changement dans leurs valeurs car le regroupement a réussi.

Le nombre d'itérations défini a été atteint.

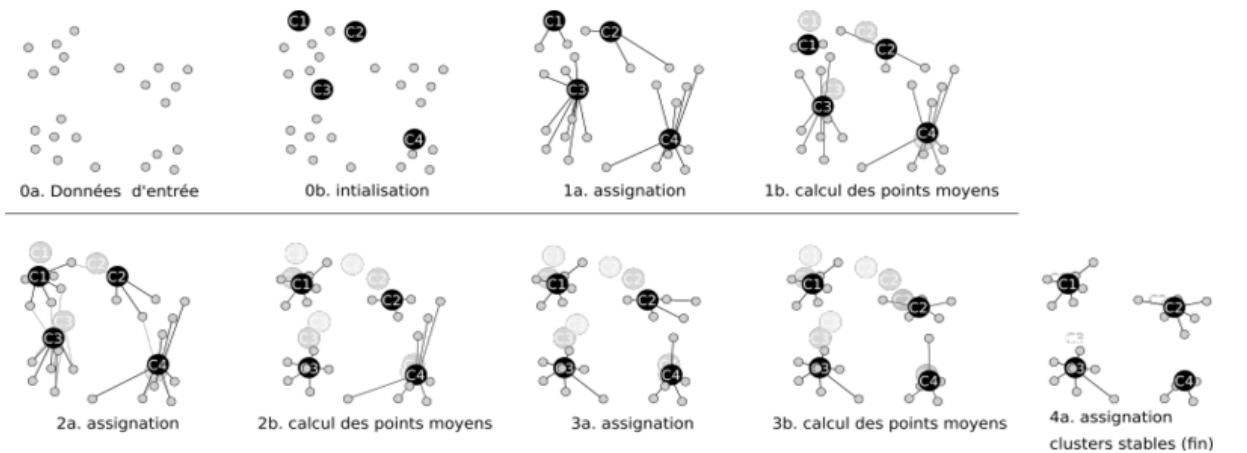


Figure 3.5: les étapes de l'algorithme k-means [8].

3.3.4 Conclusion

La classification K-means est une technique largement utilisée pour l'analyse par clusters de données. Cet algorithme est simple à comprendre. En outre, il fournit des résultats d'entraînement rapidement. Cependant, ses performances ne sont généralement pas aussi compétitives que celles des autres techniques de classification sophistiquées, car de légères variations dans les données pourraient entraîner une variance des résultats élevée. De plus, les clusters sont supposées être sphériques et de taille égale, ce qui peut réduire la précision des résultats Python en clusters à moyennes moyennes. Quelle est votre expérience avec le regroupement K-signifie en apprentissage automatique?

Chapter 4

Application des algorithmes Clustering pour routage dans réseau vanet

4.1 Introduction

nous avons utiliser deux algorithmes pour reaiser le concept de clusteriasation dans le réseaux vanet ,car ces algorithmes ils donnent milleur regroupement de voitures une quelque soit T est une relation entre temp et la mobilité,la performance des algorithmes de clusterisation donnent le dynamique dans le réseaux vanet et cette dynamique important car tout points dans cette réseaux elle changne par le critere T.

4.2 les Protocoles de routage dans Un réseaux vanets

4.2.1 Le Protocole DSDV

4.2.1.1 Definition

Destination-Sequenced Distance-Vector Routing (DSDV) est un schéma de routage piloté par table pour les réseaux mobiles ad hoc basé sur l' algorithme Bellman-Ford . Il a été développé par C. Perkins et P.Bhagwat en 1994. La principale contribution de l'algorithme était de résoudre le problème de la boucle de routage . Chaque entrée de la table de routage contient un numéro de séquence, les numéros de séquence sont généralement pairs si un lien est présent ; sinon, un nombre impair est utilisé. Le numéro est généré par la destination et l'émetteur doit envoyer la prochaine mise à jour avec ce numéro. Les informations de routage sont distribuées entre les nœuds en envoyant rarement des vidages complets et des mises à jour incrémentielles plus petites plus fréquemment [9].

4.2.1.2 Avantages

La disponibilité des chemins vers toutes les destinations du réseau montre toujours que moins de retard est nécessaire dans le processus de configuration du chemin.

La méthode de mise à jour incrémentielle avec des étiquettes de numéro de séquence, marque les protocoles de réseau câblé existants adaptables aux réseaux sans fil Ad-hoc. Par conséquent, tous les protocoles de réseau câblé disponibles peuvent être utiles aux réseaux sans

fil ad hoc avec moins de modifications[9].

4.2.1.3 Inconvénients

DSDV nécessite une mise à jour régulière de ses tables de routage, ce qui consomme de l'énergie de la batterie et une petite quantité de bande passante même lorsque le réseau est inactif.

Chaque fois que la topologie du réseau change, un nouveau numéro de séquence est nécessaire avant que le réseau ne re-converge ; ainsi, DSDV n'est pas adapté aux réseaux hautement dynamiques ou à grande échelle. (Comme dans tous les protocoles à vecteur de distance, cela ne perturbe pas le trafic dans les régions du réseau qui ne sont pas concernées par le changement de topologie[9].

4.2.2 Le Protocole GPSR

4.2.2.1 Définition

Dans les réseaux sans fil composés de nombreuses stations mobiles, le problème de routage consistant à trouver des chemins d'une source de trafic à une destination de trafic via une série de nœuds de transfert intermédiaires est particulièrement difficile. Lorsque les nœuds se déplacent, la topologie du réseau peut changer rapidement. De tels réseaux nécessitent un algorithme de routage réactif qui trouve rapidement des routes valides lorsque la topologie change et que d'anciennes routes se rompent. Pourtant, la capacité limitée du canal réseau exige des algorithmes et des protocoles de routage efficaces, qui ne conduisent pas le réseau dans un état congestionné lorsqu'ils apprennent de nouvelles routes. La tension entre ces deux objectifs, la réactivité et l'efficacité de la bande passante, est l'essence même du problème de routage mobile.

Greedy Perimeter Stateless Routing, GPSR, est un protocole de routage réactif et efficace pour les réseaux mobiles sans fil. Contrairement aux algorithmes de routage établis qui l'ont précédé, qui utilisent des notions théoriques de graphes de chemins les plus courts et d'accessibilité transitive pour trouver des itinéraires, le GPSR exploite la correspondance entre la position géographique et la connectivité dans un réseau sans fil, en utilisant les positions des nœuds pour prendre des décisions de transfert de paquets. GPSR utilise le transfert glouton pour transférer les paquets vers des nœuds qui sont toujours progressivement plus proches de la destination. Dans les régions du réseau où un tel chemin gourmand n'existe pas (c'est-à-dire que le seul chemin nécessite que l'on s'éloigne temporairement de la destination), GPSR récupère en transmettant en mode périmètre, dans lequel un paquet traverse successivement des faces plus rapprochées d'un sous-graphe planaire du graphe complet de connectivité du réseau, jusqu'à atteindre un nœud plus proche de la destination, où l'acheminement glouton reprend.

Le GPSR permettra la construction de réseaux qui ne peuvent pas évoluer en utilisant des algorithmes de routage antérieurs pour les réseaux filaires et sans fil. Ces classes de réseaux comprennent :

- Réseaux sur les toits : déploiement fixe et dense d'un grand nombre de nœuds

- Réseaux ad hoc : mobiles, densité variable, pas d'infrastructure fixe

- Réseaux de capteurs : mobile, densité potentiellement élevée, grand nombre de nœuds, ressources par nœud appauvries

- Réseaux véhiculaires : mobiles, sans contrainte de puissance, densité très variable

Nous étendons GPSR :

- Approvisionnement géographique : Nous utilisons la redirection géographique via un waypoint qui n'est pas sur le chemin trouvé par le GPSR naïf pour répartir la charge sur le réseau. Cette approche est prometteuse car sur un réseau sans fil, position et capacité sont corrélées ; la répartition géographique de la charge tire parti de la réutilisation spatiale et réduit la charge moyenne dans les régions où le trafic est concentré.

- Obstacles : nous étudions le comportement du GPSR en présence d'obstacles à la propagation radio, qui introduisent le risque que le sous-graphe planaire utilisé par le mode périmétrique du GPSR ne soit pas connecté. Nous étudions des algorithmes déterministes et randomisés pour récupérer de telles déconnexions lorsqu'elles se produisent. Nous prévoyons de construire de nouveaux systèmes de réseau sans fil dans les catégories ci-dessus qui utilisent le GPSR. Plus d'informations sur les systèmes que nous construisons apparaîtront sur cette page dans un proche avenir [10].

