Effective Communication of Messages and Path Establishment in VANETs using DABFS

Report submitted to SASTRA Deemed to be University
as the requirement of the course

CSE302 : COMPUTER NETWORKS

Submitted by
Sai Nikhitha D

(Reg No : 124156095, Computer Science and Engineering)

December 2022

SCHOOL OF COMPUTING

THANJAVUR-613401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF COMPUTING

THANJAVUR-613401

Bonafide Certificate

This is to certify that the report titled **"Effective Communication and Path Establishment in VANET using DABFS**" submitted as a requirement for the course**, CSE302: COMPUTER NETWORKS** for B.Tech. is a bonafide record of the work done by **Shri/Ms. SAI NIKHITHA D (Reg.No.124156095, COMPUTER SCIENCE AND ENGINEERING)** during the academic year 2022-23, in the School of Computing.

Project Based Work Viva voice held on December ,2022

Examiner 1 **Examiner 2**

ACKNOWLEDGEMENTS

First of all, I would like to thank God Almighty for his endless blessings.

I would like to express my sincere gratitude to **Dr S. Vaidhyasubramaniam, ViceChancellor** for his encouragement during the span of my academic life at SASTRA Deemed University.

I would forever remain grateful and I would like to thank **Dr A.Umamakeswari, Dean, School of Computing and R. Chandramouli, Registrar** for their overwhelming support provided during my course span in SASTRA Deemed University.

I am extremely grateful to **Dr. V. S. Shankar Sriram, Associate Dean, School of Computing** for his constant support, motivation and academic help extended for the past two years of my life in School of Computing.

I owe a debt of most profound gratitude to my mentor and lecturer **Shri Sujeet Jagtap S** for his valuable inputs, able guidance,encouragement,wholehearted cooperation throughout my project on the topic “Effective Communication of Messages and Path Establishment in VANETs using DABFS”.

I also thank all the Teaching and Non-teaching faculty, and all other people who have directly or indirectly helped me through their support, encouragement and all other assistance extended for completion of my project and for successful completion of all courses during my academic life at SASTRA Deemed University.

Finally, I thank my parents and all others who helped me acquire this interest in the project and aided me in completing it within the deadline without much struggle.

List of Figures

Figure No.	Title	Page No.
1.1	VANET Network Architecture	1
1.2,1.5	V2V and V2I Communication	2
1.3	Vehicle to Vehicle Communication	3
1.4	Vehicle to Infrastructure Communication	3
1.6,1.7	Cooperative Collision Avoidance	5
1.8	Position First Forwarding	5
1.9	Direction First Forwarding	6
1.3.1	Flowchart	11
1.3.2	Highway Scenario for Case 1	15
1.3.3	Highway Scenario for Case 2	16
1.3.4	Highway Scenario for Case 3	17
1.3.5	Highway Scenario for Case 4	18
1.3.6	Highway Scenario for Case 5	19
3.1,3.2	Initial Output with GUI	32
3.3	Line Connecting S and D	33
3.4	Intermediate Path	33
3.5	Final Path Establishment	34
4.1	Distance Graph	35
4.2	Energy Graph	35
4.3	Network Life Time Graph	36
4.4	Throughput Graph	37

ABBREVIATIONS

Direction Aware Best Forwarder Selection Protocol	DABFS
Cooperative Collision Avoidance	CCA
Vehicle-to-Vehicle	V2V
Vehicle-to-Infrastructure	V2I
Best Forwarder	BF
Combined Location and Velocity	CLV
Connectivity Aware Routing	CAR
Adaptive CAR	ACAR
Intersection-based CAR	iCAR
Acute Position-based Routing	ACR
Greedy Perimeter Stateless Routing	GPSR
Greedy Perimeter Coordinator Routing	GPCR
Maxduartion-Minangle-GPSR	MM-GPSR
Back-Bone-Assisted Hop Greedy Routing	BAHG
Junction-Based Routing	CJBR
Improved Directional Location Aided Routing	ID-LAR
Prediction-based Greedy Perimeter Stateless Routing	PGPSR
Predictive Directional Greedy Routing	PDGR
Improved Geographic Routing	IGR
Current Node Position	CNP
Global Positioning System	GPS

