DEVELOPMENT OF AN ENHANCED WEIGHT- BASED CLUSTER HEAD SELECTION ALGORITHM FOR VEHICULAR ADHOC NETWORKS.

BY

JIYA, Amos Gana
MENG/SEET/2018/8039

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL, FEDERAL UNIVERSITY OF TECHNOLOGY, NIGERIA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF ENGINEERING IN COMMUNICATION ENGINEERING

JUNE, 2023

DECLARATION

I hereby declare that this thesis titled: “Development of an Enhanced Weight-Based Cluster Head Selection Algorithm for Vehicular Adhoc Networks” is a collection of my original research work and it has not been presented for any other qualification anywhere. Information from other sources (published or unpublished) has been duly acknowledged.

JIYA, Amos Gana

MEng/SEET/2018/8039

FEDERAL UNIVERSITY OF TECHNOLOGY

MINNA, NIGERIA

SIGNATURE/DATE

CERTIFICATION

The thesis titled: “Development of an Enhanced Weight-based Cluster Head Selection Algorithm for Vehicular Adhoc Networks” by: JIYA, Amos Gana (MEng/SEET/2018/8039) meets the regulations governing the award of the degree of MEng of the Federal University of Technology, Minna and it is approved for its contribution to scientific knowledge and literary presentation.

Engr. Dr. A. U. Usman ____________________________

MAJOR SUPERVISOR Signature & Date

Engr. Dr. M. David ____________________________

CO-SUPERVISOR Signature & Date

Engr. Dr. A. U. Usman ____________________________

HEAD OF DEPARTMENT Signature & Date

Engr. Prof. E. N. Onwuka ____________________________

DEAN OF SCHOOL OF ELECTRICAL Signature & Date

ENGINEERING AND TECHNOLOGY

Engr. Prof. O. K. Abubakre ____________________________

DEAN OF POSTGRADUATE SCHOOL Signature & Date

ACKNOWLEDGEMENT

My most sincere gratitude goes to the Almighty God, the Most sovereign and Most supreme, for His enormous love and sustenance throughout the duration of this programme.

My heartfelt gratitude goes to my major Supervisor, Engr. Dr. A. U. Usman and my co-supervisor Engr. Dr. M. David for their patience and guidance, I must confess, this little time with you has exposed me to the rudiments and technicalities of technical writing. Thank you very much. I must also appreciate Engr. Dr. J. A. Abolarinwa with whom I started this work as well as the entire staff of Telecommunication Engineering Department.

My sincere appreciation goes to my entire family, the family of Mr. and Mrs. Zaccheaus Jiya for their support throughout this programme. Special appreciation goes to my wife, Mrs. Michal Jiya, you have been God’s best gift to me and for that I remain immensely indebted to you. To all those who supported my quest for a higher degree both morally and financially, I say thank you. This page may not be enough to mention all your names but I will try to sincerely appreciate Mr. and Mrs. Abel Jibril Gana, Engr. Frank B. Alonge, Mr. Josiah Gana, Abdulhakeem Adebukola, and Abiiba Mary Damilola. These people propelled me with their support. I am deeply grateful.

To all my friends, Yakubu Hassan who kept asking how far? Nwagu Henry Ogbu, Toluwani Alabi Bukola and Abidemi Adejumo (Stochastic Prof). Special thanks to Dawar Philemon Katfun, and Mr. Emmanuel for finding time to go through my work. I am glad our paths crossed the time they did. God bless you.

ABSTRACT

Vehicular Adhoc Networks (VANETs) are a type of mobile adhoc networks that are typically characterized by frequent topology changes, they are either vehicle to vehicle, vehicle to infrastructure or vehicle to vehicle to infrastructure networks. Clustering in VANET is a way of breaking the vehicular network into smaller units. The problem posed by existing cluster head selection algorithm is such that cluster selection process has to be carried out whenever a cluster head leaves the cluster this increases network overhead and useful time is wasted on re-selection every time. In this work, an enhanced cluster head selection algorithm whose objective is to minimize cluster head selection frequency is developed by indexing all participating nodes in decreasing order of suitability value. With this indexing, the next most suitable vehicle will automatically assume the cluster head based on predefined criteria. The algorithm was benchmarked with the existing cluster head selection algorithm in terms of end to end delay, cluster head selection time and packet delivery ratio for varying number of nodes. The cluster head selection time improved against the existing algorithm by 51% for 40 and 70 nodes and 49% for 50 and 60 nodes. For the end-to-end delay, there was 32%, 26%, 35% and 37% improvement for 40, 50, 60 and 70 nodes respectively. In terms of packet delivery ratio, there was a 0% improvement for 40 nodes, 11.4%, 26% and 28.5% for 50, 60 and 70 nodes respectively. This shows that when cluster heads are selected with minimal overhead, stable clusters can be formed, and energy is conserved as frequency of cluster head selection is reduced because of the use of suitability index.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	MEANING
AU	Auxiliary Unit
CCH	Common Control Channel
CH	Cluster Head
CHF	Cluster Head Formation
CM	Cluster Member
DSRC	Direct Short Range Communication
E2E	End to End
HetNet	Heterogenous Networks
IEEE	Institute of Electrical Electronics Engineers
ITS	Intelligent Transport System
MANETS	Mobile AdhocNetworks
MIMO	Multiple Input Multiple Output
OBU	Onboard Unit
PDR	Packet Delivery Ratio
RSU	Roadside Unit

SCH	Service Channel
VANETS	Vehicular AdhocNetworks
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
V2V2I	Vehicle to Vehicle to Infrastructure
V2N	Vehicle to Network
V2D	Vehicle to Device
V2X	Vehicle to Everything
WAVE	Wireless Access in Vehicular Environment

CHAPTER ONE

1.0 INTRODUCTION

This section provides a background knowledge on vehicular communication. It explains the basis for vehicular communication and clustering as well as cluster head selection. It also states the scope, aim and objectives of this work, justification as well as limitation. This section also states the contribution of this work to the research space and the body of knowledge.

1.1 Background to the Study

One of the fastest-growing research areas in the field of communication engineering is adhoc networks. Adhoc networks are temporary networks, they do not require a central entity coordinating them; instead, the communicating nodes are able to use tailored techniques to control communication among themselves. Vehicular adhoc networks (VANETs) are a category of mobile adhoc networks (MANETs), and are typically nodes on wheels, with mobility (Manoj and Charanjeet, 2019). Vehicular communications have emerged as an important application of wireless technology. Vehicular communication networks are an interconnection of vehicles to achieve autonomous driving. Vehicular Adhoc Networks (VANETs) could be Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N), Vehicle-to-Devices (V2D) and generally, Vehicle-to-Everything (V2X) (Khayat *et al.*, 2020).

In recent times, a trend in hybrid connectivity is fast emerging as seen in vehicle-to-vehicle-to-infrastructure (V2V2I) (Fuqiang and Lianhai, 2010). This implies that vehicles are not only communicating among themselves, or communicating with cellular towers but they communicate among themselves and with network infrastructure also referred to as road side unit (RSU) (Aljeri and Boukerche, 2017). Communication with network infrastructure allows vehicular clients access remote services (Emara *et al.* 2018; Huang *et al.*, 2020). Considering the limited capabilities of vehicles in terms of storage and processing, it is imperative that VANETS should be equipped with higher storage and processing capabilities. Alternatively, provisions are being made for storage and computation at the edge of the infrastructure (Wang *et al.*, 2018; Wang *et al.*, 2020; Zhou *et al.*, 2018). Inter-vehicular communication is made possible via direct short-range communication (DSRC), while V2I communication is made possible by wireless access in vehicular environment (WAVE) which consists of 802.11p protocol (Nkoko and Kogeda, 2013; Zhou *et al.*, 2018) among several other protocols.

Communication is possible in V2V networks using sensors. The Onboard Unit (OBU) is a network of sensors which are always in constant communication with other sensor nodes (Raza *et al.*, 2019; Storck and Duarte-Figueiredo, 2019). The VANETs perform such functions as vehicle diagnostics services, location information reporting, and communication with other vehicles and infrastructure, provision of safety information and monitoring for road users, information and entertainment (infotainment), traffic management and internet connectivity. Busari *et al.*(2019) proposed a generalized hybrid beam forming technique for connectivity in vehicular communication using massive Multiple Input Multiple Output (MIMO). A new parameter known as sub-array spacing was introduced. Varying this parameter, brings about different sub-array configuration and by extension, variations in system performance.

Clustering algorithms in VANETs group vehicles in a given spatial location based on certain properties such as speed, direction of travel and lane identification (id). Clusters are managed by cluster heads, other vehicles within the vicinity of the cluster head assumes the status of cluster members. Packet transmission and reception from or to any member is done through cluster heads. The cluster head therefore serves as routers as in traditional computer networks. The cluster head forwards packets to Road Side Units (RSUs) within the vicinity of the cluster. A vehicle that must be chosen as cluster head must meet certain criteria. Several protocols are available in literature and major emphasis of these protocols is optimal cluster head selection which will reduce packet delay, maximize throughput and also reduce packet loss. A number of challenges have been witnessed in the area of vehicular communications. One of such is cluster head selection in cluster-based communication (Duan, 2016).

In order to bolster research and development of Intelligent Transport Systems (ITS), clustering has emerged as a means or disseminating information, in clustering in vehicular communications, vehicles are either cluster heads (CH) or cluster members. cluster heads are chosen on the basis of balanced parameters and enhanced functionality (Khayat *et al*., 2020), such parameters include speed, direction of travel, driver behavior, inter-nodal distance and communication range. It is desired that a cluster have good stability, high efficiency and reduced frequency of cluster head selection. Without loss of generality, cluster heads must have good ranking in order to be chosen as cluster heads. Cluster heads are required to coordinate inter cluster and intra cluster communication.

The IEEE 802.11p standard is a proposed standard which is meant to enable Wireless Access in Vehicular Environment (WAVE). The standard specifies operation in the 5.9GHz frequency band. WAVE is composed of IEEE 802.11p and IEEE 1609.x. The IEEE 802.11p controls the physical layer and the medium access layer (PHY/MAC) while the IEEE 1609.x provides specifications for the control of upper layers. In the 1609.x family, IEEE 1609.3 specifies standards for transport and network layers. The 1609.4 documents specify standards for multi – channel operation. It is widely accepted in literature that in multi – channel operation, a WAVE system makes use of a single common control channel (CCH) and a number of service channels (SCHs) (Ren, 2019).

Cluster based data dissemination in VANETs is fast gaining prominence in the research space. This is because of the scalability and robustness that it brings into vehicular communication. Clustering is the grouping of nodes according to pre-defined parameters; closely related or similar objects are grouped together to form a cluster. Clustering is a major challenge in VANETs, primarily because of the frequent topology changes in the network. Achieving cluster stability and efficiency then becomes research worthy. This research therefore seeks to develop an enhanced weight-based cluster head selection algorithm which will use the suitability index of vehicles to achieve seamless cluster head selection process.

In this work, an enhanced weight-based cluster head selection algorithm for vehicular adhoc networks is proposed by using suitability index obtained through a weighted mechanism obtained using speed, transmission power and number of neighbors. This work is introducing transmission power as a basis for determining the weight which will help to decide which vehicle emerges as a cluster head.

1.2 Statement of the Research Problem

Determining the most suitable vehicle to serve as cluster head remains an open research issue in the research community. A good number of research works available in literature have advocated for fixed clustering approach which will require frequent cluster head selections. In recent times, interest has moved towards dynamic cluster head selection. One of such dynamic cluster head selection algorithms is the weight-based clustering algorithm. The weight-based clustering algorithm assigns weight to parameters based on their importance, this however does not take into cognizance the transmission power of the vehicles. A good algorithm like the one that is proposed in this work should take into consideration the transmission power of the vehicle at minimal frequency of cluster head (CH) selection.

1.3 Aim and Objectives

The aim of this work is to develop an enhanced weight-based cluster head selection algorithm that will reduce the frequency of cluster head election in VANETs. The following are the objectives of the research:

i. Formation of clusters using k-means algorithm and determination of cluster head using suitability index in MATLAB environment.

ii. Evaluation and validation of the algorithm developed in (i) based on stated KPIs which are Packet Delivery Ratio (PDR), End-to-End delay and Cluster head selection time.

1.4 Scope and Limitation of Study

This research seeks to investigate cluster head selection in vehicular communications using an enhanced weight-based cluster head selection algorithm. MATLAB will be used to simulate and evaluate the proposed enhanced weight-based cluster head selection algorithm in VANETs. The algorithm was then evaluated based on three important parameters, which are packet delivery ratio, end to end delay and cluster head selection time.

1.5 Thesis Outline

The rest of this thesis is outlined as follows: Chapter Two x-rays the state of research currently in clustering, types of clustering, cluster stability, parameters of interest in clustering, simulation tools for VANETs and cluster head selection techniques. Chapter Three contains the methodology while Chapters Four and Five specified the results obtained and conclusion as well as recommendations.

CHAPTER TWO

2.0 LITERATURE REVIEW

This section reviews relevant literature available in the research space. It brings to bare the extent and trend of research in recent times in the field of vehicular adhoc networks. As more user-centric applications are being developed to conform with VANETs, there is also corresponding increase in traffic, thus, in VANETs there are two types of traffic that are calling for attention, they are data traffic and road traffic. Both have become of interest in the research space. Controlling data traffic in times of the much-anticipated dense road traffic requires that optimized clustering schemes be adopted. This research therefore seeks to improve cluster stability by reducing the frequency of cluster head selection and by extension improving the stability of clusters through an enhanced weight-based cluster head selection algorithm with suitability table.

2.1 Vehicular Adhoc Networks (VANETS)

VANETs are a self-organizing network which consists of mobile nodes at high speed. They are typically characterized by frequent connections and disconnections. They are made up of such units as RSUs, OBU and very recently, they have an auxiliary unit (AU). The OBU consists of the electronic equipment needed for transmission and reception of packets; the RSU is the unit that aids VANETs in accessing remote services. Figure 2.1 shows the inter-vehicular communication and a vehicle to infrastructure connection.

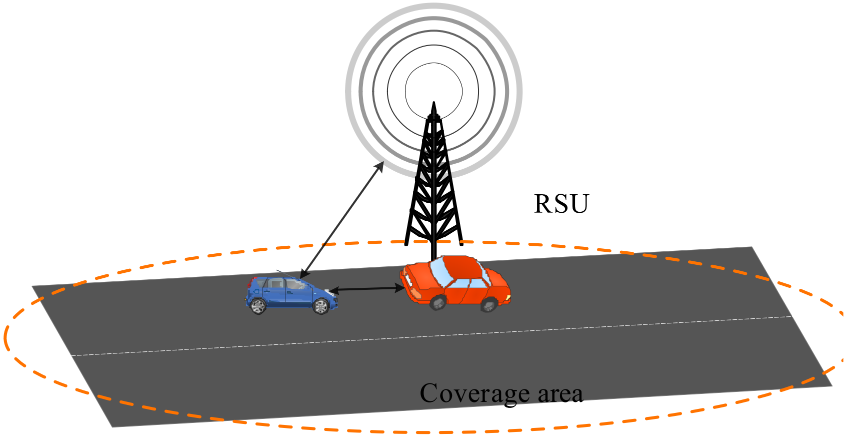


Figure 2.1: Vehicle to Vehicle to Infrastructure Communication

Figure 2.2 shows a typical VANET scenario and all the services available, this shows that much is required to evolve a scalable vehicular communication system. There is therefore need for some basic services to be within close proximity of the moving nodes hence the need for clustering and cluster heads.