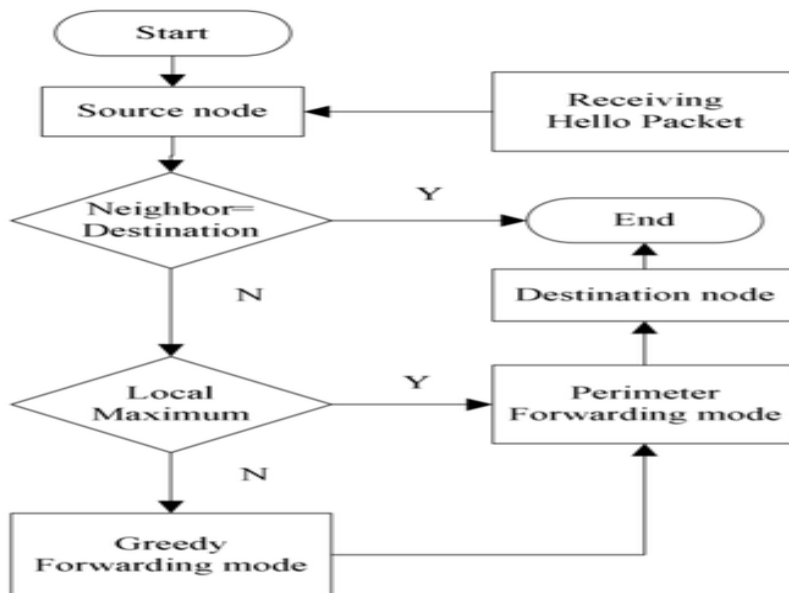


Figure 4.1: GPSR protocol [11].

4.3 Algorithm K-means

4.3.1 Introduction

nous allons déjà vu comment l'algorithme k-means travaille en côté abstrait, nous allons maintenant découvrir comment k-means donne la dynamique dans les réseaux VANET, la dynamique est toujours concernée avec le T, alors chaque voiture dans un réseau va collecter avec des autres voitures similaires, le critère de similarité est un concept général pour créer les réseaux VANET à partir de l'algorithme k-means. Ce critère détermine ou calcul par utilise deux méthodes plus utilisées sont : la distance Manhattan, distance euclidienne. nous appliquons ces méthodes dans notre application avec des données réelles.

4.3.2 Les entrées pour appliquer l'algorithme k-means

4.3.2.1 la liste des voitures

nous avons 100 voitures pour appliquer k-means et collecter 100 voitures dans 6 clusters,

4.3.2.2 Comment ca faire cette?

4.3.2.3 Déterminer les variables pour appliquer l'algorithme K-means

1-initialiser le nombre de cluster $k=6$.

```

Centroids = (df.sample(n=k))
print(Centroids)

```

Figure 4.2: initialiser le nombre de cluster

2-choisir les centres (le nombre des centres égale le nombre de cluster k)

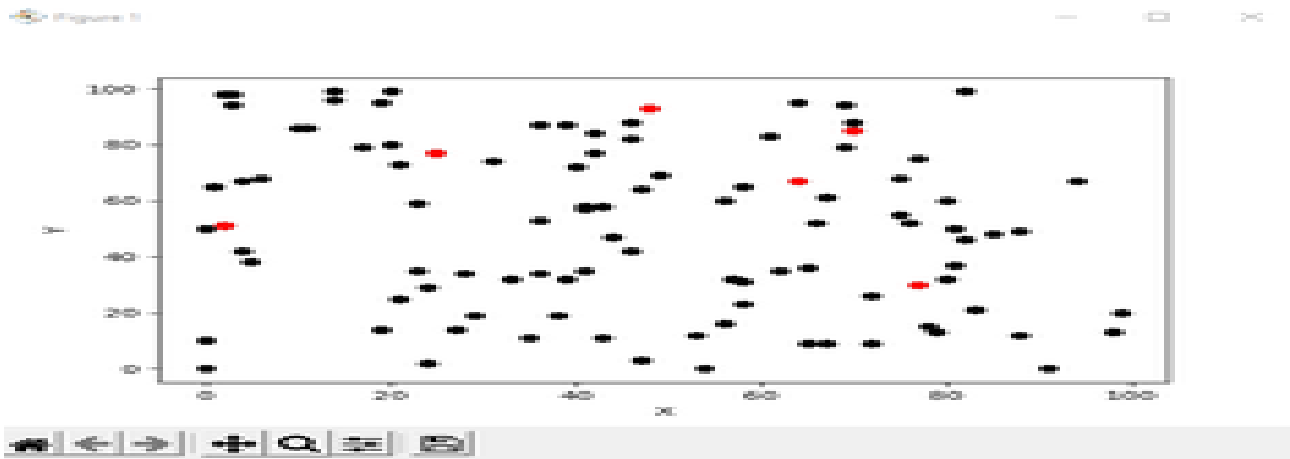


Figure 4.3: les centres

	X	Y
11	25	77
74	2	51
99	70	85
80	77	30
1	64	67
56	48	93

Figure 4.4: les centres

4.3.3 Affichage Les Centres

4.3.3.1 1:La Distance Euclidien

est une distance entre deux point ,chaque point a deux dimension ,c'est a dire chaque voiture a deux dimensions nous les appellons voiture(x,y).

$$d((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2}$$

Figure 4.5: La Distance Euclidien

4.3.3.2 1:La Distance Manhattan

est une distance entre deux point ,chaque point a deux dimension ,c'est a dire chaque voiture a deux dimensions nous les appellons voiture(x,y).

$$\text{Manhattan Distance} = \sum_{i=1}^n |p_i - q_i|$$

Figure 4.6: La Distance Manhattan

4.3.4 Les Sortees De Appliquer Algorithme kmeans

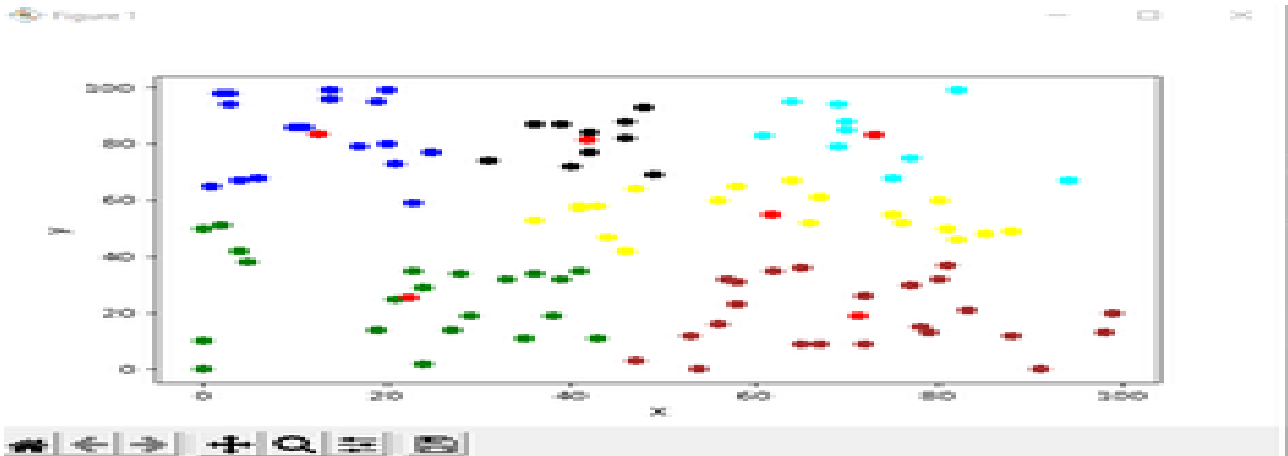


Figure 4.7: LL 'implimentation Les Algorithmes De La Clustérisation

4.4 Algorithme K-medoids

4.4.1 Introduction

nous allons déjà vu comment l'algorithme k-medoids travaille en côté abstrait, nous allons maintenant découvrir comment kmeans donne la dynamique dans les réseaux véhiculaires, la dynamique est toujours concernée avec le T, alors chaque voiture dans un réseau va collecter avec des autres voitures similaires, le critère de similarité est un concept général pour créer les réseaux véhiculaires à partir de l'algorithme k-medoids. ce critère détermine ou calculé par utilise deux méthodes plus utilisées sont : la distance manhattan, distance euclidienne. nous appliquons ces méthodes dans notre application avec des données réelles.

4.4.2 Les Entrées Pour Appliquer Algorithme K-medoids

4.4.2.1 La Liste Des Voitures

nous avons 100 voitures pour appliquer K-medoids et collecter 100 voitures dans 6 clusters,

4.4.2.2 Comment Ca Faire Cette?

déterminer les variables pour appliquer l'algorithme K-medoids 1-initialiser le nombre de clusters $k=6$.