LIST OF NOTATIONS

Notation	Description
$\gg$	Packet forwarded from left to right node
Ack	Acknowledgment received in response to a $Hello$ packet
Bf	Best forwarder intermediary relay node
χ	Set of speeds for nodes
CNP	Current position of a node
D	Destination node
δ	Final distance between NGi and D
Δ	L1-norm distance between NGi and D
Ed	End-to-end delay for packets
Er	End-to-end delay ratio
$H(.)$	Hamming distance function
n	A member node
N	Set of all nodes
NG	Neighbouring table for a node
NID	Node identity
Pl	Packets lost
Pr	Packet loss rate
Pt	Total packets transmitted across the network
Rp	Total received packets
S	Source node
τ	Time stamp of the last received Ack
Tr	Network throughput

ABSTRACT

VANET-Vehicular ad hoc networks is an intelligent component of the transport system that helps the vehicles to communicate with each other and with roadside components too. It plays a very important role in transportation systems by ensuring the fact that less accidents take place over a period of time. This vehicle-to-vehicle communication is a wireless transmission of data/messages between vehicles which are nearby and offers services for safety improvements. Generally warning messages are transferred between the vehicles using a direction based greedy approach where the next hop is selected based on the current location of relay nodes, all the way towards the destination node. This method works well with unidirectional traffic, Since the vehicles are not permitted to have unidirectional traffic, this method proved to be ineffective because of more hops and end-to-end delay in case of bidirectional traffic.

So, This project deals with the positions of source and destination nodes in a Bidirectional environment and discusses its complexity further for efficient and robust routing of warning messages using a routing protocol namely, Direction aware Best Forwarder Selection(DABFS).

DABFS takes into account directions and relative positions of nodes along with the distance parameter to determine a node's movement direction using Hamming distance and forwards the messages through neighbours and discovers the best route to reach the destination. This proves to be the efficient protocol that increases the throughput and reduces packet loss along with end-to-end delay.

KEYWORDS: VANET, Greedy Protocol, DABFS, Hamming distance, Warning Messages

Table of Contents

Title	Page No.
Bonafide Certificate	i
Acknowledgments	ii
List of Figures	iii
Abbreviations	iv
Notations	v
Abstract	vi
1. Introduction,Network Architecture,Literature Survey Proposed Framework,Methodology,Merits and Demerits	1-20
2. Source Code	21-31
3. Experimental Analysis	32-37
4. Conclusion and Future Plans	38
5. References	39-40

CHAPTER 1

INTRODUCTION,NETWORK ARCHITECTURE, LITERATURE SURVEY, PROPOSED FRAMEWORK,METHODOLOGY,MERITS AND DEMERITS OF THE WORK

1.1 INTRODUCTION :

VANETs are the recent technological development in wireless networks where messages can be transferred between two vehicles.VANETs contains a set of moving objects travelling at high speed and communicating with each other.Vanets can be categorised into 3 categories like Wireless Wide area Networks(WWAN) ,Hybrid Wireless Architecture and Ad Hoc V2V Communication. VANETs acts as an important application for development of smart cities, avoiding road accidents, travel time prediction and the list goes on. It is becoming more competent every year. We can see every year approximately 2.5 million people are dying and 20-50 million are getting injured because of road accidents which is not tolerable.Our main objective in this paper lies in the application of road accidents prediction.

NETWORK ARCHITECTURE :

<table>
<tr><th>VANET</th><th>OSI/RM</th></tr>
<tr><td>Application Layer</td><td>Application Layer</td></tr>
<tr><td rowspan="4">Network Layer
Topology Based: AODV, DSR, DSDV, OLSR
Position Based: GPSR, GSR, GyTAR</td><td>Presentation Layer</td></tr>
<tr><td>Session Layer</td></tr>
<tr><td>Transport Layer</td></tr>
<tr><td>Network Layer</td></tr>
<tr><td>Data Link Layer
IEEE802.11DCF, RR-ALOHA</td><td>Data link Layer</td></tr>
<tr><td>Physical Layer
OFDM, UTRA-TDD</td><td>Physical Layer</td></tr>
</table>

Fig 1.1. VANET Network Architecture

In terms of Network Architecture , VANET can be divided into four layers as shown in Fig 1.1 with a Comparison to the OSI Model.

Physical Layer - VANET's characteristics strictly determine requirements of the physical layer.There are concerns about signal distortion caused by high speed , frequency deviation and high data rate need which can be dealt with the help of a physical layer.