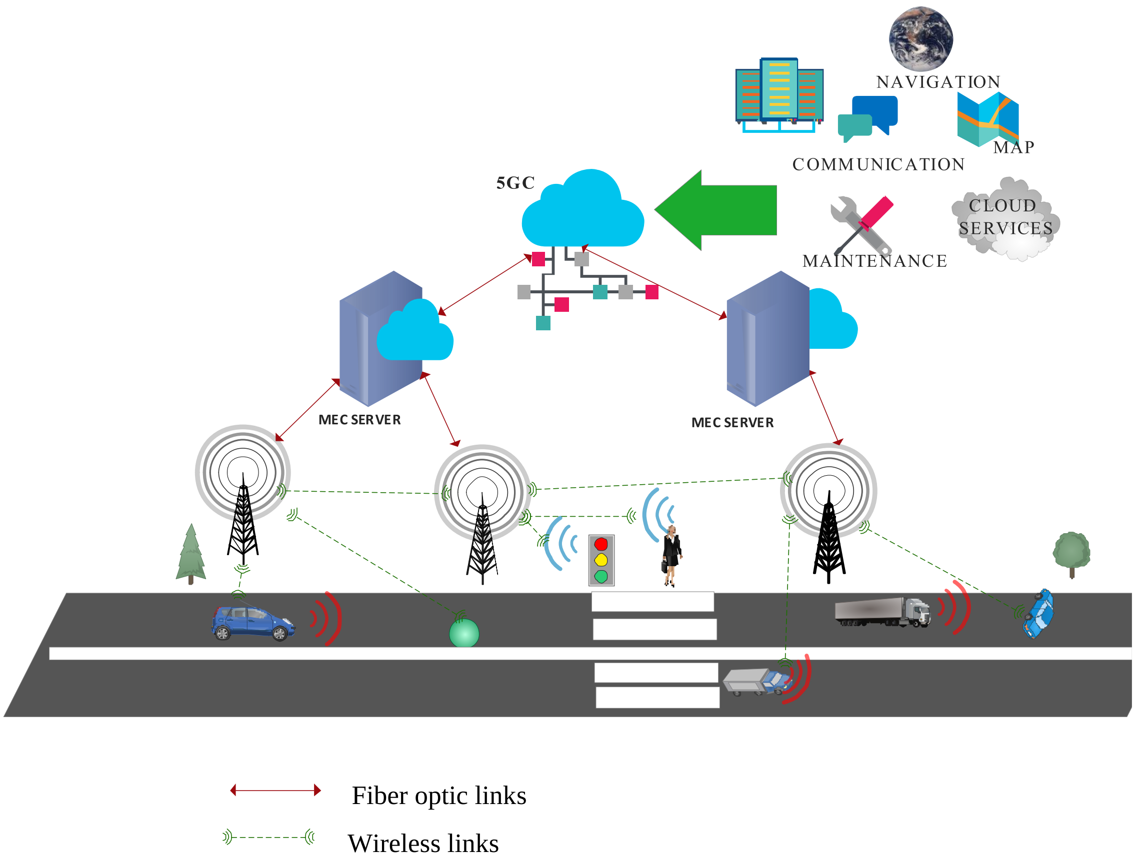


Figure 2.2: Typical VANET scenario.

2.2 Centralized and Distributed Networks

A centralized network is a type of digital communication network in which there is a central coordinating entity controlling the entire network. This coordinating entity might be a single server or even a node. The central entity is responsible for maintaining the network, managing user processes and setting policies and protocols for the network. A lot of digital platforms such as Youtube, Facebook and Twitter that we use today are centralized systems.

A decentralized system or network is one in which the control of activities of the network is not vested in a single entity. In the decentralized network, coordination of activities is distributed on a peer-to-peer basis with each user having equal or a measure of authority within the network since it is not controlled by one authority.

Centralized networks are used to achieve control and stability. It does not provide user freedom and collaboration as is obtainable in decentralized networks. Centralized systems are easy to maintain compared to decentralized systems. In decentralized networks, there is no single point of failure, but maintenance is difficult especially in large networks. This work chooses a centralized network because it is easy to maintain and less complex.

2.3 Clustering Techniques in Vehicular Adhoc Networks (VANETS)

Bevish (2015) described clustering as one of the measures applied to accommodate the frequent topology changes in VANETs. A lot of clustering techniques are available in literature, these clustering techniques employ different algorithms. Some common clustering algorithms are: prediction based clustering and highest-degree algorithm. Ahsan (2020) stated that VANETs employ two modes in their communication: centralized mode and distributed mode. According to Nimbalkar & Dass (2019) clustering is one of the measures applied to accommodate the frequent topology changes in VANETs. Stability in clustering is greatly desired in VANETs and this requires careful selection of cluster members and by extension cluster heads.

Clustering provides for effective MAC scheduling, prevention of network flooding and creation of effective data dissemination. The quality of a cluster is quantified by its stability, efficiency and consistency. Cluster heads enable effective routing. The stages in the lifetime of a cluster are: cluster formation, cluster maintenance and cluster dissolution.

Clustering is achieved with the aid of some information. The information used for clustering are generally categorized viz: topology information, mobility information and contextual information. Figure 2.3 shows a typical clustering scenario with tables of information required for the formation and maintenance of clusters.

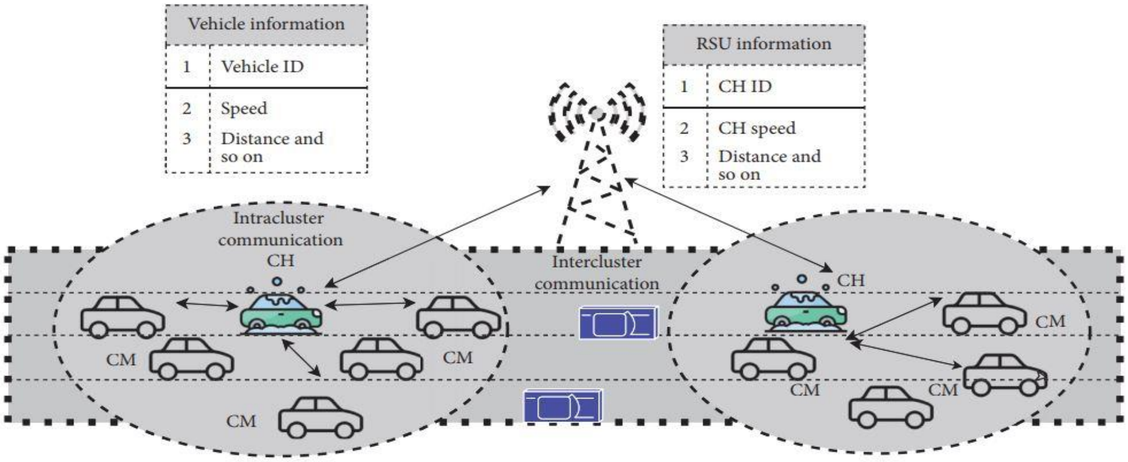


Figure 2.3: Clustering in VANET
Source: Saleem *et al* (2021).

2.4 Common Terminologies in VANET Clustering

In the study of clustering in VANETs, there are terminologies of interest which has come to constitute the body of research in VANETS clustering. Some terminologies in VANETS are:

1. Cluster Centroid: this is the vehicle whose position on the average is equidistant from all the other nodes in the cluster. The cluster centroid also known as cluster center is the mid-point or center of gravity of the cluster.
2. Cluster head: the cluster head as previously described is the vehicle which is most suitable based on defined parameters to serve as a routing station for the cluster. The cluster head coordinates communication among the cluster members and with

other cluster heads. A cluster head must have a good view of its network and other networks along its trajectory.

3. Cluster Stability: this is the length of time that a cluster exists without change in topology or configuration. For achievement of stability, clusters are formed by giving due credence to direction of travel.
4. Cluster formation time: this the time it requires for a cluster to be formed. Cluster formation time depends on the clustering algorithm. It is good practice to keep the cluster formation time at minimum. Algorithms for cluster formation time must keep the cluster formation time at minimum.
5. Cluster life time: this is the time it takes for a cluster to be active or functional. The stability of a cluster will be determined by the cluster lifetime.

2.5 Clustering Protocols

Clustering protocols are those techniques that have been evolved over time for the purpose of cluster formation and maintenance. Clustering protocols can be combined for specific purposes or based on performance metric of interest. Figure 2.4 shows the broad categorizations of these protocols, although there are variants of some of these protocols available in literature, but this classification covers all of them. Clustering protocols are chosen based on objectives set to be achieved in a given instance and will mostly depend on the layer of interest. The most dominant layer in literature at which clustering is being done is the application layer.

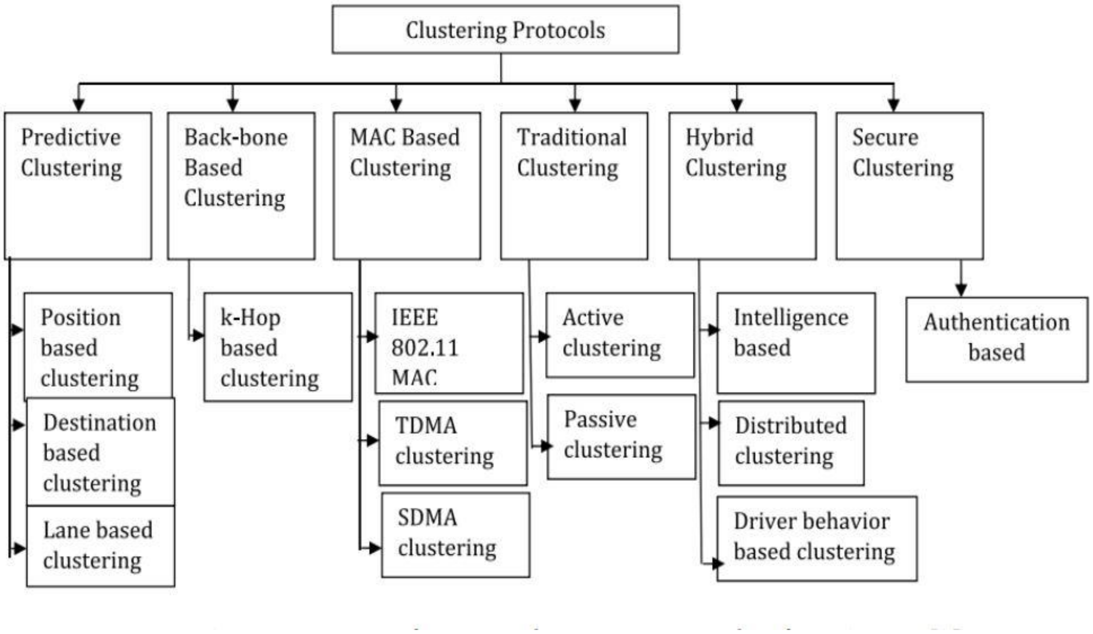


Figure 2.4: Clustering Protocols
Source: (Busari *et al;* 2019)

2.5.1 Predictive clustering

Predictive clustering is a clustering protocol in which the current position and the future behavior of nodes is used to form clusters, most common predictive clustering algorithms include lane based, destination-based clustering and position-based clustering.

2.5.2 Backbone based clustering

In backbone clustering, one of the nodes is chosen as a communication back bone. This back bone is useful in the process of cluster head selection. K-hop or multi hop clustering are examples of back bone clustering techniques.

2.5.3 Mac based clustering

The MAC based clustering makes use of 802.11 MAC protocol in the formation of clusters. It includes IEEE 802.11 MAC, Time Division Multiple Access (TDMA) as well as Spatial Division Multiple Access (SDMA) clustering techniques.

2.5.4 Traditional clustering

This clustering technique is based on the nature of vehicles; it is based on passive and active behavior associated with vehicles. Active clustering is further divided into mobility based, beacon based, density based and dynamic behavior based clustering.

2.5.5 Hybrid clustering

In this technique, more than one technique such as fuzzy logic, machine learning and artificial intelligence are used in the formation of clusters. It is sub divided into distributed, driver behavior and intelligence based clustering.

2.5.6 Secure clustering

Secure clustering involves the use of security parameters to establish clusters in the vehicular network. Certificate authentication and trusted authority are usually employed in this clustering approach. Authentication based clustering falls in this category.

2.6 Cluster Head Selection

Cluster head selection algorithms are gaining more attention in VANETs. The purpose of clustering is to segment communicating nodes according to some features such as speed, direction of travel, and communication range. Clustering in VANETs is quite complex due to frequent topology changes; hence several factors have to be considered when clustering in VANETs.

In the work of Ahsan (2020), an optimized node clustering algorithm in VANETs was developed by using meta-heuristic techniques. This algorithm used parameters such as node’s direction, communication link capacity, network area, node density and transmission range. The algorithm is based on the Grasshopper Optimization Algorithm (GOA). The algorithm mathematically modeled the swarming behavior of grasshoppers; however, this algorithm is more suitable on wider roads with high vehicular traffic.

Husnain & Shahzad (2021) proposed an intelligent cluster optimization algorithm based on whale optimization algorithm for VANETs. In the framework, an intelligent clustering approach was used to optimize the routing of packets in VANETs. The algorithm is however, complex and several analyses are required to compute the performance metrics. The work of Nimbalkar & Dass, (2014), surveyed the various cluster head selection techniques based on fuzzy logic, neural network and genetic algorithm. This work did not consider other algorithms apart from machine learning techniques. Cluster head selection algorithms should be implemented even without machine learning techniques. In Karthikeyan (2021), an adaptive clustering algorithm for stable communication in VANETs was proposed. The algorithm combined weight based and neuro-fuzzy prediction by developing a zone based clustering and a k time zone based clustering, the work does not take to cognizance the frequent topology changes in VANETs.

The scheme of Duan (2016) introduced a Software Defined Networking (SDN) programmable network structure as an enabling platform to apply intelligence and control in 5G-VANET HetNet. The SDN controller has a global view of the HetNet so as to be able to execute clustering only when needed. The dual cluster design also guarantees seamless end-user data access especially when there is Cluster Head (CH) service disruption.

In a related development, Talib (2017) proposed a center-based stable evolving clustering algorithm with grid partitioning and extended mobility features for VANETS. The article proposed a V2I based clustering framework in VANETs using a modified evolving clustering algorithm with adoption of the concept of the grid in VANETS clustering for the first time. It has developed a novel traffic generator that includes in addition to driving behavior, a novel lane changing probabilistic model. It proposes grid partitioning for the road environment before doing clustering, which makes it suitable for high density highways. It also proposes an extended mobility feature that combines in addition to relative position and velocity of vehicles, a relative acceleration which makes the clustering more dynamically aware of higher moments when mobility variables can be added. (Khayat *et al.*, 2020) proposed a VANETS clustering based on weighted trusted cluster head selection, this technique proposed a new clustering protocol with a unique cluster head selection process while still retaining the features of VANETS clustering. The cluster head selection in this protocol is based on a weighted formula. Table 2.1 summarizes some related literatures in cluster head selection.