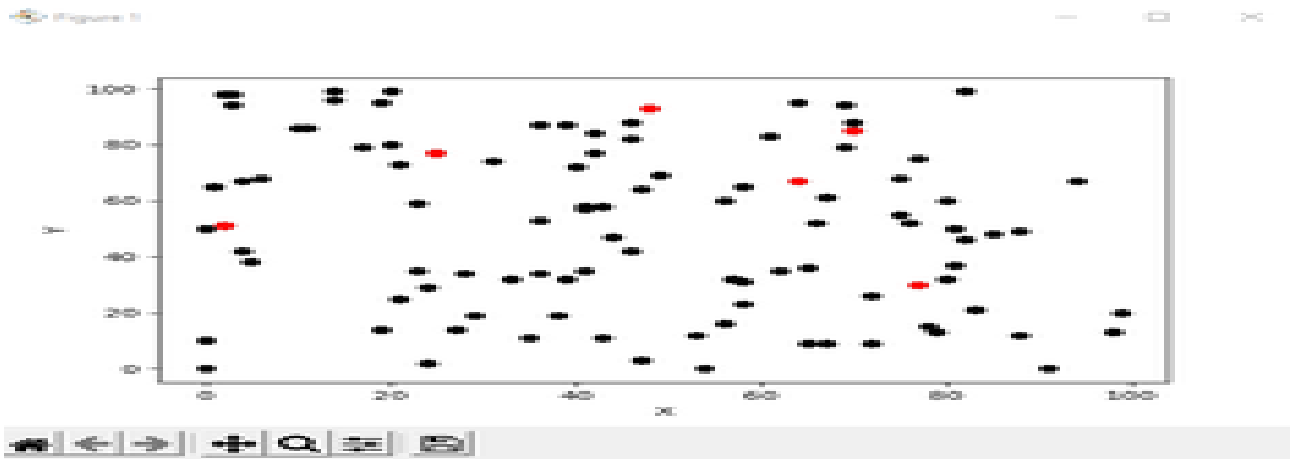


Figure 4.8: Les Centres

	X	Y
11	25	77
74	2	51
99	70	85
80	77	30
1	64	67
56	48	93

Figure 4.9: Les Centres

2-choisir les centres (le nombre des centres egale le nombre de cluster k)

4.4.3 Affichage Les Centres

4.4.3.1 1:La Distance Euclidien

est une distance entre deux point ,chaque point a deux dimension ,c'est a dire chaque voiture a deux dimensions nous les apellons voiture(x,y)

4.4.3.2 2:La Distance Manhattan

est une distance entre deux point ,chaque point a deux dimension ,c'est a dire chaque voiture a deux dimensions nous les apellons voiture(x,y)

4.4.4 Les Sortees De Appliquer Algorithme k-medoids

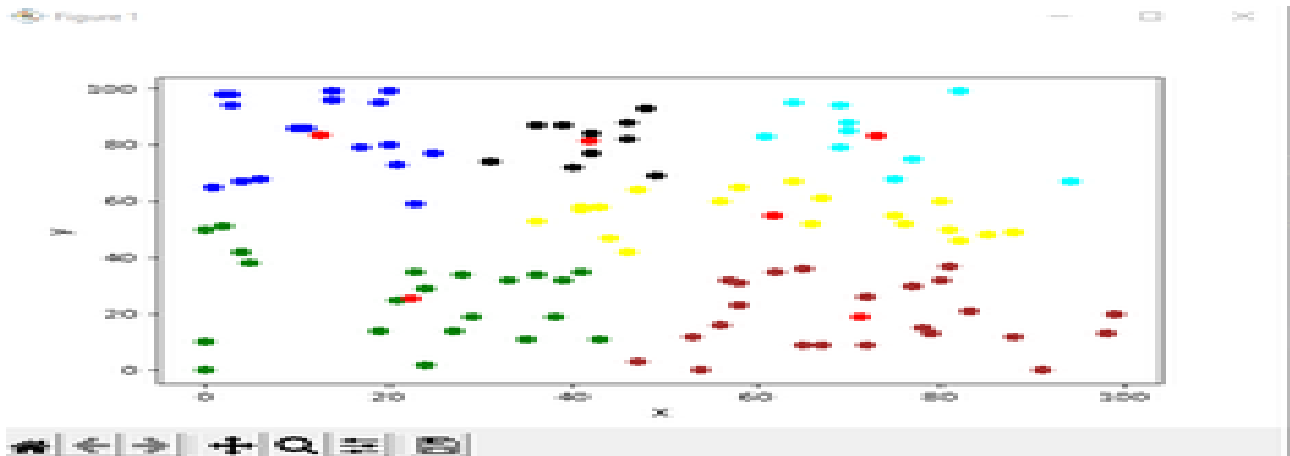


Figure 4.10: L'implimentation Les Algorithmes De La Clustérisation

4.5 Conclusion

Nous avons choisi les algorithmes k-means et k-medoids pour réaliser un réseau basé sur des clusters, en tenant compte de la grande différence entre eux dans le code et leur fonctionnement. Nous avons traité au chapitre 3. Quant au niveau de comparaison entre eux, nous en discuterons dans le dernier chapitre.

Chapter 5

Résultats et Interpretation

5.1 Introduction

Dans ce chapitre, nous expliquerons les outils et langage de programmation que nous avons choisi pour créer notre application, et nous fournirons également une explication complète du fonctionnement de l'application, ainsi que les spécifications de l'appareil qui exécutera notre application.

5.2 Les outils de l'implimentation et la langage de programmation

5.2.1 pycharm

5.2.1.1 Definition

PyCharm est un environnement de développement intégré utilisé pour la programmation informatique, en particulier la programmation en langage Python . Produit par la société tchèque JetBrains le logiciel fournit une analyse de code, un détecteur d'erreur graphique, un outil de test unitaire intégré, une intégration avec des systèmes de contrôle d'audit , une prise en charge du développement Web Django et un travail sur les applications de science des données, version PyCharm Community Edition 2021.3 .[12]

5.2.2 Python

5.2.2.1 Definition

Python est un langage de programmation généraliste interprété de haut niveau . Sa philosophie de conception met l'accent sur la lisibilité du code grâce à l'utilisation d' une indentation significative . Python est typé dynamiquement et ramasse les ordures . Il prend en charge plusieurs paradigmes de programmation , y compris la programmation structurée (en particulier procédurale), orientée objet et fonctionnelle . Il est souvent décrit comme un langage "piles incluses" en raison de sa bibliothèque standard complète . Guido van Rossum a commencé à travailler sur Python à la fin des années 1980 en tant que successeur du langage de programmation ABC et l'a publié pour la première fois en 1991 sous le nom de Python 0.9.0. Python 2.0 est sorti en 2000 et a introduit de nouvelles fonctionnalités telles que la compréhension des listes , la collecte des ordures à détection de cycle , le comptage des références et la prise en charge d' Unicode . Python 3.0, sorti en 2008, était une révision majeure qui n'est pas complètement rétrocompatible avec les versions antérieures, version 3.9. Python se classe régulièrement parmi les langages de programmation les plus populaires.[13]

5.2.3 Les Bibliothèques

5.2.3.1 Tkinter

Tkinter (de l'anglais Tool kit interface) est la bibliothèque graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques. Elle vient d'une adaptation de la bibliothèque graphique Tk écrite pour Tcl.[14]

5.2.3.2 Math

Ce module permet d'accéder aux fonctions mathématiques définies par la norme C. Ces fonctions ne peuvent pas être utilisées avec des nombres complexes ; utilisez les fonctions du même nom du module cmath si vous avez besoin de prendre en charge les nombres complexes. La distinction entre les fonctions qui prennent en charge les nombres complexes et celles qui ne le font pas est faite puisque la plupart des utilisateurs ne veulent pas apprendre autant de mathématiques que nécessaire pour comprendre les nombres complexes. La réception d'une exception au lieu d'un résultat complexe permet une détection plus précoce du nombre complexe inattendu utilisé comme paramètre, de sorte que le programmeur puisse déterminer comment et pourquoi il a été généré en premier lieu.[15]

5.2.3.3 Numpy

NumPy est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux masqués et des matrices) et un assortiment de routines pour des opérations rapides sur des tableaux, y compris mathématiques, logiques, manipulation de forme, tri, sélection, E/S, transformées de Fourier discrètes, algèbre linéaire de base, opérations statistiques de base, simulation aléatoire et bien plus encore. Au cœur du package NumPy, se trouve l'objet ndarray. Cela encapsule des tableaux à n dimensions de types de données homogènes, de nombreuses opérations étant effectuées dans du code compilé pour les performances.[16]

5.2.3.4 Random

Le module Python Random est un module intégré de Python qui est utilisé pour générer des nombres aléatoires. Ce sont des nombres pseudo-aléatoires, ce qui signifie qu'ils ne sont pas vraiment aléatoires. Ce module peut être utilisé pour effectuer des actions aléatoires telles que la génération de nombres aléatoires, l'impression aléatoire d'une valeur pour une liste ou une chaîne, etc.[17]

5.2.3.5 Pandas

pandas est un package Python fournissant des structures de données rapides, flexibles et expressives conçues pour rendre le travail avec des données « relationnelles » ou « étiquetées » à la fois simple et intuitif. Il vise à être le bloc de construction fondamental de haut niveau pour effectuer une analyse pratique des données du monde réel en Python. De plus, il a pour objectif plus large de devenir l'outil d'analyse/manipulation de données open source le plus puissant et le plus flexible disponible dans n'importe quelle langue.[18]

5.3 La fiche technique de la machine utilisée

5.3.1 Specifications de l'appareil

Spécifications de l'appareil

Nom de l'appareil	DESKTOP-BUP897J
Processeur	AMD A4-9120 RADEON R3, 4 COMPUTE CORES 2C+2G 2.20 GHz
Mémoire RAM installée	4.00 Go (3.89 Go utilisable)
ID de périphérique	D86D69A5- CB36-467B-93D0-9726B5E7DFA7
ID de produit	00328-00000-00000-AA734
Type du système	Système d'exploitation 64 bits, processeur x64
Stylet et fonction tactile	La fonctionnalité d'entrée tactile ou avec un stylet n'est pas disponible sur cet écran