Data Link Layer - The data link layer can be divided into logical link control (LLC) sub-layer and medium access control(MAC) sub-layer. The role of the MAC layer is to provide addressing and media access control methods, so that the nodes on the network can be implemented in the multi-point network communication without conflicts, usually to deal with problems such as hidden terminals and exposed terminals. Considering the wireless resources are extremely limited for VANET, it's quite critical to maximise the use of resources, so the pros and cons of the MAC layer protocol plays a decisive role in VANET's performances.
Network Layer - Network Layer lies on the top of Physical and Data Link Layer.It provides multi-hop wireless communications based on geographic addressing and routing with the help of transport layer and in the meantime executes specific functions like congestion control. Functionalities of *Session and presentation layer* were included into Network Layer.
Application Layer - Application Layer lies on the top of all the other layers and functions of Application layers are transferring and accessing files.The Application Layer allows users to send each other files through a network.

There are mainly three different modes of communication in VANETs.They are :

1) Vehicle to Vehicle Communication (V2V)
2) Vehicle to Infrastructure Communication (V2I)
3) Infrastructure to Infrastructure (I2I)

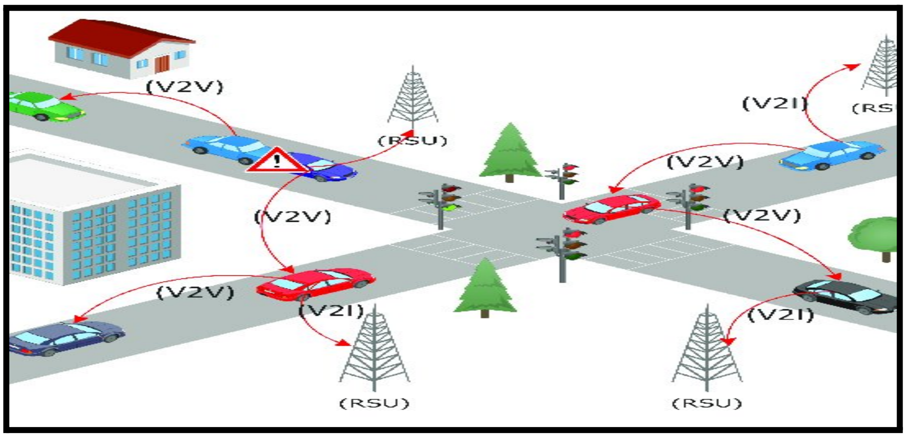

Fig 1.2. V2V and V2I Communication

A scenario of the V2V and V2I model is shown above. The concept behind V2V is that the communication is between two vehicles, hence the name Vehicle to Vehicle Communication.

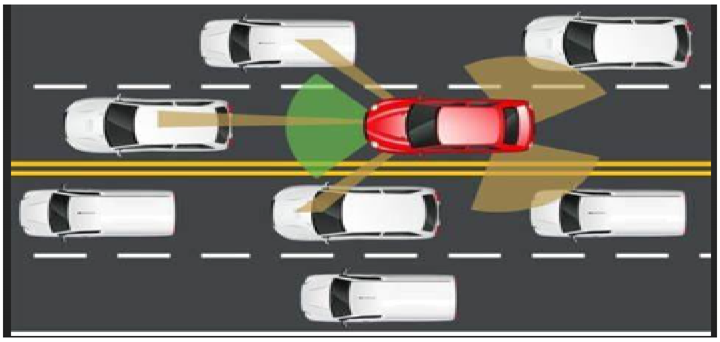

Fig 1.3. V2V Communication

V2V Communication is a wireless communication network where automobiles can be able to transmit information about their status. Apart from warning messages, it can also transmit certain other details about a car such as its location, the direction in which it is travelling, the stability of automobiles etc. an alert is sent to the driver, indicating a warning.

When it comes to V2I, the communication is between a car or any other vehicle and a traffic signal or any other stationary device near the vehicles or on RoadSide.V2I Communication plays a major role in self-driving cars or driverless cars, as the communication between vehicles and roadside infrastructure is considered very important. To establish this communication, Roadside Units(RSU) has to be established.This installation involves operational overhead because it is quite expensive to install and maintain them, leading to an increased capital. Use of sensors can capture the RSUs infrastructure and provide the drivers with necessary information such as traffic, accidents, construction sites, etc. This improves the mobility for pedestrians and vehicles.