Table 2.1: Some Related Literatures.

Author & Date	Title	Contribution	Limitations
Talib *et al.* (2017)	A Center-based Stable Evolving Clustering Algorithm with Grid Partitioning and Extended Mobility Features for VANETs	Developed a center-based approach for clustering vehicles based on their physical location alone	Physical location alone has proven not to be sufficient for clustering to be done.
Khayat *et al* (2020)	VANET Clustering Based on Weighted Trusted Cluster Head Selection	Developed a cluster head selection algorithm based on weight of vehicles and mobility parameters	Will have to require frequent cluster head election when a cluster head exits the cluster
Shahid (2019)	Fixed Cluster Based Cluster Head Selection Algorithm in Vehicular Adhoc Network	Developed a weight-based algorithm which depends on predefined trust values of vehicles	Repeated cluster election
Saleem *et al* (2021)	Deep Learning-Based Dynamic Stable Cluster Head Selection in VANET	Developed a deep learning algorithm for cluster head selection with emphasis on clusters stability	No consideration for eventual departure of cluster head.
Hafeez *et al* (2012)	A Fuzzy-Logic-Based Cluster Head Selection Algorithm in VANETs	Developed a fuzzy based clustering algorithm based on speed and position of vehicles.	Speed and position alone have proven not to be sufficient for clustering in VANETs

Duan *et al* (2016)	SDN Enabled Dual Cluster Head Selection and Adaptive Clustering in 5G-VANET	Developed a robust dual cluster head selection algorithm for the purpose of reducing overhead.	Cognizance was not taken of the overhead re-introduced when both cluster heads exit the cluster.
Bijalwan *et al* (2022)	A Self-Adaptable Angular Based K-Medoid Clustering Scheme (SAACS) for Dynamic VANETs.	Developed a K-medoid clustering scheme for dynamic VANETs considering overhead and stability as the main challenges.	Although the cluster head is selected dynamically, cluster members have to keep comparing their weights to the current cluster head without consideration for other cluster members.
Mayank *et al* (2023)	Weight-Based Clustering Algorithm for Military Vehicles Communication in VANET	Developed a weight-based clustering algorithm for military vehicles by using real time average and degree to determine the cluster head in a Rhombus network	Rhombus networks are not typical everyday networks

From all the reviewed literature, it is evident that cluster head selection algorithm that reduces overhead caused by frequent cluster head selection needs to be developed. This will also bring about stable clusters in the long run. The cluster head selection algorithm employed in this work emphasizes the use of table of suitability index by which every vehicle knows its position on the log and automatically assumes headship when the cluster head exits the cluster without any need for another selection. The clustering technique used here also allows for re-cauterization as the nodes increase so as to keep network performance in optimal state.

CHAPTER THREE

3.0 MATERIALS AND METHODS

3.1 Simulation Tools for Vehicular Networks

Simulation is the process of synthesizing or modeling real life scenarios using computer related tools. Simulation is cheap compared to building or developing prototypes. There are quite a number of simulation tools for VANETs available. They are either open source or commercial.

In Ahmed *et al.* (2019), a number of simulators available for vehicular networks were highlighted. Some of them are: VEINS, VsimRTI, VENTOS, EstiNet, iTERRIS, TraNS, NCTUns, Groovenet, MobiReal. The strength and weaknesses of each of the simulation tools were also highlighted. Kit *et al.* (2016), also described some mobility simulators such as VISSIM, Quadstone Paramics and simulation of urban mobility (SUMO). MATLAB was also used together with SUMO to implement vehicular communication. In Apratim *et al.*, (2016), vehicular communication simulators were grouped into: Traffic modeling, traffic management and communication network simulation. The vehicular traffic is modelled with the aid of simulators such as VISSIM, the traffic management is achieved by simulators such as MATLAB while NS 2 or 3 are used to simulate the communication networks. There are also proprietary VANET simulators, some of them are: TSIS-CORSIM, Paramics, Daimler-Chysler Farsi, Videlio, Carisma, Qualnet and OPNET. In Figure 3.1, a flow of signal is depicted. Simulation of VANETs can be achieved in an integrated environment as demonstrated in (Francisco *et al*. 2011; Kit *et al*. 2016; Bryan *et al*. 2020). The traffic generator, mobility generator and network simulator constitute the

VANET simulation set up. Traffic and mobility information is conveyed to the network simulator by means of the middleware.

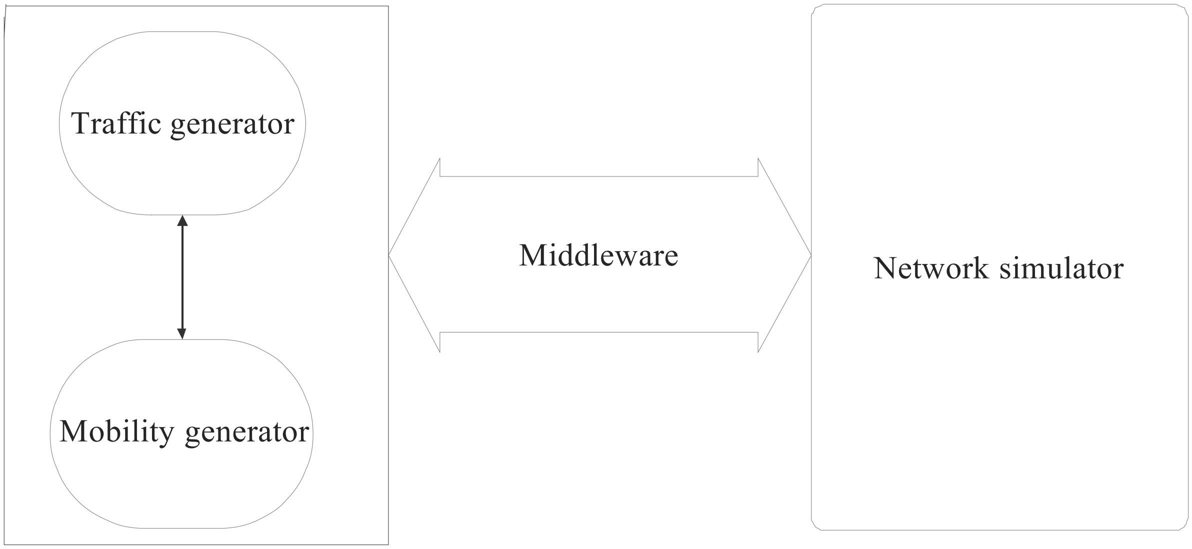
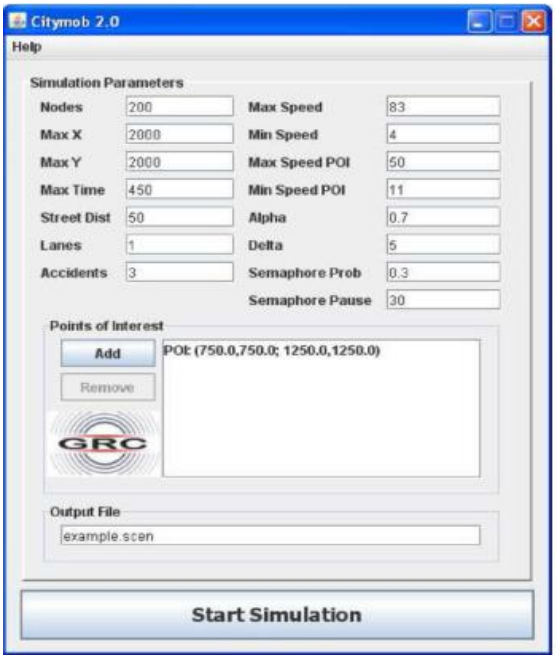


Figure 3.1: The Flow of Signal in a VANET set-up

3.1.1 Mobility generators

Mobility generators are used to develop highly realistic road traffic models in vehicular networks (VANETs) simulations. Mobility generators generate close to real life traffic scenarios or traces which are used as input for network simulators/emulators. Mobility generators' inputs are: road model, traffic parameters such as maximum velocity, vehicle arrival and departure rates. The mobility generators, produce trace information of every vehicle at any given time during the simulation period. Mobility profiles of vehicles are also provided by mobility generators. Mobility generators are able to implement street maps to enable simulations to be very similar to real life scenarios. Some mobility generators are SUMO, MOVE, City Mob, Freesim, VANET Mobisim, STRAW, Netstream. Figure 3.2(a), (b) and (c) show some commonly used mobility generators. Mobility generators allow for parameters such as street, number of lanes, map travel time to be set. A good number of mobility generators such as SUMO and City mob have graphical user interface (GUI), this allows for easy set up. In Midya *et al*. (2018), SUMO was used together with NS3.26 for implementing a cluster-based handover in vehicular communication. SUMO was used to generate mobility while NS3.26 was used to implement the handover between any two mobile access gateway (Zhou *et al*. 2018). Similarly, Oladosu *et al*. (2019) implemented an optimized handover algorithm in NS 2.35 equipped with 802.11b/g. NS2 has the advantage of being compatible with different operating systems and not just Linux. It can be seen in Apratim *et al.,*(2016), and Gyawali *et. al,* (2020) where integrated simulation environments were designed and tested.

In this work, a standalone MATLAB environment was used to simulate vehicular traffic. This does not require any form of software integration and is less complex for an average user.



Nodes 200
Max X 2000
Max Y 2000
Max Time 450
Street Dist 50
Lanes 1
Accidents 3
Max Speed 83
Min Speed 4
Max Speed POI 50
Min Speed POI 11
Alpha 0.7
Delta 5
Semaphore Prob 0.3
Semaphore Pause 30
POI: (750.0,750.0; 1250.0,1250.0)
example.scen

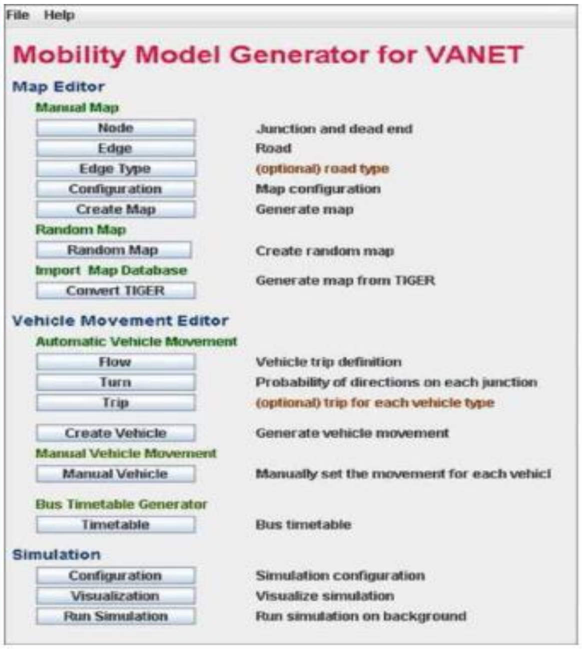


Figure 3.2(a) Citymob 2.0 Environment   Figure 3.1 (b) Mobility Model generator

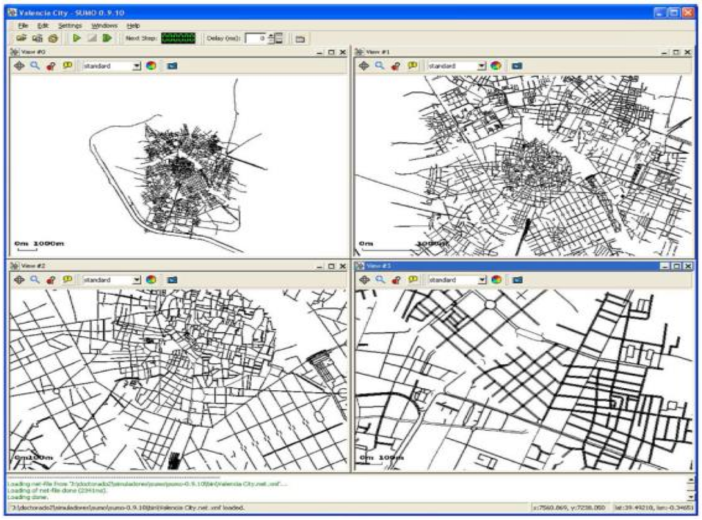
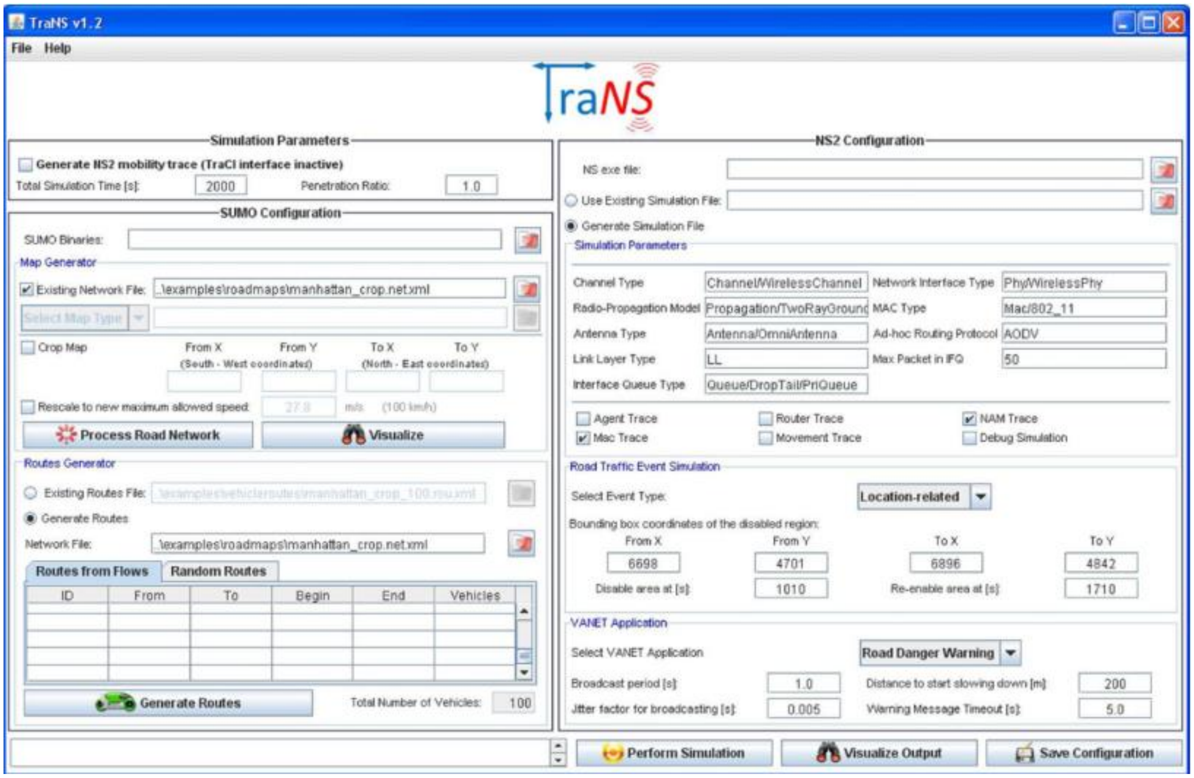


Figure 3.2 (c) Simulation of Urban Mobility (SUMO)
Figure 3.2 (a) – (c) The interface of some VANET simulators.
(Source: Francisco *et al.*, 2011)

3.1.2 Network simulators

Network simulators implement simulation of packets of data traffic. Network simulators execute source to destination transmission and reception of packets. Network information such as links, routes and channels are implemented by network simulators. The network simulators available were initially meant for Mobile Adhoc Networks (MANETs), hence, they require VANETs extension like mobility generators to enable them simulate vehicular networks. Examples of network simulators include: NS2, NS3, SNS, TraNS, GloMoSim, JiST/SWANS and GTNets. Figure 3.3 is a sample Graphical User Interface (GUI) of a network simulator, it specifies the mobility trace interface (TRACI) parameters and network parameters (Apratim, 2016).