Figure 5.1: Spécification de L'Appareil

5.3.2 Specification de Windows

Spécifications de Windows

Édition	Windows 10 Éducation
Version	21H2
Installé le	27/05/2021
Build du système d'exploitation	19044.1645
Expérience	Windows Feature Experience Pack 120.2212.4170.0

Figure 5.2: Spécification De Windows

5.4 L'Application

5.4.1 Coté De Conception:Interface

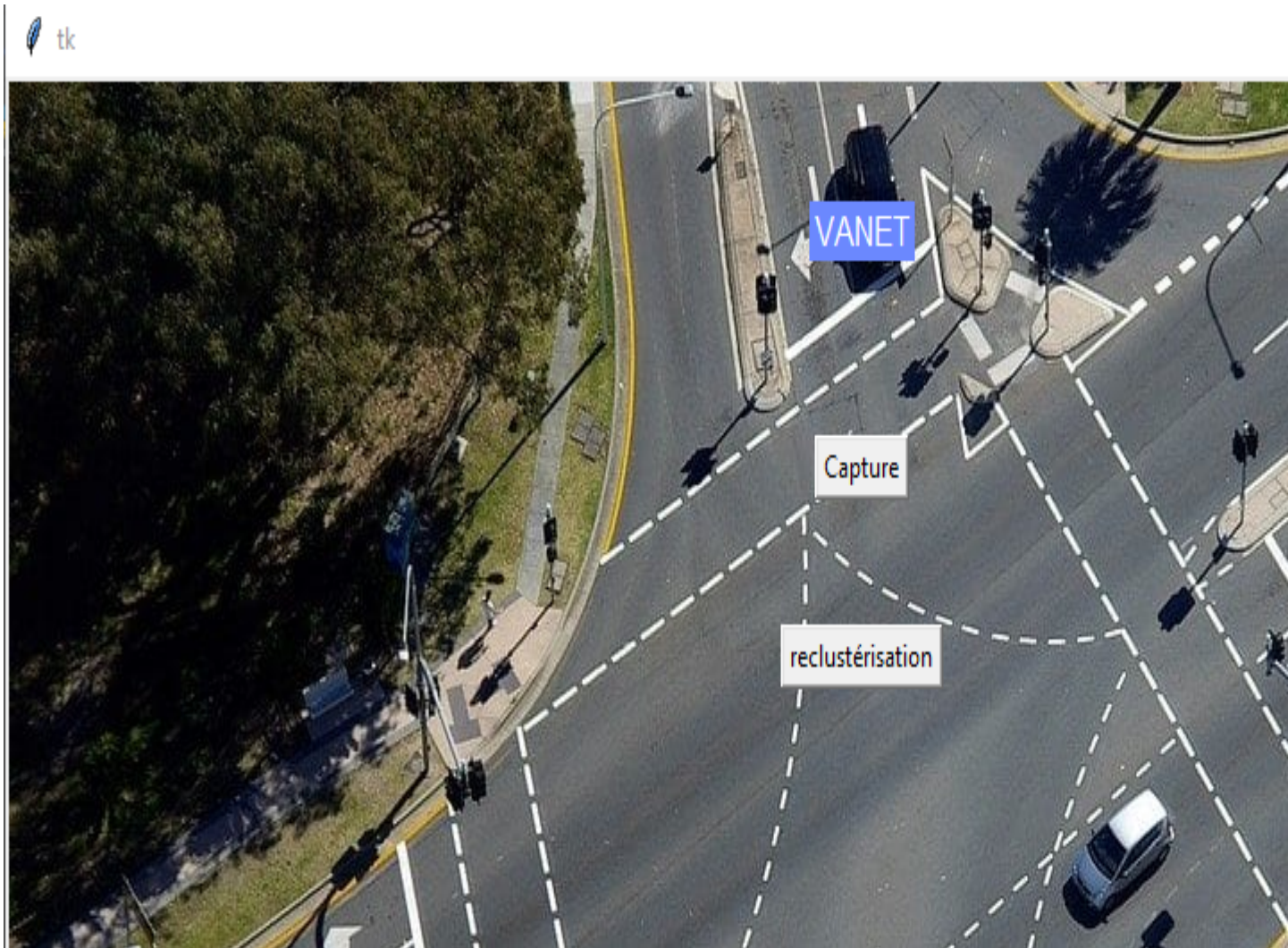


Figure 5.3: page d'accueil de l'application

5.4.1.1 Button Capture

button capture:son fonctionnement est construire un réseau Vanet structuré par base sur la clusterisation lorsque le $t=0$ (t:temp).

5.4.1.2 Button Reclustérisation

gerer et donner un nouveau structure de ce reseaux mais dans un spefique temps ,Nous pouvons Nomme ce processus est un réseau vanet dynamique.

5.4.2 Coté de programmation

5.4.2.1 Algorithm k-means: Les résultats de Capture

5.4.2.2 Reclustering La Structure De Réseau Dans Un T0

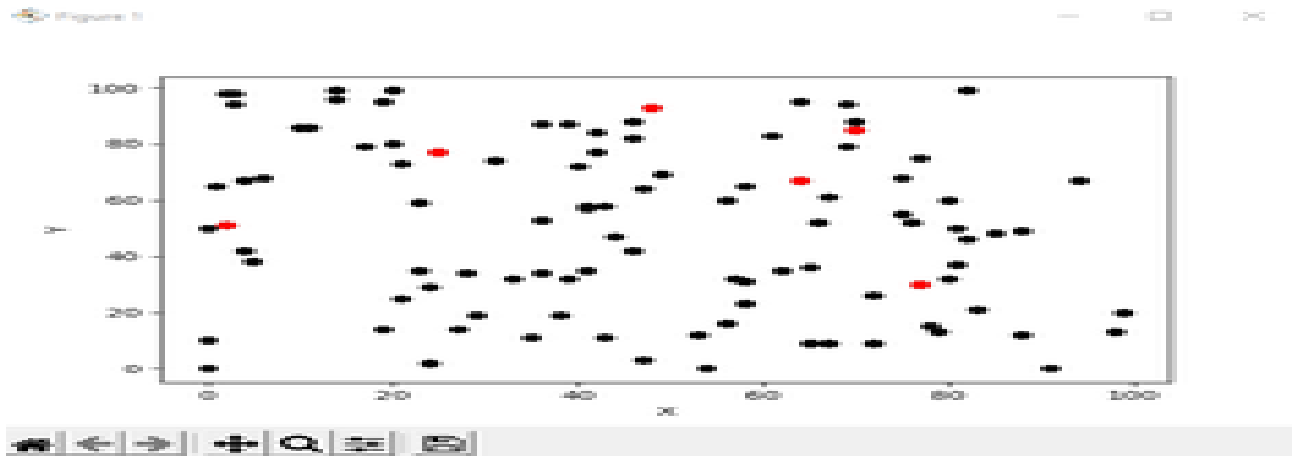


Figure 5.4: visuliation global avant Capture

```
c1 = [8, 20, 28, 36, 43, 95]
head = 28
c2 = [1, 4, 11, 19, 25, 29, 33, 35, 38, 45, 48, 51, 58, 60, 63, 65, 78, 91, 93, 94, 99]
head = 35
c3 = [0, 9, 13, 14, 15, 21, 26, 31, 32, 37, 46, 47, 52, 59, 66, 74, 77, 79, 80, 85, 88, 90, 92]
head = 21
c4 = [3, 16, 18, 23, 30, 34, 41, 54, 62, 64, 67, 70, 76, 82, 83]
head = 62
c5 = [5, 22, 40, 42, 61, 68, 69, 71, 84, 87, 89, 98]
head = 69
c6 = [2, 6, 7, 10, 12, 17, 24, 27, 39, 44, 49, 50, 53, 55, 56, 57, 72, 73, 75, 81, 86, 96, 97]
head = 75
```

Figure 5.5: après Capture

```

c1 = [17, 21, 25, 30, 32, 37, 61, 62, 64, 86, 98]
head1 = 62
c1 = [17, 21, 25, 30, 32, 37, 61, 64, 86, 98]
message1 = [64, 10]
etat1 = [0 0 0 0 0 0 0 0 1 1]
message1 = [64, 8]

```

Figure 5.6: réclustérisation t=1

```

c2 = [0, 4, 5, 8, 9, 12, 24, 31, 35, 45, 47, 48, 55, 60, 67, 81, 84, 85, 92, 97, 99]
head = 8
c2 = [0, 4, 5, 9, 12, 24, 31, 35, 45, 47, 48, 55, 60, 67, 81, 84, 85, 92, 97, 99]
message2 = [9, 20]
etat2 = [1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 0]
message2 = [9, 6]

```

Figure 5.7: réclustérisation t=1 Cluster2

```

c3 = [3, 13, 15, 28, 29, 38, 43, 46, 51, 53, 57, 58, 65, 66, 69, 72, 76, 82, 93, 95]
head3 = 66
c3 = [3, 13, 15, 28, 29, 38, 43, 46, 51, 53, 57, 58, 65, 69, 72, 76, 82, 93, 95]
message3 = [69, 19]
etat3 = [1 0 0 1 1 1 0 0 1 0 0 1 1 0 1 1 1 0 0]
message3 = [69, 9]