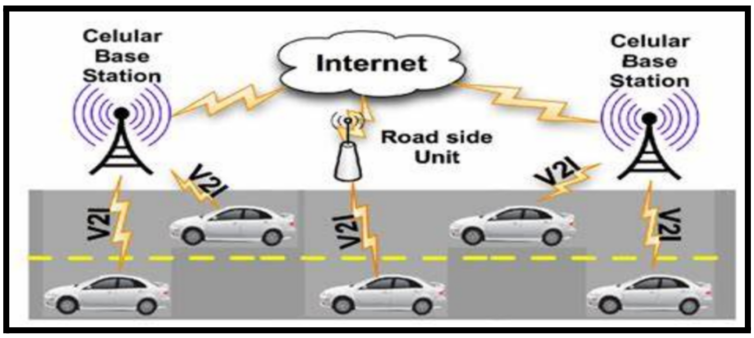

Fig 1.4. V2I Communication

V2V and V2I communications involve transmitting data to predict movement of vehicles in its surroundings, so that they can be prepared or plan their routes in advance. Both these schemes involve improving the communication network between automobiles and ensure safety by avoiding any accidents or collisions, hence saving lives.DABFS uses both V2V and V2I communication, which is the reason for its success.

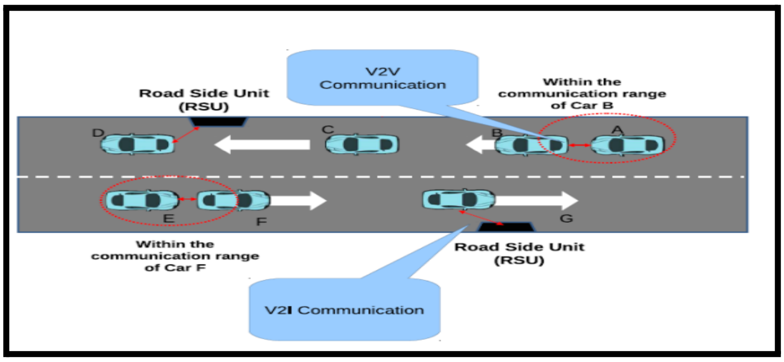

Fig 1.5.V2V and V2I Communication

Actually, to avoid such unpleasant situations like road accidents, the Cooperative Collision Avoidance (CCA) scheme was introduced. The CCA scheme computes probability among the nodes in regular intervals. It works in such a way that when the predicted collision probability exceeds threshold, a safe speed for the target node along with the probability of collision are computed and encapsulated in a warning message and this message will be transmitted to node on the other side to intimate about possible collision.CCA scheme works with both V2V and V2I Infrastructures.**The main disadvantage of this scheme is that this takes more time to compute the probability of collision**.The ability to send data properly in busy and fast traffic conditions is another challenge. So, we need a model that is reliable and it must be able to deliver the messages faster without loss to the destination node.

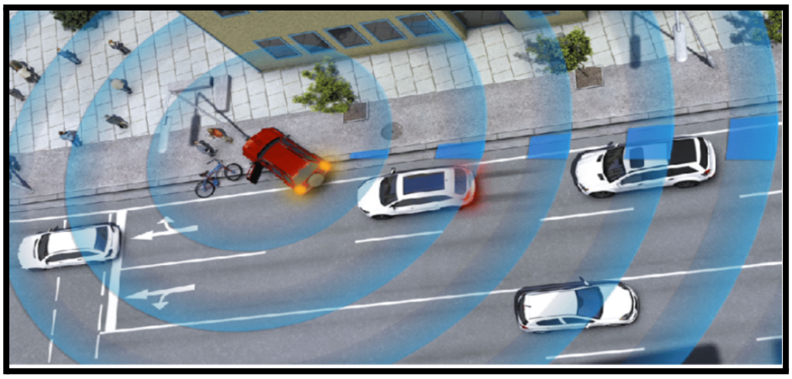

Fig 1.6.Cooperative Collision Avoidance(CCA)

As we can see in fig 1.1 when an accident takes place, the information about that will be transmitted to all the other vehicles in reachable range. This information including location and velocity readings of vehicles (Combined Location and Velocity), can allow the software implementation of collision detection or accident avoidance systems by giving an audible warning tone, when a vehicle is on a collision course. Fig 1.6 is a brief description of how the Cooperative Collision Avoidance scheme works.