☐ Generate NS2 mobility trace (TraCI interface inactive)
Total Simulation Time [s]: 2000
Penetration Ratio: 1.0
SUMO Binaries:
☑ Existing Network File: ..\examples\roadmaps\manhattan_crop.net.xml
☐ Crop Map
From X, From Y (South - West coordinated); To X, To Y (North - East coordinated)
☐ Rescale to new maximum allowed speed
○ Existing Routes File:
● Generate Routes
Network File: ..\examples\roadmaps\manhattan_crop.net.xml

ID	From	To	Begin	End	Vehicles

Total Number of Vehicles: 100

NS exe file:
○ Use Existing Simulation File:
● Generate Simulation File
Channel Type: Channel/WirelessChannel
Network Interface Type: Phy/WirelessPhy
Radio-Propagation Model: Propagation/TwoRayGround
MAC Type: Mac/802_11
Antenna Type: Antenna/OmniAntenna
Ad-hoc Routing Protocol: AODV
Link Layer Type: LL
Max Packet in IFQ: 50
Interface Queue Type: Queue/DropTail/PriQueue
☐ Agent Trace
☐ Router Trace
☑ NAM Trace
☑ Mac Trace
☐ Movement Trace
☐ Debug Simulation
Select Event Type: Location-related
Bounding box coordinates of the disabled region:

From X	From Y	To X	To Y
6698	4701	6896	4842

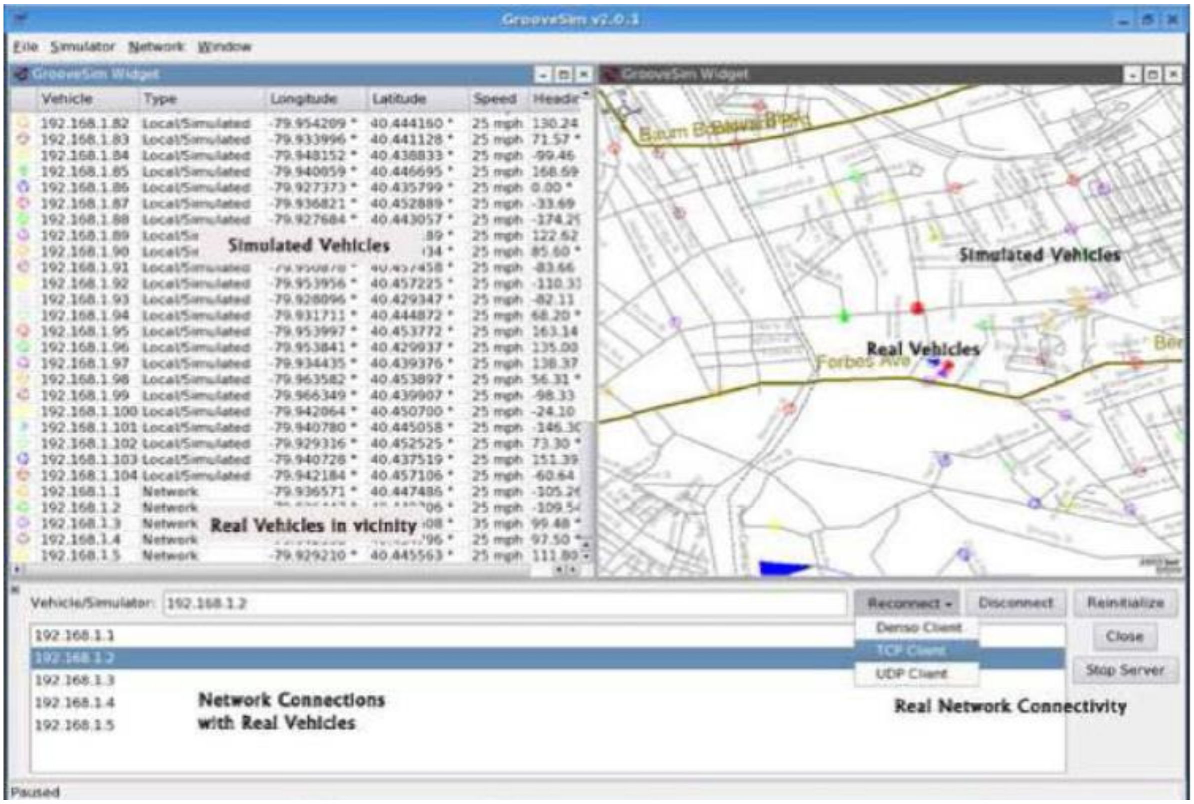
Disable area at [s]: 1010
Re-enable area at [s]: 1710
Select VANET Application: Road Danger Warning
Broadcast period [s]: 1.0
Distance to start slowing down [m]: 200
Jitter factor for broadcasting [s]: 0.005
Warning Message Timeout [s]: 5.0

Figure 3.3: TraNS network simulator.
(Source: Google images)

3.1.2 VANET Simulators

VANET Simulation software provides network and road traffic simulation. VANETs enable integration of mobility generators and network simulators on a single platform.

Before now, traffic simulation and network simulation were implemented in two separate simulators, however, in recent times, simulation of both traffic and network is carried out on VANET simulators. Some VANET simulators as cited in Francisco (2011) and Kit *et al*. (2016) are: Traffic and network simulation environment (TraNS), GrooveNet, National Chiao Tung University Network Simulator (NCTUns). In figure 3.4, a GUI of Groovenet is shown, it has both mobility trace function and network simulation function. It also shows the IP addresses of the simulated vehicles.



Vehicle	Type	Longitude	Latitude	Speed	Headi
192.168.1.82	Local/Simulated	-79.954209 °	40.444160 °	25 mph	130.24
192.168.1.83	Local/Simulated	-79.933996 °	40.441128 °	25 mph	71.57 °
192.168.1.84	Local/Simulated	-79.948152 °	40.438833 °	25 mph	-99.46
192.168.1.85	Local/Simulated	-79.940059 °	40.446695 °	25 mph	168.69
192.168.1.86	Local/Simulated	-79.927373 °	40.435799 °	25 mph	0.00 °
192.168.1.87	Local/Simulated	-79.936821 °	40.452889 °	25 mph	-33.69
192.168.1.88	Local/Simulated	-79.927684 °	40.443057 °	25 mph	-174.2
192.168.1.89	Local/Si	[illegible]	89 °	25 mph	122.62
192.168.1.90	Local/Si	[illegible]	34 °	25 mph	85.60 °
192.168.1.91	Local/Simulated	[illegible]	[illegible]458 °	25 mph	-83.66
192.168.1.92	Local/Simulated	-79.953956 °	40.457225 °	25 mph	-110.3
192.168.1.93	Local/Simulated	-79.928096 °	40.429347 °	25 mph	-82.11
192.168.1.94	Local/Simulated	-79.931711 °	40.444872 °	25 mph	68.20 °
192.168.1.95	Local/Simulated	-79.953997 °	40.453772 °	25 mph	163.14
192.168.1.96	Local/Simulated	-79.953841 °	40.429937 °	25 mph	135.00
192.168.1.97	Local/Simulated	-79.934435 °	40.439376 °	25 mph	138.37
192.168.1.98	Local/Simulated	-79.963582 °	40.453897 °	25 mph	56.31 °
192.168.1.99	Local/Simulated	-79.966349 °	40.439907 °	25 mph	-98.33
192.168.1.100	Local/Simulated	-79.942064 °	40.450700 °	25 mph	-24.10
192.168.1.101	Local/Simulated	-79.940780 °	40.445058 °	25 mph	-146.3
192.168.1.102	Local/Simulated	-79.929316 °	40.452525 °	25 mph	73.30 °
192.168.1.103	Local/Simulated	-79.940728 °	40.437519 °	25 mph	151.39
192.168.1.104	Local/Simulated	-79.942184 °	40.457106 °	25 mph	-60.64
192.168.1.1	Network	-79.936571 °	40.447486 °	25 mph	-105.2
192.168.1.2	Network	[illegible]	[illegible]06 °	25 mph	-109.5
192.168.1.3	Network	[illegible]	[illegible]08 °	35 mph	99.48 °
192.168.1.4	Network	[illegible]	[illegible]96 °	25 mph	97.50 °
192.168.1.5	Network	-79.929210 °	40.445563 °	25 mph	111.80

Vehicle/Simulator: 192.168.1.2

192.168.1.1
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5

Figure 3.4: Groovesim VANET Simulator
(Source: Google Images)

The cluster head is the acting backbone for its cluster. The cluster head is responsible for network management and coordination of activities of member nodes in its cluster. The vehicles are clustered based on their direction and velocity. K means clustering was used in this work because of its ease of implementation on MATLAB.

The cluster consists of a cluster head. The cluster head is responsible for coordinating communication within the cluster and with the external infrastructure. In this work, the cluster head is chosen based on four criteria, namely:

i. Communication range
ii. Speed vehicles
iii. Number of neighbors
iv. Transmission power

Every cluster member has a cluster number or position. In the event that the cluster head fails or leaves the cluster, the next node in terms of the aforementioned criteria automatically assumes the position of the cluster head.

Given a set of vehicles, $X = (x_{1,}\ x_{2,}\ x_{3 \ldots\ldots} x_n)$, there exists a set of vehicles that meet the requirements for cluster heads, this set is represented as:

$X_h = \{x_{hi}\}$ this set signifies the cluster head (h) of the ith cluster.

The ith cluster is represented as a set of vehicles named

$C_i = \{x_1, x_2, x_{3\ldots}\}$ which is a subset of Y with velocities $\{v_1, v_2, v_3\ldots..\}$ the following assumptions are made:

1. The vehicles are equipped with GPS which enables them to know their location relative to their trajectory.

2. The vehicles are moving in the same direction as specified by lane numbers. Vehicles with the same lane number are moving in the same direction.
3. The road traffic density range is 40-70 vehicles. The average velocity of the cluster is dependent on the specification of the group of vehicles.
4. The vehicles are equipped with 802.11p interface for direct short-range communication (V2V) and wireless interface for (V2I) communication.
5. The first vehicle to send hello messages is chosen as the cluster center. The cluster center is the vehicle around which the cluster is created.
6. All vehicles forming the cluster run the proposed algorithm.
7. The algorithm sits in the application layer of the OSI reference model.

The vehicles are clustered based on their velocities. Vehicles with velocities within the vicinity of the mean velocity can constitute a cluster. The cluster boundary is specified by the distance of other vehicles from the cluster center. The vehicles that come together to form a cluster send hello messages to vehicles within its transmission range R_{max}. Vehicles within that range that reply the hello messages sent and do not belong to any cluster, begin cluster formation while those belonging to a cluster will ignore the hello message received.

3.2 Number of Neighbors

The number of neighbors at any time t of a vehicle x_i is the number of vehicles within its transmission range R_{max}. The number of neighbors N at any time depends on (3.1).

$$\sum_{j=1}^{k} d(i,j,k) \leq R_{max} \tag{3.1}$$

3.3 Mean Speed

In this work, the speed of the vehicles is assumed to have a Gauss/Random distribution with probability density function (pdf) expressed as (3.2).

$$f(v) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{1}{2}\left(\frac{(x_v - \mu_v)^2}{\sigma^2}\right)\right]} \qquad \sigma > 0 \tag{3.2}$$

Where μ and σ are the mean and the standard deviation respectively while $\frac{1}{\sigma\sqrt{2\pi}}$ is a constant factor that makes the area under the normal distribution curve equal to 1. The internodal distance between any two vehicles is given by equation (3.3).

$$D_{a,b} = \sqrt{(X_b - X_a)^2 + (Y_b - Y_a)^2} \le R_{max} \tag{3.3}$$

The mean velocity $\mu_v = \frac{1}{N}\sum v_i$ or $\mu_v = \sum_{j=1}^{m} \frac{\Delta d}{\Delta t}$ (3.4)

And the normalized velocity is expressed as:

$$v_n = \frac{x_v - \mu_v}{\sigma} \tag{3.5}$$

3.4 Mean Distance

By using the Euclidean distance, the mean distance between the nodes is given by:

$$\mu_d = \frac{\sum_{j=1}^{m} D_{a,b}}{N(t)} \tag{3.6}$$

$$d_{normal} = \frac{n_p - \mu_d}{\sigma_d} \tag{3.7}$$

3.5 Weight Computation

The suitability value of a vehicle is computed based on its aggregate weighted value in terms of the parameters discussed. Each node computes its mean distance μ_d.

$$\mu_d = \frac{\sum D_{AB}}{N_i\,(t)} \tag{3.8}$$

The aggregate weight of each vehicle is computed as follows:

$$\alpha_i = W_1 * N_i(t) + W_2 * v_{normal} + W_3 * f(v)_{normal} + W_4 * P_{TX} \tag{3.9}$$

Subject to:

$$W_1 + W_2 + W_3 + W_4 = 1 \tag{3.10}$$

where:

W_1, W_2, W_3 and W_4 are the weights associated with each parameter.

P_{TX} = transmission power of the vehicle

α_i = the weight of each vehicle *i*

To be able to form a table of suitability value in ranking order, a vehicle computes its position by the equation

$$\beta_i = \left(\sum \alpha_n\right) - \alpha_i \tag{3.11}$$

β_i = suitability index of vehicle i

Mathematically, the smaller the value of β, the better the position of the vehicle in the ranking this is because equation (3.11) makes it clear that when a large β value is taken away from the total sum of β values, the outcome is a smaller value. Vehicles with smaller suitability values have good chances of appearing at the top of the table while vehicles with high suitability values do not have a good chance of appearing at the top of the suitability value table.

3.6 Packet Delivery Ratio

The packet delivery ratio PDR is the ratio of total packet received by each node to the total number of packets sent by the cluster head. It is expressed as (3.12).

$$PDR = \frac{TP_i}{TP_{CH}} \tag{3.12}$$

Where TP_i is packet received by vehicle i and TP_{CH} is total packet sent by the cluster head

3.7 Cluster Head Formation Time

Cluster head formation time is the measurement of time taken to successfully choose a cluster head from among the vehicles in a cluster.

3.8 Average End-to-End Delay

The end-to-end delay is the difference between the transmission times from the transmitting end to the reception time at the receiving end. It is given as (3.13).

$$Delay = \frac{(t_{r-}\, t_r)}{n} \tag{3.13}$$

where t_r the reception time, t_t is the transmission time and n is the total number of packets sent from the transmitting vehicle.