```

Figure 5.8: réclustérisation t=1 Cluster3

```
c4 = [1, 14, 19, 27, 33, 34, 36, 41, 50, 70, 87, 91]
head = 1
c4 = [14, 19, 27, 33, 34, 36, 41, 50, 70, 87, 91]
message4 = [14, 11]
etat4 = [1 1 0 0 1 0 1 1 1 1 1]
message4 = [14, 3]
```

Figure 5.9: réclustérisation t=1 Cluster4

```
c5 = [2, 6, 7, 10, 11, 26, 40, 42, 44, 49, 59, 63, 68, 73, 78, 80, 88, 90, 94, 96]
head5 = 42
c5 = [2, 6, 7, 10, 11, 26, 40, 44, 49, 59, 63, 68, 73, 78, 80, 88, 90, 94, 96]
message5 = [44, 19]
etat5 = [0 1 1 1 1 1 0 1 1 0 1 0 1 1 0 1 0 0 0]
message5 = [44, 8]
```

Figure 5.10: réclustérisation t=1 Cluster5

```
c6 = [16, 18, 20, 22, 23, 39, 52, 54, 56, 71, 74, 75, 77, 79, 83, 89]
head6 = 23
c6 = [16, 18, 20, 22, 39, 52, 54, 56, 71, 74, 75, 77, 79, 83, 89]
message6 = [39, 15]
etat6 = [0 1 0 1 0 0 0 0 1 0 1 1 0 0 1]
message6 = [39, 9]
```

Figure 5.11: réclustérisation t=1 Cluster6

5.4.2.3 Reclusterisation la structure de réseau Dans Un T1,T2,T3,T4,T5

```
c1 = [17, 21, 25, 30, 32, 37, 61, 62, 64, 86, 98]
head1 = 62
c1 = [17, 21, 25, 30, 32, 37, 61, 64, 86, 98]
message1 = [64, 10]
etat1 = [0 0 0 0 0 0 0 0 1 1]
message1 = [64, 8]
```

Figure 5.12: Réclusterisation a structure de réseau dans un T1

```
c1 = [15, 18, 19, 46, 48, 49, 72, 73, 80, 92, 96]
head1 = 80
c1 = [15, 18, 19, 46, 48, 49, 72, 73, 92, 96]
message1 = [92, 10]
etat1 = [1 0 1 1 0 1 0 1 0 0]
message1 = [92, 5]
c2 = [12, 20, 24, 28, 35, 36, 37, 44, 47, 55, 57, 67, 77, 81, 88, 91]
head = 28
c2 = [12, 20, 24, 35, 36, 37, 44, 47, 55, 57, 67, 77, 81, 88, 91]
message2 = [35, 15]
etat2 = [0 0 0 1 1 1 0 1 1 0 0 0 1 0 1]
message2 = [35, 8]
c3 = [4, 7, 8, 9, 10, 13, 26, 30, 40, 52, 53, 56, 63, 71, 74, 79, 83, 87, 90]
head3 = 26
c3 = [4, 7, 8, 9, 10, 13, 30, 40, 52, 53, 56, 63, 71, 74, 79, 83, 87, 90]
message3 = [30, 18]
etat3 = [0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 0 1]
message3 = [30, 10]
c4 = [1, 2, 6, 17, 29, 38, 39, 42, 45, 58, 75, 76, 84, 85, 86, 95, 97]
head = 86
c4 = [1, 2, 6, 17, 29, 38, 39, 42, 45, 58, 75, 76, 84, 85, 95, 97]
message4 = [95, 16]
etat4 = [0 1 0 1 0 1 0 1 1 0 0 0 1 1 1 0]
message4 = [95, 8]
```

Figure 5.13: Réclusterisation a structure de réseau dans un T2


```

c1 = [5, 8, 21, 22, 45, 46, 48, 59, 64, 68, 87, 90, 95, 97]
head1 = 8
c1 = [5, 21, 22, 45, 46, 48, 59, 64, 68, 87, 90, 95, 97]
message1 = [21, 13]
etat1 = [1 0 1 1 0 0 0 1 1 0 0 1 1]
message1 = [21, 6]
c2 = [2, 4, 10, 12, 18, 20, 30, 33, 39, 41, 42, 44, 49, 56, 62, 63, 67, 71, 74, 79, 80, 81, 86, 93, 98]
head = 30
c2 = [2, 4, 10, 12, 18, 20, 33, 39, 41, 42, 44, 49, 56, 62, 63, 67, 71, 74, 79, 80, 81, 86, 93, 98]
message2 = [33, 24]
etat2 = [0 1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 1 0 0 0 0 0]
message2 = [33, 13]
c3 = [3, 7, 9, 16, 24, 31, 34, 38, 50, 69, 72, 77, 84, 88, 91, 94, 99]
head3 = 50
c3 = [3, 7, 9, 16, 24, 31, 34, 38, 69, 72, 77, 84, 88, 91, 94, 99]
message3 = [69, 16]
etat3 = [0 0 1 0 0 0 1 1 0 0 1 1 0 1 1 0]
message3 = [69, 9]
c4 = [6, 11, 13, 26, 27, 29, 40, 54, 57, 60, 73, 75, 78, 82, 89, 92, 96]
head = 13
c4 = [6, 11, 26, 27, 29, 40, 54, 57, 60, 73, 75, 78, 82, 89, 92, 96]
message4 = [26, 16]
etat4 = [0 0 0 0 0 0 1 1 1 1 0 1 0 1 0 0]
message4 = [26, 10]
c5 = [0, 14, 15, 17, 23, 36, 37, 47, 52, 58, 61, 70, 83]
head5 = 36

```

Figure 5.14: Réclustering a structure de réseau dans un T3

```

c1 = [7, 12, 17, 22, 32, 33, 35, 36, 37, 44, 49, 54, 55, 72, 78, 86, 99]
head1 = 54
c1 = [7, 12, 17, 22, 32, 33, 35, 36, 37, 44, 49, 55, 72, 78, 86, 99]
message1 = [55, 16]
etat1 = [1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 0]
message1 = [55, 8]
c2 = [0, 4, 8, 11, 13, 27, 31, 38, 39, 42, 43, 47, 58, 63, 70, 74, 76, 91, 93, 98]
head = 93
c2 = [0, 4, 8, 11, 13, 27, 31, 38, 39, 42, 43, 47, 58, 63, 70, 74, 76, 91, 98]
message2 = [98, 19]
etat2 = [0 1 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 1 1]
message2 = [98, 5]
c3 = [14, 21, 23, 28, 41, 52, 57, 59, 60, 61, 65, 75, 82, 83, 84, 85, 90, 92, 96]
head3 = 23
c3 = [14, 21, 28, 41, 52, 57, 59, 60, 61, 65, 75, 82, 83, 84, 85, 90, 92, 96]
message3 = [28, 18]
etat3 = [0 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0]
message3 = [28, 12]
c4 = [1, 5, 15, 19, 25, 30, 45, 51, 53, 56, 66, 68, 69, 79, 81, 87, 88]
head = 56
c4 = [1, 5, 15, 19, 25, 30, 45, 51, 53, 66, 68, 69, 79, 81, 87, 88]
message4 = [66, 16]
etat4 = [0 0 0 1 1 1 1 0 0 1 0 1 1 0 1 0]
message4 = [66, 8]

```

Figure 5.15: Réclustering a structure de réseau dans un T4

```

c1 = [3, 17, 23, 28, 42, 47, 54, 56, 59, 60, 61, 63, 72, 74, 85, 95, 96]
head1 = 59
c1 = [3, 17, 23, 28, 42, 47, 54, 56, 60, 61, 63, 72, 74, 85, 95, 96]
message1 = [60, 16]
etat1 = [0 0 0 1 1 0 1 0 0 0 0 1 0 0 1 0]
message1 = [60, 11]
c2 = [2, 6, 22, 27, 30, 34, 39, 43, 49, 52, 67, 76, 82, 84]
head = 67
c2 = [2, 6, 22, 27, 30, 34, 39, 43, 49, 52, 76, 82, 84]
message2 = [76, 13]
etat2 = [1 0 1 0 0 1 1 1 1 1 1 1 1 0]
message2 = [76, 4]
c3 = [4, 7, 9, 26, 29, 31, 35, 45, 48, 50, 69, 70, 81, 91, 99]
head3 = 7
c3 = [4, 9, 26, 29, 31, 35, 45, 48, 50, 69, 70, 81, 91, 99]
message3 = [9, 14]
etat3 = [1 1 1 0 0 1 0 1 0 0 0 0 1 0]
message3 = [9, 8]
c4 = [8, 10, 15, 18, 19, 21, 32, 33, 37, 38, 40, 51, 57, 58, 87, 92]
head = 40
c4 = [8, 10, 15, 18, 19, 21, 32, 33, 37, 38, 51, 57, 58, 87, 92]
message4 = [51, 15]
etat4 = [1 0 1 0 0 0 1 1 1 0 0 0 0 1 0]
message4 = [51, 9]