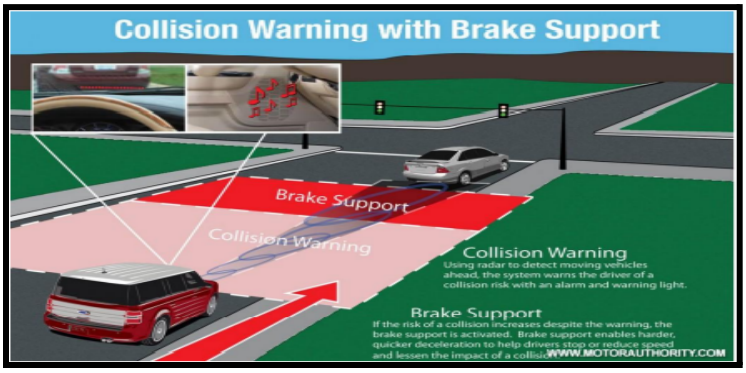

Fig 1.7.Cooperative Collision Avoidance(CCA)

Basically, when a source node is ready to transmit a message it will encounter two situations where it has only one route and multiple routes to reach destination.When more than one route is available, selection of best route with minimal latency and packet loss is important.This condition is considered to select the next hop of a path and that hop can be called as Best Forwarder.Selecting Best Forwarder will be a crucial decision to make while transmission of messages.Among all the proposed works direction-based Greedy routing protocols are considered to be efficient for Unidirectional scenarios for selecting the next hop towards the destination node. Direction Greedy Routing is based on greedy forwarding under consideration of different forwarding strategies.

1. Position First Forwarding :

This technique works based on the position of the node. Given the preferred forwarding direction of a packet this strategy tries forwarding the packet to nearest hop close to destination.It works well with unidirectional traffic and intends to reduce the hops and end-to-end delay.But in case of Bidirectional traffic this strategy won't work very well.Let's analyse a real life example where a car A is moving in a direction and car B is moving exactly opposite direction.Now car A sends a data packet and car B happens to be it's next hop closest to destination, packet will be sent to car B. Since the two cars will be at the same distance to each other car A will be nearest to car B and vice versa. This leads to a loop because of which data packet needs more hops to reach destination and end-to-end delay will be increased.

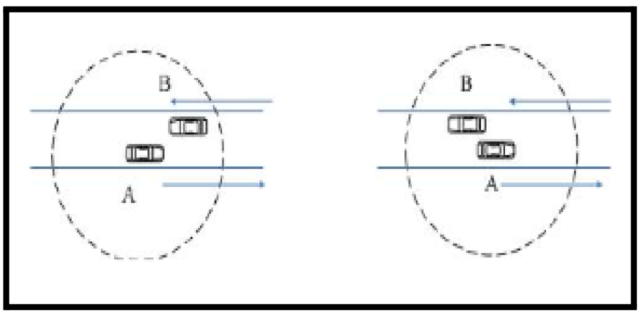

Fig 1.8. Position First Forwarding Scenario

2.Direction First Forwarding :

Direction-first Forwarding strategy will first select the nodes moving toward the destination.Among those nodes, the one closest to the destination will be chosen as the next hop. This scheme intends to reduce routing loops in the forwarding process. But another problem arises when we look into the example in Figure 1.9. Node A and B are moving towards the destination while C is moving in the opposite direction.A and B are very close. B is closer to the destination. C is the closest one to the destination among these three nodes. When A wants to forward a packet to the destination, it can choose B as the next hop if only the direction-first scheme is used. This may cause more hops and delays.Thus we can say Greedy based approaches are useful only with unidirectional traffic.

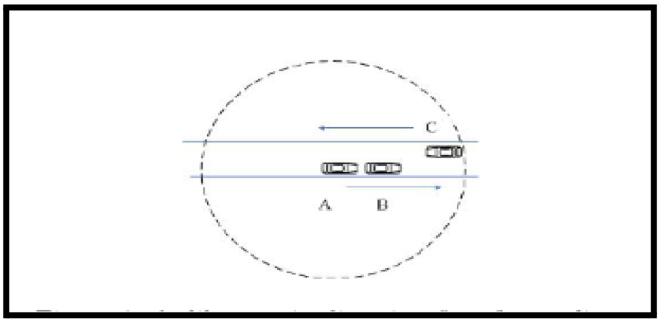

Fig 1.9. Direction First Forwarding Scenario

When speaking of bi-directional scenarios, it is observed that the performance of the same is reduced. In a bi-directional highway scenario, since the nodes travel in different directions, the distance between them keeps changing and this may lead to few problems. The topology created by the vehicle varies rapidly, which makes the path unpredictable. From this it is understood that the route which is chosen initially may or may not be the same till it reaches the destination node and it also has chances of changing the routes as we proceed further more steps. This has the possibility of causing network partitions. This leads to increased packet loss and latency in transmission of warning messages and reduced network throughput.