3.9 Flow Chart

The flow chart for the proposed algorithm is presented in Figure 3.4. It shows the flow of command from cluster formation to cluster head selection. This work is specific about cluster head selection which is a backbone of cluster maintenance, hence the cluster head selection greatly helps in cluster maintenance.

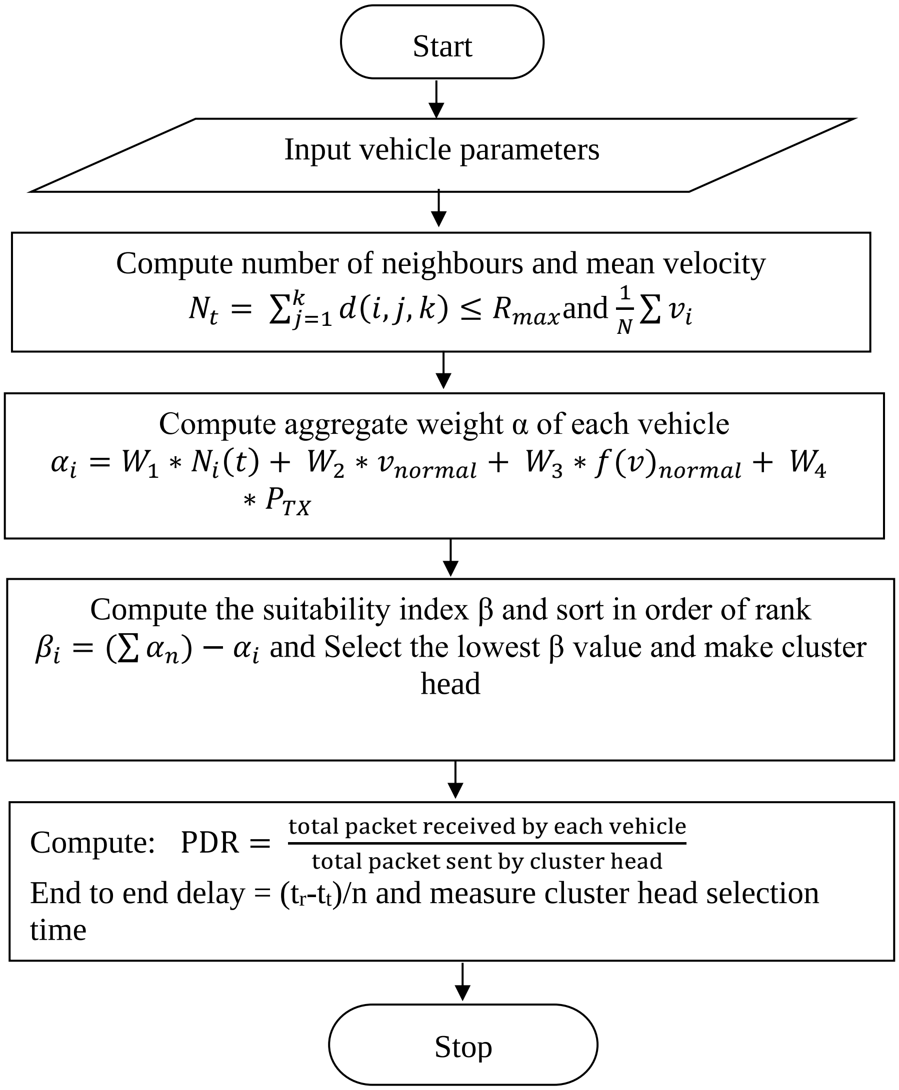


Figure 3.4 Flow Chart for Enhanced Weight-based Cluster Head Selection Algorithm.

3.10 Clustering procedure

```
i.     Initialize speed, direction, transmission power, and position
ii.    Check for available clusters from hello messages received
iii.   If there are more than one clusters,
iv.    {
v.         choose most suitable cluster and join
vi.    }
vii.   Else,
viii.  Initiate cluster formation
ix.    Compare speed of vehicles within range
x.     Compute average speed and pdf
xi.    Begin cluster head selection
xii.   compute weight
xiii.  compute suitability index
xiv.   Select cluster head
```

The detailed procedure for the cluster-head selection algorithm can be found in appendix B.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

The enhanced weight-based cluster head selection algorithm is implemented based on the algorithm in Chapter three. Figure 3.1 depicts a bidirectional road with vehicles moving in either direction. From the observations in the simulations, it is seen that vehicles with close probabilities are represented by same shapes. Different vehicle densities in the range of 40 to 70 will then be used to study the algorithm developed in this work, this is because this algorithm is designed and tested for 40,50,60 and 70 vehicles. Different parameter combination has been used in previous works such parameters include velocity, buffer size of vehicles among several other parameters. In this work, the transmission power is added as a basis for determining the index of vehicles for the table of suitability values used in the selection of cluster head and this is the distinction between this work and existing works. The weight-based technique can then be improved upon using the proposed approach in this work. The results in this work studies how road traffic density affects some network KPIs. The clustering and cluster head selection is shown in Figure 4.0 by a road of dimension Y x X. The simulation parameters are stated in appendix A.

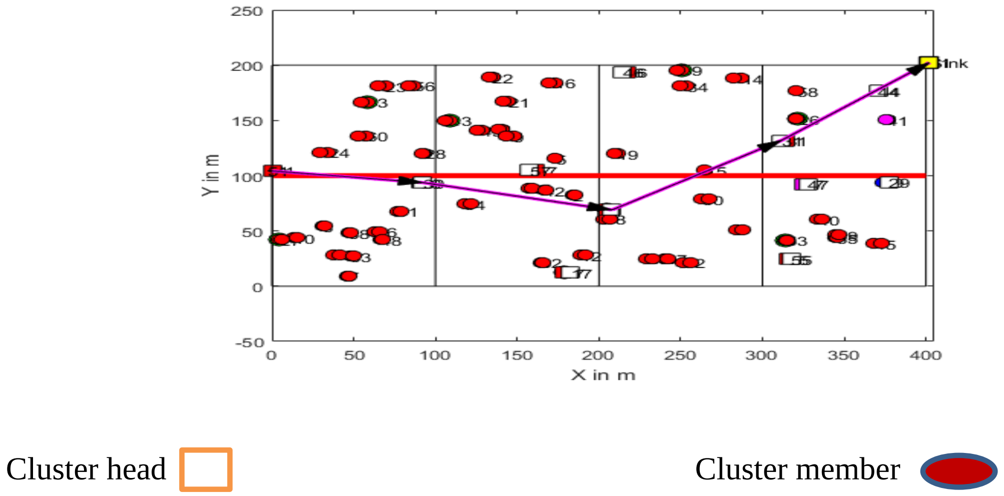
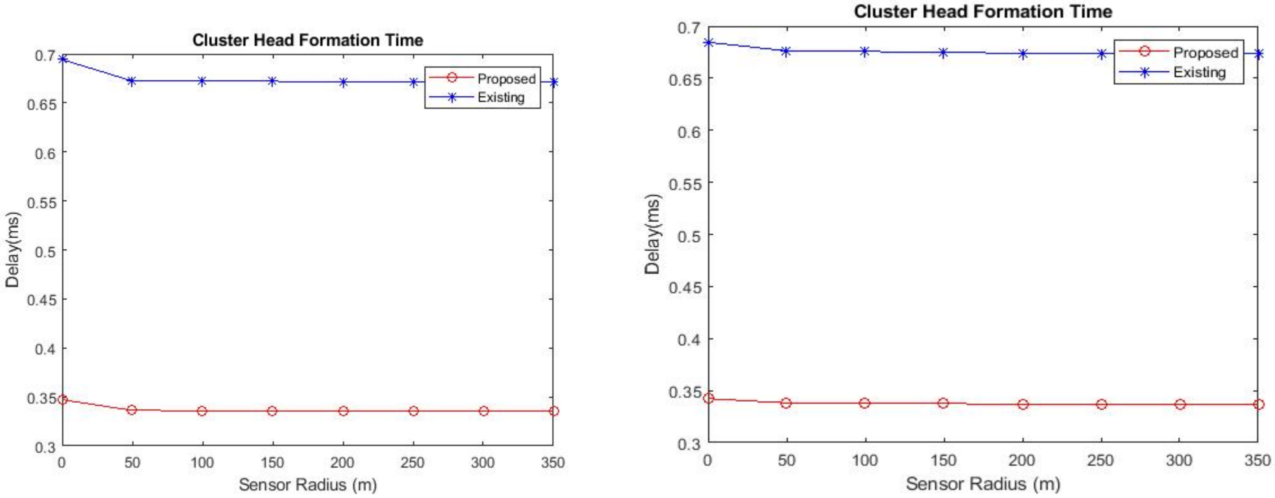


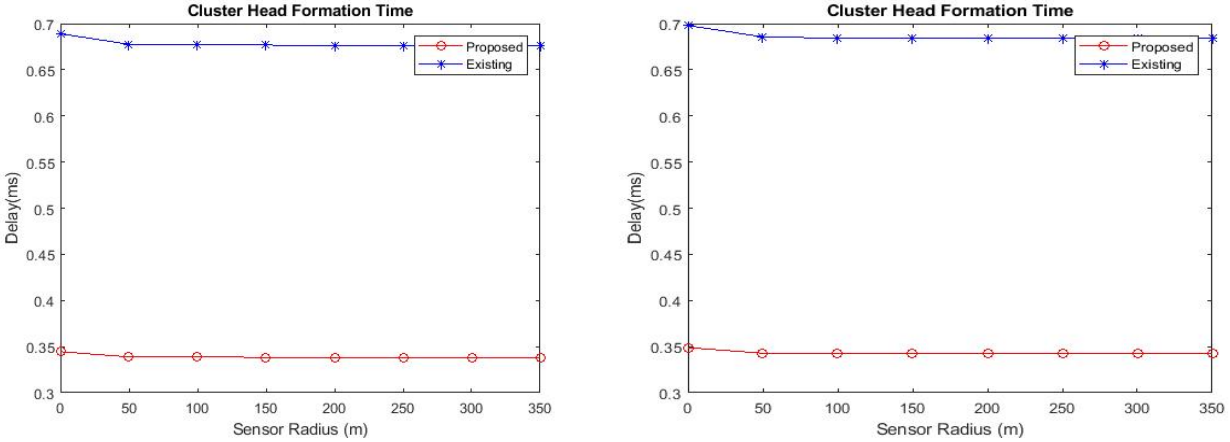
Figure 4.0: Clustering and cluster head selection

4.1 Cluster Head Selection Time

The cluster head formation time is the time it takes to choose a cluster head from among the vehicles as previously described. Comparing the cluster head selection for the existing and proposed algorithm, we see a significant improvement in the cluster head selection/formation time, the range in the existing algorithm is quite larger than that of our proposed model for different sensor radius. While the existing work has a range of 0.04, this work has 0.01. There was however no significance difference in pattern when the algorithm was tested against the existing one for 40, 50, 60 and 70 nodes. This suggests that for cluster head formation or selection delay, the effect of variation in sensor radius does not hold much significance with increase in vehicular nodes. At lover vehicle density, the graph appears less chaotic and stabled, this is because, with fewer nodes, selection will always be faster. At 40 nodes, the proposed algorithm outperformed the existing scheme by 51%. On further simulations with 50, 60 and 70 nodes, the margin of improvement were 49%, 49%, and 51% respectively. This shows that cluster head formation delay will vary based the internodal distance and node densification. Higher inter nodal distance and higher node density will increase cluster head selection delay. However, because of reduction of cluster size with increase in node density, cluster head selection time performance against the existing scheme for 70 nodes returns to 51% . The cluster head selection time is more optimal for this algorithm within 50 to 60 nodes in comparing the two algorithms.



(a) For 40 nodes (b) For 50 nodes

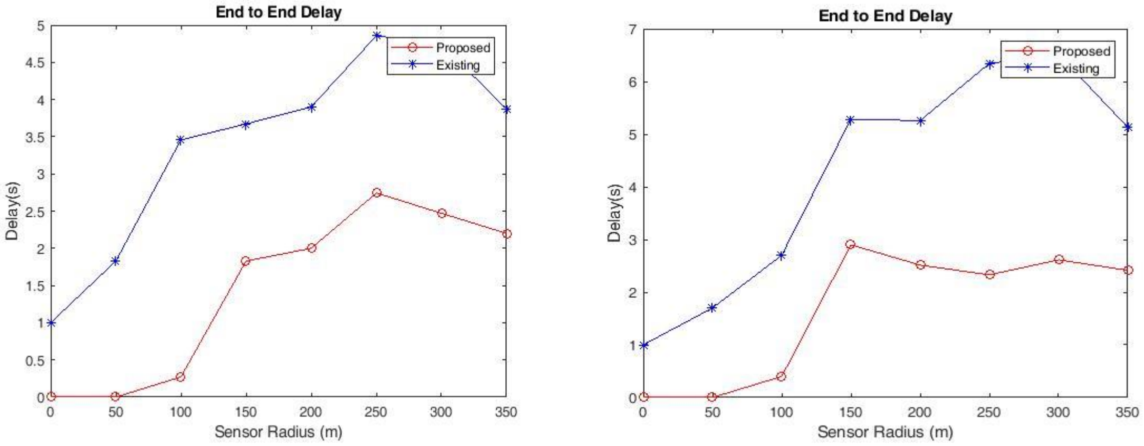


(c) 60 nodes (d) 70 nodes

Figure 4.1 a-d: Cluster Head Formation Time versus Sensor Radius for varying number of nodes

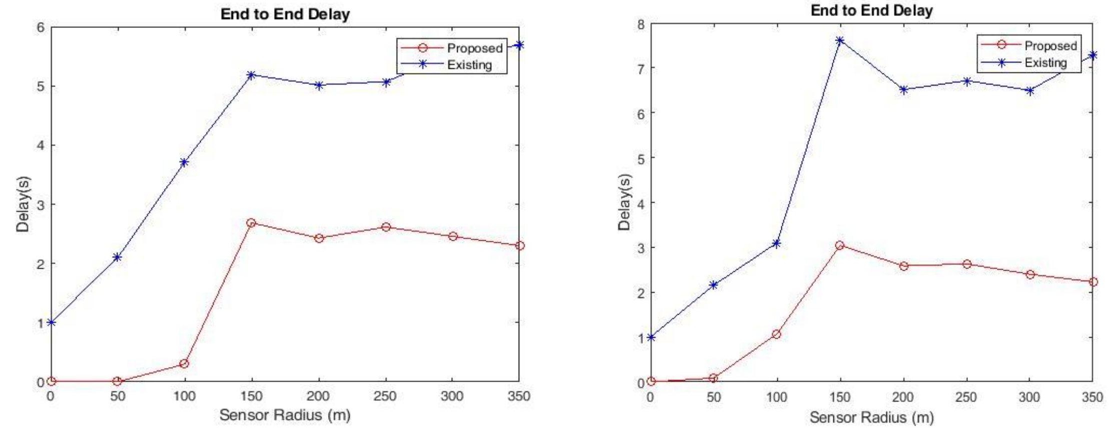
4.2 End-to-End Delay

From the study so far carried out, end to end delay remains the most affected parameter as demonstrated in Figure 4.2 the end delay maintained irregular pattern generally, it is also observable that due to lower vehicle density, and wider inter nodal distance, the range for lower vehicle densities tend to be higher hence, demonstrating chaotic patterns. The end to end delay will reduce with higher node density because of lower inter nodal distance and the fact that vehicles relay information among adjacent vehicles in VANETs hence, the stability observed pattern-wise at higher node density. In all cases, it is clear that the proposed approach performs better than the existing approach. However, as sensor radius increases, both the existing and the proposed techniques seems to exhibit higher delay, hence, there is a sensor radius beyond which our proposed system will not perform optimally, as it can be observed, beyond 180m, the end to end delay seems to depreciate in both the proposed and the existing schemes. At 40 nodes, there was a reduced delay with 32% improvement over the existing technique, at 50, 60 and 70 nodes, there were 26%, 35% and 37% improvement respectively when compared with the weight based cluster head selection algorithm.