```

Figure 5.16: Réclustering a structure de réseau dans un T5

5.4.2.4 Algorithm K-medoid: Les résultats de Capture

5.4.2.5 Capturer la structure de réseau dans un T0

```

head1 = 59
head2 = 81
head3 = 22
head4 = 44
head5 = 91
head6 = 74
c1 = [1, 16, 23, 25, 34, 36, 38, 42, 43, 55, 67, 76, 80, 90]
c2 = [5, 8, 33, 49, 52]
c3 = [4, 6, 10, 11, 12, 13, 15, 17, 18, 21, 24, 27, 29, 30, 35, 39, 40, 41, 45, 46, 50, 51, 53, 54, 58, 61, 63, 65, 71, 72, 73, 79, 85, 88, 94, 98, 99]
c4 = [0, 7, 28, 37, 60, 62, 64, 66, 68, 69, 70, 75, 77, 78, 82, 83, 87, 89, 92, 93, 95]
c5 = [2, 3, 19, 26, 32, 47, 56, 57, 84]
c6 = [9, 14, 20, 31, 48, 86, 96, 97]

```

Figure 5.17: Capturer la structure de réseau dans un T0

```
c1 = [10, 40, 42, 44, 49, 68, 73, 78, 88]
head1 = 80
message1 = [80, 9]
etat1 = [1 0 1 1 0 1 0 0 0]
message1 = [80, 5]
```

Figure 5.18: Cluster 1

```
c2 = [2, 6, 7, 11, 13, 17, 21, 25, 26, 30, 32, 38, 53, 57, 58, 59, 62, 63, 76, 82, 86, 93, 94, 96]
head2 = 82
message2 = [82, 24]
etat2 = [1 0 1 1 0 1 1 1 0 0 1 1 1 0 0 0 1 0 0 1 1 0 0 1]
message2 = [82, 11]
```

Figure 5.19: Cluster2

```
c3 = [3, 5, 15, 28, 29, 31, 37, 43, 46, 51, 61, 64, 65, 66, 67, 69, 72, 81, 95, 98]
head3 = 16
message3 = [16, 20]
etat3 = [1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 1 0 1 0 0]
message3 = [16, 11]
```

Figure 5.20: Cluster3

```
c4 = [14, 19, 27, 33, 34, 36, 41, 50, 70, 91]
head4 = 83
message4 = [83, 10]
etat4 = [0 0 1 1 1 0 1 1 0 1]
message4 = [83, 4]
```

Figure 5.21: Cluster4

```
c5 = [1, 16, 18, 20, 22, 23, 39, 52, 54, 56, 71, 74, 75, 79, 83, 89]
head5 = 48
message5 = [48, 16]
etat5 = [1 1 1 0 0 1 1 0 1 0 1 1 1 0 1 1]
message5 = [48, 5]
```

Figure 5.22: Cluster5

```
c6 = [0, 4, 9, 12, 24, 35, 45, 47, 55, 60, 84, 85, 92, 97, 99]
```

Figure 5.23: Cluster6

```
head6 = 1
message6 = [1, 15]
etat6 = [0 1 1 1 1 1 1 0 1 1 0 1 0 1 0]
message6 = [1, 5]
```

Figure 5.24: Cluster6

5.4.2.6 Algorithm k-medoids: Les résultats de Réclusterisation

```
c1 = [10, 40, 42, 44, 49, 68, 73, 78, 88]
head1 = 80
message1 = [80, 9]
etat1 = [1 0 1 1 0 1 0 0 0]
message1 = [80, 5]
```

Figure 5.25: Réclusterisation la structure de Réseau dans un T1

```
c1 = [0, 3, 5, 11, 14, 17, 18, 21, 23, 25, 27, 39, 43, 46, 51, 54, 59, 60, 61, 64, 65, 66, 68, 80, 89, 93, 94, 95]
head1 = 82
message1 = [82, 28]
etat1 = [1 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0]
message1 = [82, 20]
c2 = [10, 40, 56, 74, 79]
head2 = 84
message2 = [84, 5]
etat2 = [0 0 0 1 0]
message2 = [84, 4]
c3 = [12, 15, 19, 20, 24, 28, 35, 36, 37, 44, 47, 48, 55, 57, 67, 72, 73, 81, 88, 91, 92, 96]
head3 = 83
message3 = [83, 22]
etat3 = [0 0 1 0 1 0 1 1 0 0 0 1 0 0 1 0 0 0 0 1 1 0]
message3 = [83, 14]
c4 = [1, 2, 6, 22, 29, 42, 45, 58, 75, 76, 84, 85, 86, 97]
head4 = 57
message4 = [57, 14]
etat4 = [1 1 1 0 1 0 0 0 0 0 0 1 1 0]
message4 = [57, 8]
c5 = [4, 8, 13, 16, 26, 31, 32, 33, 34, 41, 49, 50, 62, 69, 70, 78, 90, 98, 99]
head5 = 45
message5 = [45, 19]
etat5 = [1 1 0 0 0 0 1 0 1 1 0 0 1 0 1 1 1 1 1]
```

Figure 5.26: Réclusterisation la structure de Réseau dans un T2

```

c1 = [2, 10, 12, 18, 19, 25, 28, 30, 32, 33, 41, 49, 63, 65, 66, 67, 71, 76, 80, 81, 85, 98]
head1 = 20
message1 = [20, 22]
etat1 = [1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 1 1 0 0]
message1 = [20, 12]
c2 = [1, 35, 43, 51, 53]
head2 = 73
message2 = [73, 5]
etat2 = [1 0 1 1 0]
message2 = [73, 2]
c3 = [5, 8, 21, 45, 46, 54, 59, 60, 64, 82, 90, 92, 95, 97]
head3 = 46
message3 = [46, 14]
etat3 = [0 1 1 0 0 0 0 1 0 1 0 1 0 0]
message3 = [46, 9]
c4 = [3, 4, 7, 9, 24, 31, 34, 38, 39, 42, 62, 69, 84, 88]
head4 = 87
message4 = [87, 14]
etat4 = [1 1 0 1 0 0 0 0 0 1 0 0 1 1]
message4 = [87, 8]
c5 = [6, 11, 13, 16, 17, 22, 26, 29, 40, 44, 48, 50, 56, 57, 68, 72, 73, 74, 75, 78, 79, 86, 87, 89, 91, 93, 94, 96, 99]
head5 = 23
message5 = [23, 29]
etat5 = [0 1 1 1 1 0 0 0 0 1 1 1 1 0 1 1 1 0 0 1 0 0 0 0 0 1 0 1]

```

Figure 5.27: Réclusterisation la structure de Réseau dans un T3

```

c1 = [25, 32, 55, 66, 72, 86, 88]
head1 = 22
message1 = [22, 7]
etat1 = [1 1 0 0 0 0 0]
message1 = [22, 5]
c2 = [3, 7, 9, 13, 18, 26, 29, 34, 46, 48, 50, 59, 63, 64, 67, 77, 80, 83, 94, 99]
head2 = 37
message2 = [37, 20]
etat2 = [1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1]
message2 = [37, 4]
c3 = [6, 12, 15, 56, 81]
head3 = 42
message3 = [42, 5]
etat3 = [0 1 0 1 0]
message3 = [42, 3]
c4 = [0, 2, 4, 8, 10, 11, 14, 16, 24, 27, 31, 38, 39, 40, 42, 43, 47, 58, 62, 70, 71, 73, 74, 76, 89, 91, 93, 97, 98]
head4 = 94
message4 = [94, 29]
etat4 = [0 0 0 0 0 0 1 1 1 1 0 1 0 0 0 0 0 1 0 0 1 1 1 1 0 1 0 0 0]
message4 = [94, 18]
c5 = [17, 21, 23, 28, 33, 35, 36, 41, 44, 49, 52, 54, 57, 60, 61, 65, 75, 78, 82, 84, 85, 90, 92, 96]
head5 = 96
message5 = [96, 24]
etat5 = [0 1 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 1 0 0 1 1]

```

Figure 5.28: Réclusterisation la structure de Réseau dans un T4

```

c1 = [51]
head1 = 37
message1 = [37, 1]
etat1 = [1]
message1 = [37, 0]
c2 = [11, 13, 20, 25, 36, 41, 46, 53, 55, 62, 64, 66, 71, 73, 78, 79, 80, 83, 86, 88, 94, 97]
head2 = 70
message2 = [70, 22]
etat2 = [0 1 0 0 1 0 0 0 1 0 1 1 0 1 1 1 1 1 0 1 1 1]
message2 = [70, 9]
c3 = [2, 3, 5, 6, 9, 22, 23, 26, 27, 28, 30, 34, 39, 42, 43, 47, 49, 52, 54, 56, 59, 61, 63, 67, 68, 70, 74, 76, 82, 85, 96, 98]
head3 = 29
message3 = [29, 32]
etat3 = [0 1 0 1 1 0 1 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 1 1 1 1 0 1 0 1 0 0]
message3 = [29, 17]
c4 = [4, 7, 17, 29, 31, 35, 45, 48, 60, 69, 72, 81, 91, 95, 99]
head4 = 50
message4 = [50, 15]
etat4 = [1 1 1 1 0 0 0 0 1 0 0 1 0 1 0]
message4 = [50, 8]
c5 = [10, 15, 18, 19, 21, 32, 33, 38, 40, 57, 58, 65, 87, 92]
head5 = 39
message5 = [39, 14]
etat5 = [1 1 1 1 0 1 0 1 1 1 1 1 0 0 1]