To overcome the above disadvantages, we use a protocol called DABFS.

- DABFS Routing Protocol makes sure to select the best route for transmission of warning messages with the use of Greedy approach in a V2V highway scenario.This ensures a fast and reliable delivery of warning messages thereby eliminating all topological changes and breakage of link routes.
- DABFS takes into account directions and relative positions of nodes, besides the distance parameter.

- DABFS determines the node movement direction using Hamming distance function and forwards warning messages through neighbours and discovers the best route for message transmission.
- The positioning of Source(S) and Destination(D) nodes plays an important role in transmitting. To have a fast transfer between the nodes, DABFS Protocol selects the most appropriate route from the sets of routes available, thereby making sure that the selected route is the most perfect for transmission.
- The most important use of DABFS is that it increases network throughput and reduces packet losses as compared to other protocols making it more efficient.
- DABFS acts as a robust protocol compared to all other protocols used by VANETS.

1.2 LITERATURE SURVEY:

There are several routing protocols proposed to create VANET systems. Vehicle to Vehicle data transmission is one of the key challenges in VANET architecture since it is required to design complex routing protocols.Some of the major categories of Routing Protocols in VANETS are mentioned below:

1.2.1 Location based, Geo-Cast and Geographic Routing Protocols :

In a Location based,Geo-cast and Geographic Routing Protocols, routing decisions are made based on the geographical position of the car.

Alzamzami, O. et. al (Alzamzami, O,2020) proposed a geographic routing protocol called Geo-LU. Generally, all the protocols select next hop within a single hop distance but this protocol improves the routing performance by selecting the next forwarder which has two hop neighbours information. It also uses a Link utility measure which considers the utility of neighbour link and packet loss rate.

Husain, A et. al(Husain, A,2020) proposed an optimised Geo-cast routing protocol. In most of geo-cast routing protocols the packets are transmitted within a geographic range. This proposed protocol uses some bio inspired soft computing techniques to send information from source to destination. It uses an approach called Particle swarm optimization.

Srivastava,A et. al(Srivastava, A,2020) proposed a Location based routing protocol in VANET. This protocol is unique and provides the best solution because it doesn't depend on predestination route entries. They analysed some challenges such as local optimal problem and optimum forward selection.

Many of the location-based,geo-cast and geographic protocols have less success rate due to dynamic VANET topology.

1.2.2 Cluster Based Routing:

Grouping in VANETs can be described as virtual partition of hierarchical nodes into different classes. A number of nodes identify themselves to be a part of the cluster. Each cluster has a cluster head and that is responsible for routing, handling cluster traffic and broadcasting packets to the cluster.

Pourroostaei Ardakani, S. et. al(Pourroostaei Ardakani, S,2018) proposed a cluster based routing protocol. They used a new addressing scheme using which every node will be given an address based on its mobility. This protocol decreases end to end delay and reduces traffic in comparison with AODV and DSDV protocols.

Hosmani, S, et. al (Hosmani, S,2020) proposed a protocol based on weight clustering technique. It selects a lightweight cluster head. Cluster information is gathered by considering mobility of vehicles and neighbours are selected using geographic distances. It is also called Weight based clustering as it assigns weight to every vehicle.

Cluster Based Routing protocols fail due to delays and network overhead faced when creating high-mobility VANET clusters.

1.2.3 Position Based forwarding:

Position based routing protocols use combined Location and Velocity tree to keep track of neighbour nodes for the selection of next hop.

Li, Zet. al(Li, Z,2019) proposed a position-based routing protocol called connectivity aware data dissemination protocol(CAR). In this, a new metric is added to every node to select the best forwarder. That metric is a combination of a throughput function and active connection time estimation. As of now, this protocol outperforms all other position based routing protocols.