(a) For 40 Nodes

(b) For 50 Nodes



(c) For 60 nodes

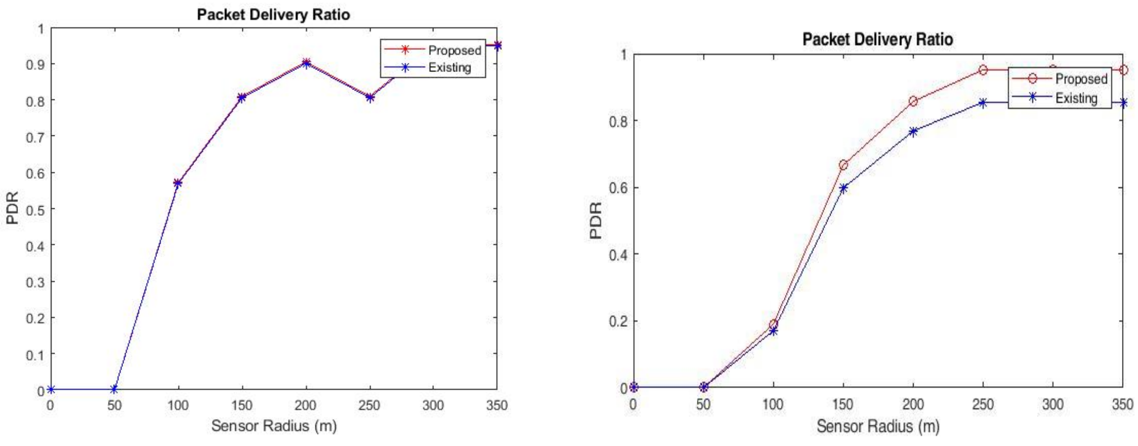
(d) For 70 nodes

Figure 4.2a-d: End-to-End delay versus Sensor Radius for varying number of nodes

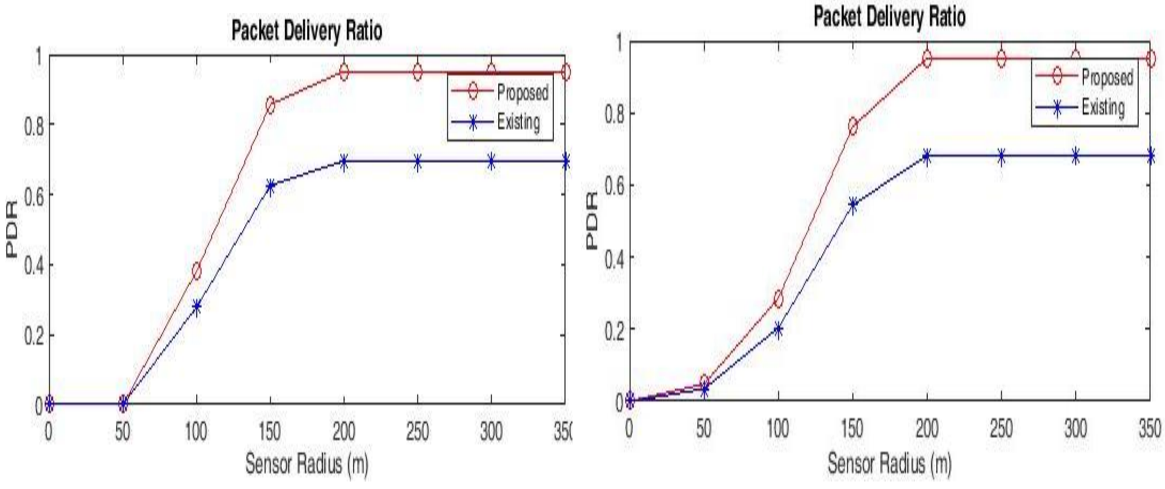
4.3 Packet Delivery Ratio

The packet delivery ratio is the ratio of packets received to packets sent. Packet delivery ratio is an important kpi when discussing clustering and cluster head selection in vehicular communication. In this work, the packet delivery ratio at 40 nodes was not any different hence there was a 0% difference in packet delivery ratio; this is because at this level, node density is low and hence, both algorithms behave alike.

At 50 nodes, the packet delivery ratio began to vary for the two algorithms, the proposed algorithm outperforms the existing by 11.4% and this is because as nodes increased, cluster size is readjusted hence, the reason for the difference experienced. At 60 nodes, the packet delivery improved by 26%, there was a flattening of the curve beyond 200m and this suggests that there is no improvement in packet delivery ratio for 60 nodes beyond 200m.At 70 nodes, there was a 28.5% improvement in packet delivery ratio with a flattening beyond 200m. This confirms that beyond 60 nodes and above 200m, there is no improvement in PDR.



(a) For 40 nodes (b) For 50 nodes



(c) For 60 Nodes (d) For 70 Nodes

Figure 4.3 a – d: Packet Delivery Ratio for varying Number of Nodes

4.4 Comparison among KPIs for Different Number of Nodes

In comparing the proposed algorithm in terms of KPIs for different number of nodes, it was observed from Table 4.1 which shows a comparison of the KPIs for the different number of nodes and figures 4.4, 4.5 and 4.6 which shows the curves for the comparisons that because of the stochastic nature of mobility patterns which results in frequent topology changes, the curves obtained do not have regular patterns. The cluster head formation time for 40, 50, 60 and 70 nodes. On the average, it was observed that the cluster head formation time for 40, 50, 60 and 70 nodes are 0.3381, 0.3394, 0.3404 and 0.3453 ms respectively. This suggests that cluster head formation time typically increases with increase in number of nodes. The margin of difference is not too obvious between successive number of nodes because the proposed algorithm reconfigures the cluster size as the number of nodes increase.

Table 4.1: Comparison among Performance Metrics for Different Node Density

SENSOR RADIUS (m)	END TO END DELAY (s)				CLUSTER HEAD FORMATION DELAY (ms)				PACKET DELIVERY RATIO			
	NO. OF NODES				NO. OF NODES				NO. OF NODES			
	40	50	60	70	40	50	60	70	40	50	60	70
0	0.0021	0.0001	0.0002	0.0000	0.3440	0.3450	0.3461	0.3506	0.0020	0.0100	0.0230	0 .0006
50	0.0010	0.0011	0.0003	0.1000	0.3377	0.3391	0.3409	0.3447	0.0031	0.0003	0.0342	0.0476
100	0.2714	0.4000	1.2000	0.6000	0.3380	0.3393	0.3398	0.3439	0.5714	0.1905	0.3810	0.2857
150	1.8286	2.9000	2.8000	2.6000	0.3372	0.3388	0.3401	0.3442	0.8095	0.6667	0.8571	0.7619
200	2.0000	2.5143	2.8500	3.1000	0.3373	0.3387	0.3386	0.3461	0.9048	0.8571	0.9524	0.9524
250	2.7429	2.3286	2.8600	2.5500	0.3373	0.3381	0.3392	0.3453	0.8095	0.9524	0.9524	0.9524
300	2.4714	2.6143	2.4000	2.2000	0.3368	0.3384	0.3396	0.3437	0.9524	0.9524	0.9524	0.9524
350	2.200	2.4143	2.2000	2.1000	0.3367	0.3376	0.3387	0.3438	0.9524	0.9524	0.9524	0.9524

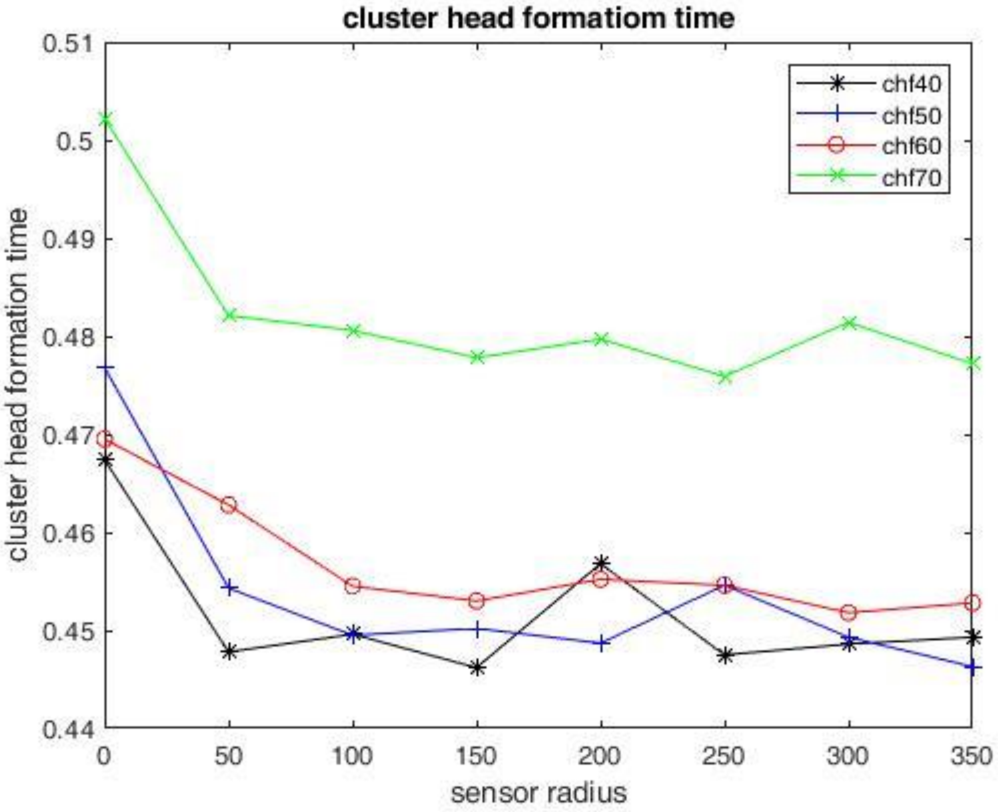


Figure 4.4: Comparison among Cluster Head Formation Time for Different Number of Nodes

The end to end delay which is the time taken for transmitted packets to be received is also affected by the stochastic nature of the traffic and frequent topology changes as well as channel impairments and terrain effects. When comparing the average end to end delay for 40, 50, 60 and 70 nodes, the end to end delay were 1.4393, 1.6464, 1.7887 and 1.6562 ms. it is observed that average end to end delay for 70 nodes is less than that of 60 nodes, this is because there is a reduction in cluster size as nodes increase hence, the end to end delay reduces this is one of the advantages of this algorithm. Figure 4.5 shows the graph of comparison for the end to end delay.

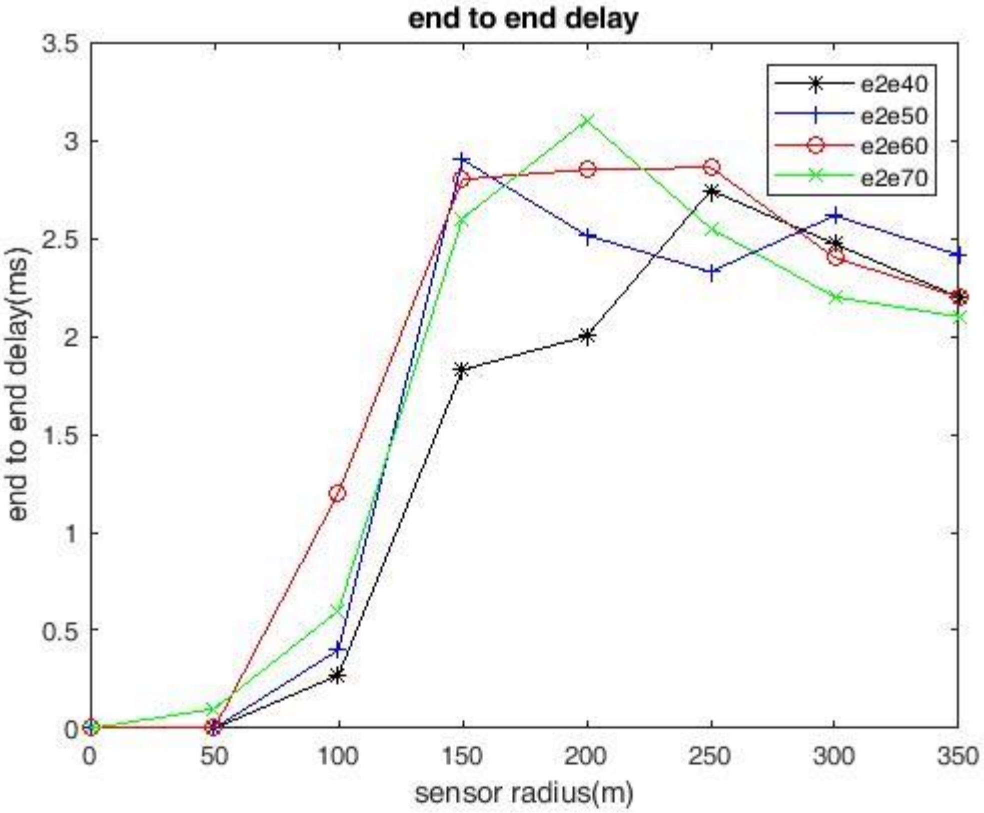


Figure 4.5: Comparison Among End-to-End Delay for Different Number of Nodes

The packet delivery ratio on the average for 40, 50, 60 and 70 nodes are 0.6250, 0.5714, 0.6310 and 0.6131. It is observed that the average PDR for 50 appears to be the least, this is because cluster size keeps reducing and hence, packet collision keeps reducing, this will cause PDR to improve at higher traffic density. It is thus observed that at 200m and beyond, PDR becomes constant for 60 and 70 nodes while at 250 m, it becomes constant for 50 nodes and for 40 nodes, it becomes constant at 300 m. PDR becomes constant when intermodal distance and terrain factors remain unchanged over time. Also, with each cluster size readjustment, there will be a change in packet delivery ratio as observed in Figure 4.6.