```

Figure 5.29: Réclusterisation la structure de Réseau dans un T5

5.4.2.7 Evaluation les résultats:évaluer un seul echantillon de résultats

voiture	etat
15	N
18	N
19	N
46	N
48	N
49	N
72	N
73	N
73	N
80	H
96	N
92	N

Table 5.1: Cluster 1 EN T2

voiture	etat
5	N
8	H
97	N
21	N
22	N
65	N
48	N
59	N
64	N
68	N
87	H
98	N
95	N

Table 5.2: Cluster 1 EN T3

5.4.2.8 Mobilite(Cluster1)

5.4.2.9 temp:t2

Nb Voiture	Etat Head
12	oui

Table 5.3: Mobilite Cluster 1 EN T2

Nb Voiture	Etat Head
13	oui

Table 5.4: Mobilite Cluster 1 EN T3

5.4.2.9.1 Analyse

- nous avons voir dans ces deux table precedent le nombre des voiture dans clusterv1 elle change ou sinigfie elle eleve entre deux temps different(t2,t3 grace a la mobilité dans ce resau car les voitures elles movent dans ce reseaux
- l’algorithme k means elle bien fonctionné de performance au niveau de realisation la structure de reseau vanet base a clusterisation.

5.4.2.10 Routage

5.4.2.11 Analyse

- head de cluster 1 en temp=2 contient 10 message pour livrer a tout les noued de notre cluster ,mais je observe dans table 3,les message 10 nest pas livrer a tout les noued
- grace la mobilité dans ce reseau elle perte des message ,dans ce cas elle perte 5 messages sur 10 messages

Head	Etat Nb Message
80	10

Table 5.5: Avant Routage Les Message Dans Cluster 1 EN T2

Voiture	Etat De Message
15	1
18	0
19	1
46	1
48	0
49	1
72	0
73	1
96	0
92	0

Table 5.6: Après Routage Les Message Dans Cluster 1 EN T2

- 0:n pas reception de message
- 1:reception de meassage

Head	Etat Nb Message
80	5

Table 5.7: Après Routage Les Message Dans Cluster 1 EN T2

5.4.2.12 Evaluation Les Resultats De kmedoids:Évaluer Un Seul Echantillon De Résultats

Voiture	Etat
9	N
30	N
52	N
63	N
71	N
87	N
89	H

Table 5.8: Cluster 6 EN T2

Voiture	Etat
1	N
5	H
19	N
30	N
45	N
53	N
69	N
79	N
87	N
64	H

Table 5.9: Etat Routage De Cluster 6 EN T4

5.4.2.13 Mobilite(cluster6)

5.4.2.13.1 temp:t2

nb voiture	etat head
6	oui

Table 5.10: Mobilite Cluster 6 EN T4

Nb Voiture	Etat Head
9	oui

Table 5.11: Mobilite Cluster 6 EN T2

5.4.2.13.2 Analyse

- nous avons vu dans ces deux tables précédentes le nombre de voitures dans le cluster 6, elle change ou signifie-elle l'élévation entre deux temps différents (t_2, t_4) grâce à la mobilité dans ce réseau car les voitures elles-mêmes bougent dans ce réseau
- l'algorithme k-medoids, bien qu'il fonctionne, a de mauvaises performances au niveau de la réalisation de la structure de réseau basée sur la clusterisation

5.4.2.14 Routage

5.4.2.14.1 Analyse

- le head du cluster 6 au temps $t=2$ contient 6 messages pour être livrés à tous les nœuds de notre cluster, mais je remarque dans le tableau 3, les messages 6 ne sont pas livrés à tous les nœuds, ainsi dans ce cluster au temps $t=4$, les messages 9 ne sont pas livrés à tous les nœuds, dans ce temps elle perd 3 messages sur 9 messages
- grâce à la mobilité dans ce réseau, elle perd des messages

head	etat nb message
89	6

Table 5.12: avant routage les messages dans le Cluster 6 EN T2

Voiture	Etat De Message
9	0
30	1
52	0
63	0
71	0
87	1

Table 5.13: Après Routage Les Messages Dans le Cluster 6 EN T2

- 0: n pas de réception de message
- 1: réception de message

voiture	etat de message
1	1
5	1
19	0
30	1
45	0
53	1
69	0
79	1
87	1

Table 5.14: etat de messge Cluster 6 EN T4

head	etat nb message
64	9

Table 5.15: avant routage :les message dans Cluster 6 EN T4

5.4.3 Interprétations

Nous notons qu'il existe une grande différence entre la façon dont le réseau automobile est formé à l'aide de cummins et de ketweed, et cela est dû à la grande différence dans leur fonctionnement. Nous notons que le routage au sein d'un réseau ne donne pas un objectif satisfaisant, qui est causé par la mobilité, c'est-à-dire le mouvement des voitures, pour constater que les messages sont perdus et que les messages n'atteignent pas leurs destinations, et la raison en est que la voiture s'est déplacée et a quitté le groupe, et donc le message ne l'a pas atteint. C'est selon l'un des principes de ce type de pilotage que nous avons retenu (guidage par clusters).

5.4.4 Statistiques

Nous avons supposé que la raison de la satisfaction élevée (recevoir et ne pas perdre un grand nombre de messages) dans certains clusters, est que la stabilité était élevée par rapport à la mobilité de ces clusters.

Algorithmes	Nb cluster	Taux de messages temps t0	Taux de messages temps t1	Taux de messages temps t2	Taux de messages temps t3	Taux de messages temps t4	Taux de messages temps t5
<u>k-means</u>	6	aucun	C1=20% C2=70% C3=52.63% C4=72.72% C5=57.89% C6=40%	C1=50% C2=44.44% C3=19.75% C4=50% C5=46.66% C6=50%	C1=53,84% C2=45,83% C3=43,75% C4=37,5% C5=33,33% C6=38,46%	C1=50% C2=73,68% C3=33,33% C4=50% C5=50% C6=26,31%	C1=31,25% C2=61,53% C3=42,85% C4=46,66% C5=37,5% C6=20%
<u>k-medoid</u>	6	aucun	C1=44,44% C2=54,16% C3=45%	C1=28,57% C2=20% C3=36,36%	C1=45,45% C2=60% C3=35,71%	C1=28,57% C2=70%	C1=0 C2=59,09%
			C4=60% C5=68,75% C6=66,66%	C4=42,85% C5=57,89% C6=46,66%	C4=42,85% C5=51,72% C6=31,25%	C3=20% C4=37,93% C5=50% C6=29,31%	C3=46,87% C4=46,66% C5=71,42% C6=31,55%