Bujari et. al(Bujari,2019) proposed a probabilistic routing protocol,Acute Position-based Routing(APR). The approach they used was to maintain contention windows among vehicles which is used to select forwarder probabilistically. This reduces the number of hops to transmit a single message.But overhead in probability calculations increases computation cost.

Naderi, M. et. al (Naderi, M,2019) proposed Adaptive beacon broadcast opportunistic routing protocol(ACAR). By considering all the VANET characteristics like speed position and direction in this protocol an update strategy is introduced to send messages. It uses the lifetime of links between vehicles. It predicts the expiry time of links. This strategy reduces cost of routing and results in high network performance.

Position based Routing Protocols works well with Unidirectional Scenarios but fails with Bidirectional Scenarios as well as cannot give effective results in high speed mobility networks.

1.2.4 Greedy forwarding:

direction based greedy approach where the next hop is selected based on the current location of relay nodes, all the way towards the destination node

Nebbou, T. et. al (Nebbou, T.,2018) proposed a Greedy curve metric routing protocol. It selects the next hop based on curve metric distance instead of Euclidean distance. This protocol outperforms many position-based routing protocols.

Silva,A et. al (Silva, A,2019) proposed an extension to GPSR (Greedy perimeter stateless routing) called PA-GPSR (Path aware GPSR). It is a geographic position-based routing protocol. In this protocol packet routing loops are avoided which in turn reduces the chance of delivering the same packet to the same node again and again. This protocol also overcomes link breakage issues.

Yang, X et. al (Yang, X,2018) proposed an extension to GPSR which is MM-GPSR (Maxduration-Minangle GPSR). In this protocol, the neighbour node whose cumulative communication range is maximum is selected as the next hop node. The concept of minimum angle ensures that the selected next hop node is optimal. In MM-GPSR packet loss rate is decreased in comparison to GPSR.

Zahedi, K et. al (Zahedi, K,2019) proposed protocol concentrates on connectivity of junction-based routing(CJBR) in VANETs. It follows a metric junction selection mechanism. It enhances connectivity among nodes to achieve maximum performance.

This Protocol won't give efficient results with a high speed mobility network and this protocol remains costly due to its high control messages overhead.

1.2.5 Direction based Greedy Forwarding:

Direction-based routing protocols take into account the link between the nodes to identify the best route. High mobility of nodes leads to frequent link breakages.

Karimi, R. et. al (Karimi, R,2018) proposed Predictive Geographic Routing protocol (PGRP). In this protocol, every node gives weight to its neighbour nodes depending on the direction and angle of vehicle of the neighbouring nodes. This is used to predict the current position of every vehicle and the forwarding is done depending on the position of the vehicle. This protocol better packet delivery ratio

Wang, C., et. al (Wang, C., 2018) proposed a Prediction based greedy perimeter stateless routing(PGPSR) which is an extension of GPSR. In this protocol, the hello packet is structured in a different way. In this structure the information of new nodes such as the new node's speed and direction information are added. This information is used to predict the next node in new changed network topology and the information about outdated nodes is deleted periodically.

Direction based routing protocol gives better results with less number of hops in Unidirectional traffic but in case of bidirectional traffic it doesn't work effectively.

1.3 PROPOSED FRAMEWORK:

The brief details of the proposed Direction Aware Best Forwarder selection protocol (DABFS). for V2V communication in bi-directional scenario is shown through the following flowchart.