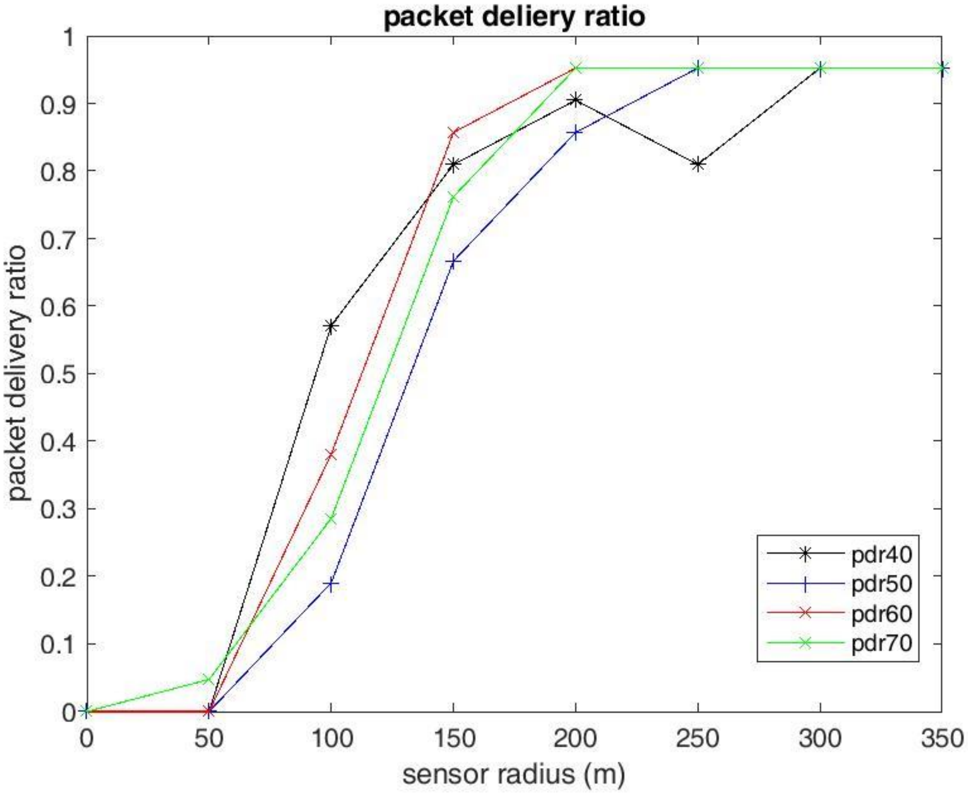


Figure 4.6 Comparison among Packet Delivery Ratio for Different Number of Nodes

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Vehicular adhoc networks remains one of the focal areas of future wireless technology. With continuous evolution in this field, realistic models are being developed to suit today's chaotic road traffic nature. Development of techniques that will cater for the frequent topology changes that are typical of VANETs has become very important. Clustering remains one of the techniques that can help in keeping quality of service at optimum level with rising road traffic as this work has demonstrated. It is ideal that cluster sizes reduce as the number of nodes in traffic increases, this will improve end to end delay and packet delivery ratio. Interference can also be reduced with hard bound clustering.

Cluster head selection remains a pivotal aspect in clustering in VANETs. Several works have investigated cluster head selection in VANETs, the distinguishing attribute in this work is that cluster head selection was done with consideration for transmission power and hence, KPIs were evaluated and plotted against the sensor radius. The result presented in Figures 4.1 and 4.2 demonstrates improvements in several respect, however, at higher sensor radius, parameters such as end to end delay begin to deteriorate. This suggests that keeping sensor radius small can greatly reduce the latency that will be experienced by mobile nodes in future vehicular adhoc networks.

5.2 Recommendations

Further studies can be carried out to establish the boundaries for sensor radius that will keep KPIs at optimal level. Other clustering techniques can also be employed apart from the one used in this work to develop very superior clustering techniques for VANET so that an industry standard can be established in the shortest possible time.

5.3 Contribution to Knowledge

The major contribution to the body of knowledge in this work is the introduction of transmission power as a basis for cluster head selection in vehicular adhoc networks. Furthermore, an evaluation of KPIs for different traffic density and sensor radius was carried out. This gives rise to the development of an enhanced weight-based cluster head selection algorithm for VANETs.

REFERENCES

Ahmed B., Malik A. W., Hafeez T., & Ahmed, N. (2019). Services and simulation frameworks for vehicular cloud computing: a contemporary survey. *EURASIP Journal on Wireless Communications, 4*, 1 - 21. doi:doi.org/10.1186/s13638-018-1315-y

Ahsan W, Muazzam M, Khan F, Aadil F, Ashraf S, Nam Y & Rho S. (2020). Optimized Node Clustering in VANETs by Using Meta-Heuristic Algorithms. *Electronics, 9*(3) 394. doi:10.3390

Aljeri N., & Boukerche A. (2017). *Performance evaluation of movement prediction techniques for vehicular networks.* Paper presented at the 2017 IEEE International Conference on Communications (ICC). 21 – 25 May, 2017. Paris, France.

Apratim C., Tomasz M., Chetan B. M., Hong L., & Justin D. (2016). *An Integrated Simulation Environment for Testing V2X Protocols and Applications*. Paper presented at the International Conference on Computational Science (ICSS) 2016, San Digo, California, USA. https://www.sciencedirect.com/science/article/pii/S1877050916310146

Bevish J. K. (2015). Rough Set Based Fuzzy Scheme for Clustering and Cluster Head Selection in VANET. *ELEKTRONIKA IR ELEKTROTECHNIKA, 21*(1). doi:http://dx.doi.org/10.5755/j01.eee.21.1.7729

Bijalwan A., Purohit K.C., Malik P., Mittal M. A. Self-Adaptable Angular Based K-Medoid Clustering Scheme (SAACS) for Dynamic VANETs. Electronics 2022, 11, 3071. https://doi.org/10.3390/ electronics11193071

Bryan C., Mani A., Chen-Nee C., H. Michael Z., & Dipak G. (2020). Enabling performance and security simulation studies of intelligent traffic signal light control with VENTOS-HIL. *Vehicular Communications, 24*. doi:https://doi.org/10.1016/j.vehcom.2020.100230

Busari S. A., Huq K.M.S., Mumtaz S, Rodriguez J. Fang Yi., Sicker D. C., Al-Rubaye S. Tsourdos A. (2019). Generalized Hybrid Beamforming for Vehicular Connectivity Using THz Massive MIMO. *IEEE Transactions on Vehicular Technology, 68*, 8372 - 8383.

Duan X., Wang Y., Liu Y., & Zheng K. (2016). SDN Enabled Dual Cluster Head Selection and Adaptive Clustering in 5G-VANET. 2016 *IEEE* 84th Vehicular Technology Conference (VTC-Fall), 18 – 21 September, 2016. Montreal, Canada, pp 1-5, doi:10.1109/VTCFall.2016.7881214

Emara M., Filippou M. C., & Sabella D. (2018). *MEC-assisted end-to-end latency evaluations for C-V2X communications*. Paper presented at the 2018 European Conference on Networks and Communications (EuCNC), Ljubljana, Slovenia.

Francisco J. M., Chai K., Toh Juan-Carlos C., Carlos T. C., & Pietro M. (2011). A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wireless communications and mobile computing, 11*, 813 - 828. doi:10.1002/wcm.859

Fuqiang L., & Lianhai S. (2010). Heterogeneous vehicular communication architecture and key technologies. *ZTE Communications, 8*(4), 39-44.

Gyawali S., Qian Y., & Hu R. Q. (2020). Machine Learning and Reputation based Misbehavior Detection in Vehicular Communication Networks. *IEEE Transactions on Vehicular Technology (Early Access),* 1 - 1. doi: 10.1109/TVT.2020.2996620

Hafeez K. A., Zhao L., Liao Z. and Ma B. N. -W., "A fuzzy-logic-based cluster head selection algorithm in VANETs," *2012 IEEE International Conference on Communications (ICC)*, Ottawa, ON, Canada, 2012, pp. 203-207, doi: 10.1109/ICC.2012.6363839

Husnain G & Shahzad. A. (2021). An intelligent cluster optimization algorithm based on whale optimization algorithm for VANETs (WOACNET). *PLoS ONE, 16*(4). doi:. https://doi.org/10.1371/journal.pone.0250271

Huang C.-M., Lin S.-Y., & Wu Z.-Y. (2020). The k-hop-limited V2V2I VANET data offloading using the Mobile Edge Computing (MEC) mechanism. *Vehicular Communications, 26*, 1 - 12. doi:10.1016/j.vehcom.2020.100268

Khayat G, Mavromoustakis C. X., Mastorakis G, Batalla J. M, Maalouf H, Pallis E. (2020). VANET Clustering Based on Weighted Trusted Cluster Head Selection. 16th International Wireless Communication and Mobile Computing(IWCMC), 15th – 19th June, 2020, virtual conference, pp 623-628, doi:10.1109/IWCMC48107.2020.9148339.

Karthikeyan H. & Usha G. (2021). Adaptive Clustering Algorithm for Stable Communication in Vanet. *Turkish Journal of Computer and Mathematics Education,* vol. *12*, number 9. pp 1778- 1785.

Kit G. L., Chun H. L., Renee K. Y., Kiam B. Y., & Kenneth T. K., (2016). *Simulators for vehicular Adhoc network (VANET) development*. Paper presented at the 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, South Korea. https://ieeexplore.ieee.org/document/7804797

Manoj S. & Charanjeet S. (2019). Comparative Study of Cluster Head Selection Techniques in VANET under traffic conditions. *Journal of Emerging Technologies and Innovative Research (JETIR), Vol. 6 issue 2.* 66 - 78. Available:http://www.jetir.org/papers/JETIREL06015.pdf

Mayank S, Pradeep K & Ranjeet S. T. (2023). *Weight-based Clustering Algorithm for Military Communication in VANET.* South African Institute of Electrical Engineers. Vol 114(1) March 2023.

Midya S., Roy A., Majumder K., & Phadikar S. (2018). *A Cluster-Based Handoff for Vehicular Ad-hoc Network*. Paper presented at the International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, India.

Nimbalkar N. B & Dass S. (2014). A Survey on Cluster Head Selection Techniques. *Multidisciplinary Journal of Research in Engineering and Technology, 1*(1), 5.

Nkoko S. S, & Kogeda, O. P. (2013) *A Cross-Layer Based Enhanced Handover Scheme Design in Vehicular Ad Hoc Networks.* Paper presented at the World Congress on Engineering and Computer Science San Francisco, USA.. Vol. 2

Oladosu G., Tu C., & Owolawi P. A. (2019). *Optimized Handover Algorithm for Vehicular Ad hoc Network.* Paper presented at the Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City, Shanghai China. Pp 317-321 https:/doi.org/10.1145/3377170.3377277.

Raza S., Wang S., Ahmed M., & Anwar M. R. (2019). A survey on vehicular edge computing: architecture, applications, technical issues, and future directions. *Journal of Wireless Communications and Mobile Computing, 2019*, 1 - 20. doi:doi.org/10.1155/2019/3159762

Ren M. (2019). *Cluster-based Data Dissemination in Vehicular Ad-hoc Networks (VANETs).* (Masters), Université de Technologie de Troyes, https://tel.archives-ouverts.fr/tel-02974809.

Saleem M. A, Shijie Z, Sanwar M. U, Ahmed T, Maqpool A, Shivachi C. S & Tariq M (2021) Deep Learning-Based Dynamic Stable Cluster Head Selection in VANET. Journal of Advanced Transportation, vol. 2021, Article ID 9936299, pp 1-21, https://doi.org/10.1155/2021/9936299 , Hindawi.

Shahid M. A (2019). Fixed Cluster Based Cluster Head Selection Algorithm in Vehicular Adhoc Network. Electronic Theses and Dissertations. 7737. https://scholar.uwindsor.ca/etd/7737.

Storck C. R., & Duarte-Figueiredo F. (2019). A 5G V2X ecosystem providing internet of vehicles. *Sensors, 19*(3), 550. https://doi.org/10.3390/s19030550.

Talib M. S. Hassan A. Thamer Alameri, Abas A. Z , Mohammed A. A, Ibrahim A. J & Abdullah N.I (2017). A Center-based Stable Evolving Clustering Algorithm with Grid Partitioning and Extended Mobility Features for VANETs. *IEEE Access,* vol. 8 pp 169908-169921. doi:10.1109/ACCESS.2020.3020510

Wang S., Chou W., Wong K.-S., Zhou A., & Leung V. C. (2018). Service Migration in Mobile Edge Computing. *Journal of Wireless Communications and Mobile Computing*, *2018*, 1 - 2. doi:doi.org/10.1155/2018/3823721

Wang Y., Lang P., Tian D., Zhou J., Duan X., Cao Y., & Zhao D. (2020). A game-based computation offloading method in vehicular multi-access edge computing networks. *IEEE Internet of Things Journal*, *7*(6), 4987 - 4996. doi:10.1109/JIOT.2020.2972061

Zhou Z., Liu P., Chang Z., Xu C., & Zhang Y. (2018) *Energy-efficient workload offloading and power control in vehicular edge computing.* Paper presented at the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain. (15-18 April). Pp 191-196, doi:10.1109/WCNCW.2018.8368975.

APPENDIX A (SIMULATION PARAMETERS)

Parameter	**Value**
Number of lanes	2
Length of road	4km
Packet size	500
Number of vehicles	40, 50, 60, 70
Sensor radius	0, 50, 100, 150, 200, 250, 300, 350
Cluster size	Variable
Number of clusters	Variable

APPENDIX B (ALGORITHM)

Inputs: number of neighboring nodes, speed, road id, discover message, node id, TX power.