Figure 5.30: Algorithme k-medoid :les statistiques de routage

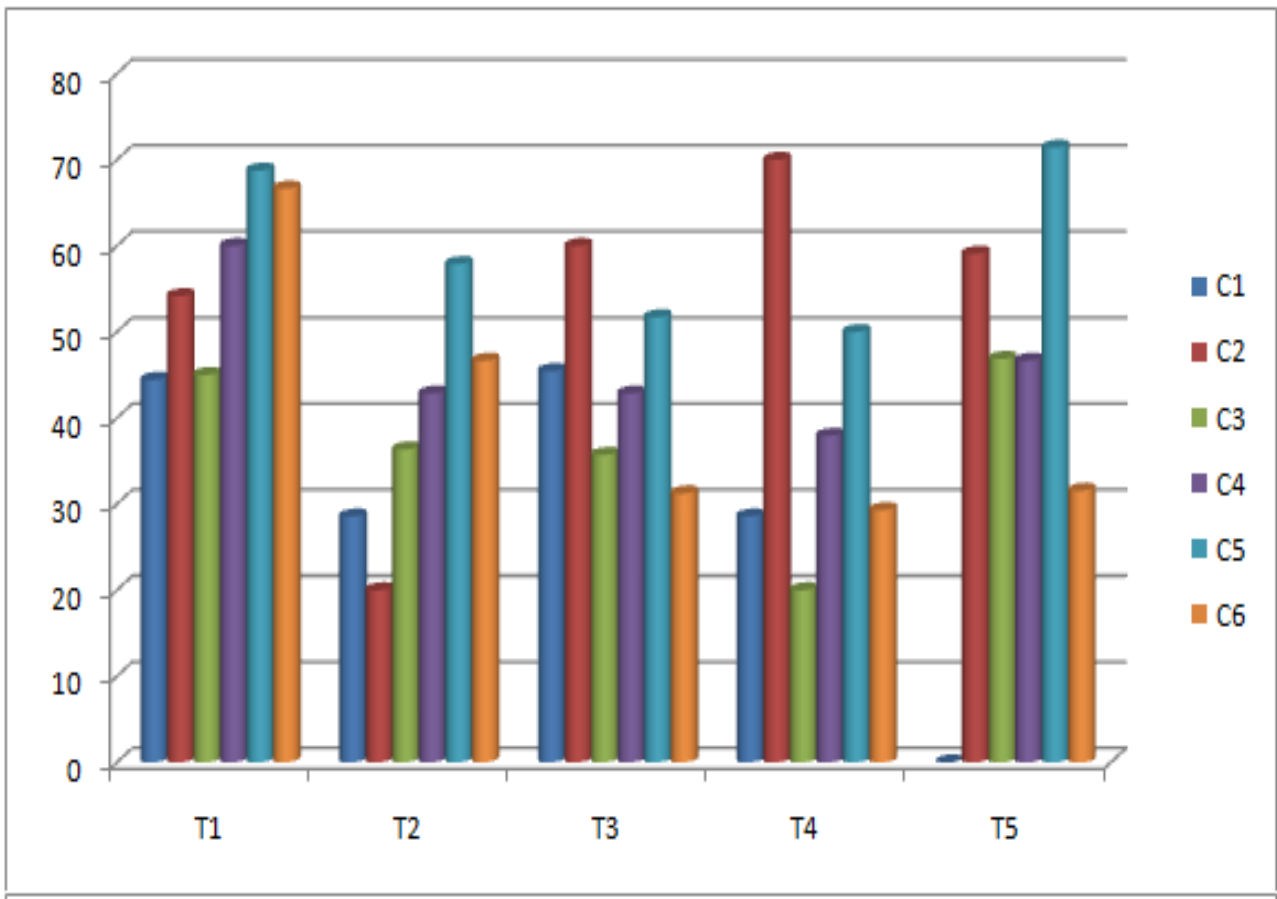


Figure 5.31: Algorithmes K-means et k-medoid :les statistiques de routage

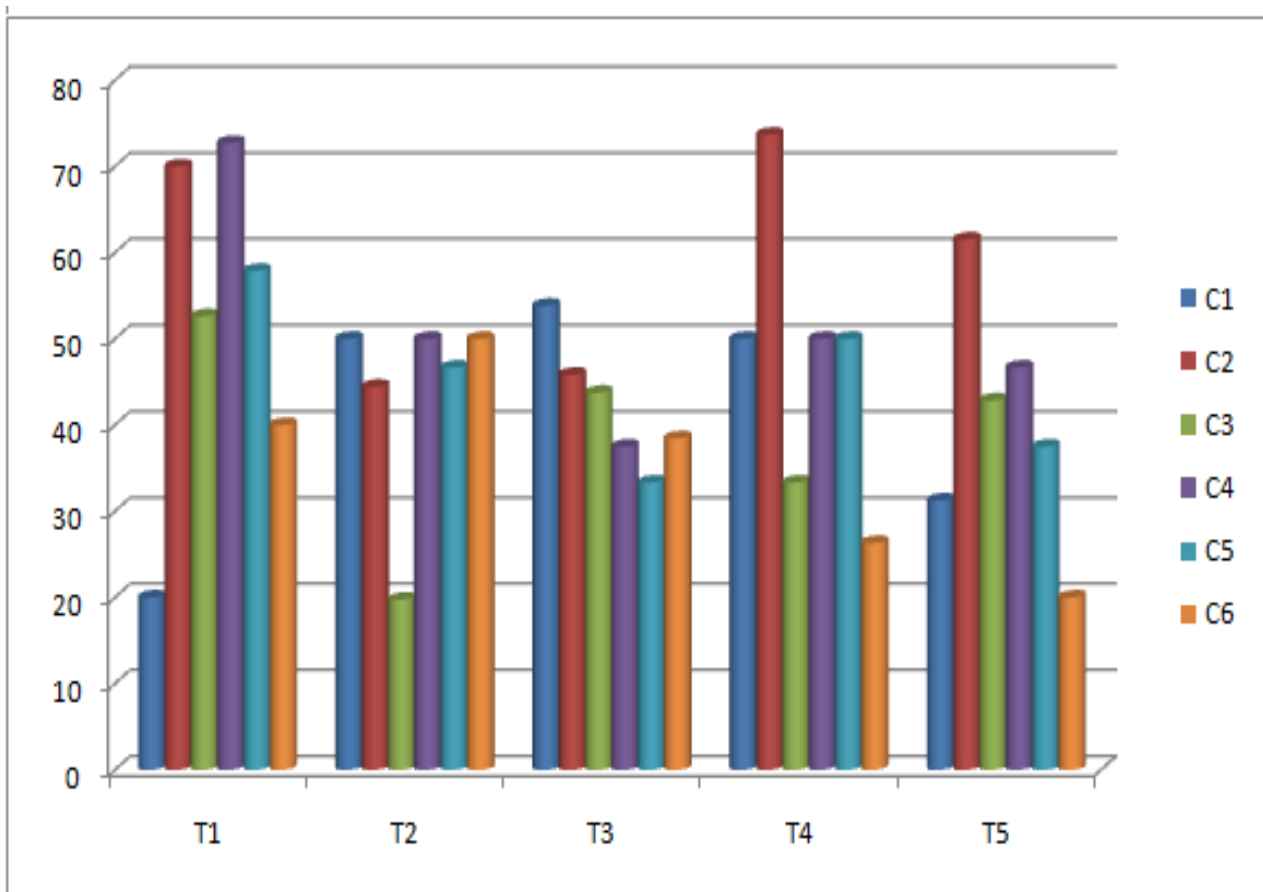


Figure 5.32: Algorithme K-means :les statistiques de routage

5.5 Conclusion

Chacun des algorithmes utilisés pour réaliser un réseau a bien fonctionné au niveau de la mobilité, mais ce dernier nous a causé des problèmes de perte de messages dans le routage au sein de chaque cluster.

Conclusion générale

Le projet de notre mémoire traite la QoS dans les réseaux véhiculaires VANET et spécifiquement dans le routage. Dans ce mémoire, nous avons utilisé des techniques de clustering du réseau VANET pour permettre un routage d'informations fiable et rapide dans ce genre de réseaux. Cette technique de clustérisation est mise en œuvre par des métaheuristiques comme l'algorithme de clustérisation k-medoids et l'algorithme k-means pour concevoir un protocole de routage pour les VANETs. Ces techniques maximisent la stabilité du réseau en diminuant les ruptures des liens dues à la forte mobilité des nœuds. Nous avons proposé un modèle structuré avec des clusters que nous prenons à un niveau dynamique comme le mouvement des voitures et le routage des messages entre eux. Notre but était de concevoir un modèle permettant de réduire les conséquences des accidents de la route et offrir plus de confort aux conducteurs et aux voyageurs. En effet, les résultats auxquels nous sommes parvenus ouvrent la voie à plusieurs perspectives qui peuvent faire objet de travaux futurs. Comme l'utilisation d'autres métaheuristiques pour optimiser la QoS du routage dans les VANETS comme les algorithmes génétiques et les algorithmes par essais particuliers (PSO : SwarmOptimization).

Bibliography

- [1] Résolution à base d'heuristiques du problème de routage dans les réseaux ad hoc de véhicules, author=Hajlaoui, Rejab, year=2018, school=Bourgogne Franche-Comté https://www.ummo.dz/dspace/bitstream/handle/ummo/6587/DadounOuiza_GuendouzOuiza.pdf?sequence=1&isAllowed=y,
- [2] Le routage dans les réseaux véhiculaires VANETs, https://www.ummo.dz/dspace/bitstream/handle/ummo/6587/DadounOuiza_GuendouzOuiza.pdf?sequence=1&isAllowed=y,
- [3] l'implimentation l'algorithmes k-medoid , <https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/>,
- [4] Regroupement avec K-medoids et Common-nearest-neighbors, <https://scikit-learn-extra.readthedocs.io/en/stable/modules/cluster.html>,
- [5] les étapes de l'algorithme k-medoids , <https://www.youtube.com/watch?v=uTiGtmGuQrk>,
- [6] Regroupement avec KMedoids et Common-nearest-neighbors, <https://en.wikipedia.org/wiki/K-medoids>,
- [7] l'implimentaion de algorithm k-means, <https://datascience.eu/fr/apprentissage-automatique/k-means-le-clustering-dans-lapprentissage-machi>
- [8] les étapes de l'algorithme k-means , <https://commons.wikimedia.org/wiki/File:K-means.png>,
- [9] Routage à vecteur de distance séquencé par destination, https://en.wikipedia.org/wiki/Destination-Sequenced_Distance_Vector_routing,
- [10] GPSR protocol.,<https://www.icir.org/bkarp/gpsr/gpsr.html?fbclid=IwAR2vKRXBdxM6PWf2GWrKujDTX1eTaEAqZXPZGnUj7AgSs6xWOY-FGiNcXxE>,
- [11] GPSR protocol.,<https://journals.sagepub.com/doi/full/10.1177/0020294020932359>,
- [12] PyCharm., <https://en.wikipedia.org/wiki/PyCharm>,
- [13] Python., [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)),
- [14] Tkinter GUI, <https://fr.wikipedia.org/wiki/Tkinter>,
- [15] Math Bibliotheque, <https://docs.python.org/3/library/math.html>,

- [16] Numpy Bibliotheque, <https://numpy.org/doc/stable/user/whatisnumpy.html>,
- [17] Random Bibliotheque., <https://www.geeksforgeeks.org/python-random-module/>,
- [18] Pandas Bibliotheque., https://pandas.pydata.org/docs/getting_started/overview.html,