Outputs: weights value

1. Tx_x_i
2. If $x_i \longleftarrow C_{jn}$
3. //there is an existing cluster within the vicinity of x_i
4. If beacon_messages> 1 // determine the best cluster from the list of clusters and join
5. If (pos_CH_i>pos x_i) and (rel_vel_CH_i<rel_vel_$CH_j$and lane id_x_i= lane id CH_i or CH_i) // compare relative velocity and position

```
6.  {
    Join $CL_i$ and connect to $CH_i$
    Status_$x_i$ = $CM_i$// vehicle joins cluster
    }
    Else,
    {
    join $CL_j$ and connect with $CH_j$
    }
7.  If CH leaves cluster
8.  Go to 26
9.  End if
10. End if
11.         End if
12. $CH_i$ or $CH_j$ Arrange cluster in array[] //in descending order of weights
13. Broadcast CM position in cluster to all cluster members
14. Cluster formation
15. If $x_i \in x_l$ // $x_i$ belongs to set $x_l$ which is a set of all vehicles within TX range
16. $X_i$ suitability // compute the suitability of $x_i$ to initiate cluster formation
17. Broadcast_$x_i$_ hello message
18. Tx_$\longleftarrow$$x_i$$x_i$_Tx() // compute waiting time for CH to respond
19. While Tx_$x_i$> 0,
20. Initiate cluster formation and assume CH and admit CMs
21. While weight_value_$x_i$is max($CL_i$)
22.  {
```

```
23. $X_i$ is cluster head,
            }
24. If $X_i$ leaves cluster. Then
25. {
26. $X_{i+n}$ is cluster head.
        }
27. Reconfigure cluster in descending order
28. If $CH_i$ leaves $CL_i$,
29. $X_{i+1}$ assumes cluster head.
```

APPENDIX C (some VANET simulation codes in MATLAB)

```
clc;
clear all;
close all;
warning offall;

restoredefaultpath;
addpath(genpath(pwd));
global N X Y XbYb E zXzY max1
rand('seed',1)
N=70;% Total No. of Nodes
RSU=[];%[250];
```

```
min1=0;

max11=400;

max1=200;


X = min1+(max11-min1)*rand(1,N);

Y = min1+(max1-min1)*rand(1,N);

R=100; %sensor field Radius

%co-ordinates of base station

Xb = max11+2+rand(1,1);

Yb =(max1+2)+rand(1,1);

%% Normalized vehicle velocity

maxv=50;

minv=10;

vel=rand(1,N);


%% Node Buffersize

E=rand(1,N);%1.*ones(1,N); % intialize node Buffersize

minTh=E/2;   %% minTh for buffer

Ptx = rand(1,N)


%%

figure,

for i2 = 1:N

        plot(X(i2),Y(i2),'o','LineWidth',1,...
```

```
'MarkerEdgeColor','k',...

'MarkerFaceColor','b',...

'MarkerSize',8');

xlabel('X in m')

ylabel('Y in m')

        text(X(i2), Y(i2), num2str(i2),'FontSize',10);

        hold on;

end

hold on

plot(Xb,Yb,'s','LineWidth',1,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','y',...

'MarkerSize',12');

xlabel('X in m')

ylabel('Y in m')

text(Xb, Yb, 'Sink','FontSize',10);

        hold on;

%% Equal-zone division

zX=0:100:max11;

zY=0:100:max1;

id=zeros(1,N);

%ipd=1;

ik1=1;

forik=1:numel(zX)-1
```

```
forij=1:numel)-1
   rectangle('Position',[zX(ik) zY(ij) 100 100])
   points =[zX(ik) zY(ij); zX(ik)+100 zY(ij);  zX(ik)+100 zY(ij)+100 ;zX(ik) zY(ij)+100
;zX(ik) zY(ij)]

   line(1:400,100.*ones(1,400),'Color','r','LineWidth',4)

if(~isempty(RSU))

      plot(RSU(1),100,'^','LineWidth',1,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','g',...
'MarkerSize',11');
xlabel('X in m')
ylabel('Y in m')
%text(Xb, Yb, 'Base','FontSize',10);
        hold on;
end

%figure()
%plot(points(:,1),points(:,2),'-*r')
   [in,on]=inpolygon(X,Y,points(:,1),points(:,2))

ind=find(in==1);
```

```
if(~isempty(RSU))

if(RSU(1)>zX(ik) && RSU(1)<zX(ik+1))

ind=[];

end

else

    %% Max Buffersize CHM

Ec=E(ind);

   SN1=ind;

   SN(ik1).id=ind;

   L=0.*ind;%label



   [val1,ind2]=sort(Ec,'descend');

MonitorID(ik1)=ind(ind2(1));



   ipd0=find(ind==ind(ind2(1)));

   L(ipd0)=1; %CHM

   plot(X(MonitorID(ik1)),Y(MonitorID(ik1)),'o','LineWidth',1,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','g',...

'MarkerSize',10');

xlabel('X in m')

ylabel('Y in m')

%text(Xb, Yb, 'Base','FontSize',10);
```

```
        hold on;

    %% Pch%
chmtosinkdist=abs(X(MonitorID(ik1))-Xb);
    pc=distpch(chmtosinkdist);
    s1=ceil((pc/100)*numel(ind))
    SN(ik1).s1=s1;

    %% Adaptive Clustering
    pos1=[X(ind) Y(ind)];
    vel1=vel(ind);
Bs=E(ind);
posd=pdist2([X(ind); Y(ind)]',[zX(ik)+50 zY(ij)+50]);

    w1=0.3;
    w2=0.2;
    w3=0.2;
    w4=0.3
    W=w1.*posd+w2.*vel1'+w3.*Bs'+w4*Ptx;
rng('default')  % For reproducibility
    [idx3,C,sumdist3] = kmeans([X(ind)+W';
Y(ind)+W']',s1,'Distance','cityblock','Display','final');

     w1=0.6;
```

```
   w2=0.2;

   w3=0.2;

   w=w1.*posd+w2.*vel1'+w3.*Bs';

rng('default')  % For reproducibility

   [idx3,C,sumdist3]                              =                              kmeans([X(ind)+w';

Y(ind)+w']',s1,'Distance','cityblock','Display','final');

idx=unique(idx3);


forip=1:numel(idx)

     ind3= find(idx3==idx(ip));

% select CH

     [val,ind2]= min(abs(W(ind3)-C(ip)))


     L((ind3(ind2(1))))=ip+1;


% assign cluster member

     ind3(ind2(1))=[];

     L((ind3))=ip+1+0.1;


end

%L(ipd0)=1; %CHM

%plot


%
```

```
    L
disp('hello');
%     if(numel(ind)~= numel(L))
%        pause,
%        break;
%     end
    SN(ik1).L=L;
    SN(ik1).Z=(ik1).*ones(1,numel(L));

end
    L=[];
    ik1=ik1+1;
%id(ind)=ipd;
    hold on
end
%ipd=ipd+1;0
end
L1=[SN.L];
id1=[SN.id];
ind4=find(L1==1)% High Buffersize node
hold on
plot(X(id1(ind4)),Y(id1(ind4)),'o','LineWidth',1,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','g',...
```

```
'MarkerSize',8');

xlabel('X in m')

ylabel('Y in m')

%text(Xb, Yb, 'Base','FontSize',10);


code=['r''m''y''k''c''w'];

for iv=2: max([SN.s1])


 ind5=find(round(L1)==iv)

 ind6=find(L1==iv)

hold on

plot(X(id1(ind5)),Y(id1(ind5)),'o','LineWidth',1,...

'MarkerEdgeColor','k',...

'MarkerFaceColor',code(iv-1),...

'MarkerSize',8');

xlabel('X in m')

ylabel('Y in m')

hold on

plot(X(id1(ind6)),Y(id1(ind6)),'s','LineWidth',1,...

'MarkerEdgeColor','k',...

'MarkerFaceColor',code(iv-1),...

'MarkerSize',11');

xlabel('X in m')

ylabel('Y in m')
```

```
end

axis([0 max11+5 -50 max1+50])


%% Neurofuzzy

outFIS=readfis('DataAnfis.fis');

%% Simulation Starts

round=500;

delv=[10 20 30 40 50 60 70 80];

R=[0 50 100 150 200 250 300 350];

Az=1;


foriu=1:8

ik=1;

while(ik<=20)

%     openfig('file.fig','new','visible')

for ik1=1:numel(zX)-1

forij=1:numel-1

    rectangle('Position',[zX(ik1) zY(ij) 100 100])

    points   =[zX(ik1)   zY(ij);   zX(ik1)+100   zY(ij);     zX(ik1)+100   zY(ij)+100   ;zX(ik1)

zY(ij)+100 ;zX(ik1) zY(ij)]

    line(1:400,100.*ones(1,400),'Color','r','LineWidth',4)

end
```

```
end


    z1=[SN.Z];

    L1=[SN.L];

    id1=[SN.id];


indw=find(mod(z1,2)==0)

    vel1=vel;

    vel1(id1(indw))=-vel(id1(indw));

if(Az==1)

     X=X+delv(iu).*vel1

else

     X=X+delv.*vel1

end


    X(X<-20)=400;

    X(X>420)=0;



    code=['r''m''y''k''c''w'];

for iv=2: max([SN.s1])


    ind5=find(floor(L1)==iv)

    ind6=find(L1==iv)
```

```
    hold on
    plot(X(id1(ind5)),Y(id1(ind5)),'o','LineWidth',1,...
'MarkerEdgeColor','k',...
'MarkerFaceColor',code(iv-1),...
'MarkerSize',8');
xlabel('X in m')
ylabel('Y in m')
    hold on
    plot(X(id1(ind6)),Y(id1(ind6)),'s','LineWidth',1,...
'MarkerEdgeColor','k',...
'MarkerFaceColor',code(iv-1),...
'MarkerSize',11');
xlabel('X in m')
ylabel('Y in m')
end
    A=randperm(N);
    path=[];
    L1=[SN.L];
    id1=[SN.id];
    z1=[SN.Z];

A(1)
indw=find(id1==A(1))
%find(L1(indw)==round(L1(indw))
```

```
    Zo=z1(indw)

Lz=SN(Zo).L;

indb=find(Lz==floor(L1(indw)))

    path1=SN(Zo).L(indb)

    %% new

if(ik>20)

    x=[X' Y' vel' E'];

    tic

    y=evalfis(x,outFIS);

    tp1=toc

    ind2w=find(uint8(y)==1);

    indCH1=[ind2w' N+1];


else

    tp1=1;

indCH=(find(L1==1 | L1==2 |  L1==3 | L1 ==4 | L1==5 | L1==6 | L1==7));

indCH=id1(indCH);


    indCH1=[indCH N+1]

end

    X1=[X Xb];

    Y1=[Y Yb];

    hold on

    plot(X1(indCH1),Y1(indCH1),'s','LineWidth',1,...
```

```
'MarkerEdgeColor','k',...
'MarkerFaceColor','w',...
'MarkerSize',11');
xlabel('X in m')
ylabel('Y in m')
        hold on
foricc=1:numel(indCH1)
        text(X1(indCH1(icc)),Y1(indCH1(icc)),num2str(indCH1(icc)),'FontSize',10);
end
        hold on
       plot(X1(end),Y1(end),'s','LineWidth',1,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','y',...
'MarkerSize',11');
xlabel('X in m')
ylabel('Y in m')
  distCHtoCH=pdist2([X1(indCH1);Y1(indCH1)]',[X1(indCH1);Y1(indCH1)]')  %  CH  to
CH distance
  trust=distCHtoCH;
%    trust(distCHtoCH>(R+200))=inf;
if(Az==1)
  trust(distCHtoCH>(R(iu)))=inf;
else
  trust(distCHtoCH>(R(3)))=inf;
```

```
end
  w1=trust;% Closing Time and Traffic
  [r_path, r_cost] = Predictive(path1,size(trust,1), w1)
  [r_pathE, r_costE] = hopbyhop(path1,size(trust,1), w1)
  indCH1(r_path)

t_cost(ik)=r_cost;
t_path{ik}=r_path;
  path=indCH1(r_path);
for p =1:(length(path)-1)
  line([X1(path(p)) X1(path(p+1))], [Y1(path(p)) Y1(path(p+1))],
'Color','m','LineWidth',2.5, 'LineStyle','-')
  arrow([X1(path(p)) Y1(path(p)) ], [X1(path(p+1)) Y1(path(p+1)) ])
end
axis([0 max11+5 -50 max1+50])

% PDR EStimate
    PDR(ik)= 0;
if(~isempty(path))
if(path(end)>=30)
        PDR(ik)= 1;
end
end
```

```
% Avg end to end delay


    E2E(ik)=numel(r_path);%r_cost;

    E2Ex(ik)=numel(r_pathE);%r_costE;

% CH formation Delay

    CHF(ik)=tp1

CHFe(ik)=*tp1


ik=ik+1;

pause(0.02);

clf;

end

AvgPDR(iu)=sum(PDR)/ik

E2E(isinf(E2E))=0;

E2Ex(isinf(E2Ex))=0;

E2Edelay(iu)=mean(E2E)

E2Edelaye(iu)=mean(E2Ex)

CHdelay(iu)=mean(CHF)

CHdelaye(iu)=mean(CHFe)

end

%

%


%if (Az==1)
```

```
%% Performance Analysis

% Velcoicty Vs PDR

AvgPDRE=AvgPDR./(1+rand(1))

figure,

plot(delv,AvgPDR,'*-r')

hold on

plot(delv,AvgPDRE,'*-b')

legend('Proposed','Existing')

title('PDR')

xlabel('velocity')

ylabel('PDR')


% Velocity Vs Average End to End Delay

figure,

plot(delv,E2Edelay,'*-r',delv,E2Edelaye,'-*b')

title('End to End Delay')

legend('Proposed','Existing')

xlabel('Velocity')

ylabel('Delay(s)')


% Velocity Vs CH FormationDelay

figure,

plot(delv,CHdelay,'*-r',delv,CHdelaye,'*-b')

title('CHformation Delay')
```

```
legend('Proposed','Existing')

xlabel('Velocity')

ylabel('Delay(ms)')

%else

AvgPDRE=AvgPDR./(1+rand(1));

figure,

plot(R,AvgPDR,'*-r')

hold on

plot(R,AvgPDRE,'*-b')

legend('Proposed','Existing')

title('PDR')

ylabel('PDR')

xlabel('Sensor Radius')


% Velocity Vs Average End to End Delay

figure,

plot(R,E2Edelay,'*-r',R,E2Edelaye,'-*b')

title('End to End Delay')

legend('Proposed','Existing')

xlabel('Sensor Radius')

ylabel('Delay(s)')

% Velocity Vs CH FormationDelay

figure,

plot(R,CHdelay,'*-r',R,CHdelaye,'*-b')
```

```
title('CHformation Delay')

legend('Proposed','Existing')

xlabel('Sensor Radius')

ylabel('Delay(ms)')

% R Vs PDR

% R Vs Average End to End Delay

% R Vs CH FormationDelay

%end

%histogram(PDR)

%histogram(E2Edelaye)

%histogram(AvgPDRE)
```

CLUSTERING ALGORITHM

Inputs: number of neighbouring nodes, speed, road id, direction, discover message, node id, TX power.

Outputs: weights value

```
1.    Tx_$x_i$
2.    If $x_i \longleftarrow C_{jn}$ // there is an existing cluster within the vicinity of $x_i$
3.    If beacon_messages> 1 // determine the best cluster from the list of clusters and join
4.    If (pos_$CH_i$>pos $x_i$) and (rel_vel_$CH_i$<rel_vel_$CH_j$and lane id_$x_i$= lane id $CH_i$ or
      $CH_i$) // compare relative velocity and position.
5.    {
6.    Join $CL_i$ and connect to $CH_i$
```

```
7.    Status_$x_i$ = $CM_i$// vehicle joins cluster
8.    }
9.    Else,
10.   {
11.   join $CL_j$ and connect with $CH_j$
12.   }
13.   If CH leaves cluster
14.   Go to 26
15.   End if
16.   End if
17.         End if
18.   $CH_i$ or $CH_j$ Arrange cluster in array[] //in descending order of weights
19.   Broadcast CM position in cluster to all cluster members
20.   Cluster formation
21.   If $x_i \in x_l$ // $x_i$ belongs to set $x_l$ which is a set of all vehicles within TX range
22.   $X_i$ suitability // compute the suitability of $x_i$ to initiate cluster formation
23.   Broadcast_$x_i$_ hello message
24.   Tx_$x_i x_i$_ $\longleftarrow$ Tx() // compute waiting time for CH to respond
25.   While Tx_$x_i$> 0,
26.   Initiate cluster formation and assume CH and admit CMs
27.   While weight_value_$x_i$is max($CL_i$)
28.   {
29.    $X_i$ is cluster head,
30.                     }
```

```
If $X_i$ leaves cluster. Then
{
$X_{i+n}$ is cluster head.
                }
Reconfigure cluster in descending order
If $CH_i$ leaves $CL_i$,
$X_{i+1}$ assumes cluster head.
```