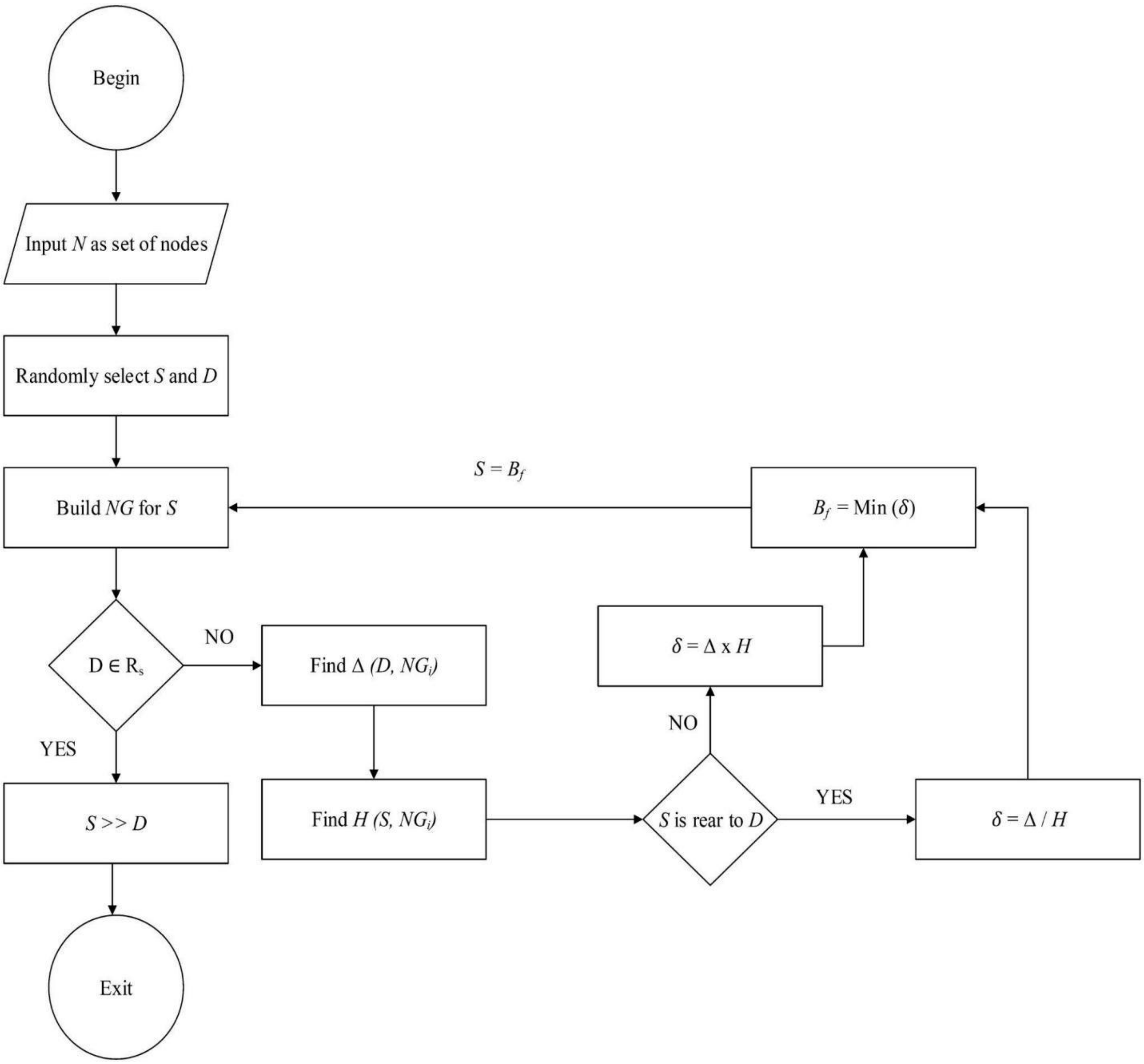

Fig 1.3.1. Flowchart

1.4 METHODOLOGY :

DABFS protocol uses the following parameters in addition to distance parameters.

- The movement direction of nodes is computed using the Hamming distance function.
- Hamming Distance Function : If the two nodes are moving in the same direction their hamming distance is considered as 0.If the nodes are moving in opposite direction their hamming distance function is taken as 1.
- Relative position of source and destination nodes.

This DABFS protocol is explained using 3 Algorithms

1) Message Transmission
2) Formulating Neighbouring Table
3) Best Forwarder Discovery

Algorithm 1 : Message Transmission

Input : S and D
Output : Success or Failure in transmission of message

Begin
    Build NG_s for S
    Loop
        If D belongs to NG_s **Then**
            S >> D
            Exit
        Else
            Identify Bf
            S >> Bf
            Set Bf as S
            Build NG for newly deputed S
        End If
    End Loop
End

Explanation :

The Algorithm takes input S and D where S represents Source node and D represents Destination node. The source node's neighbouring table is built.S is ready to transmit messages. For the availability of a Destination node there are two possibilities.

1. The Destination is at one hop distance - In this case the message is transmitted directly.
2.The Destination is not found in the neighbouring table which means the destination is at multi hop distance from source node - In this case, Source node takes the help of intermediate nodes by choosing a forwarder to reach destination.

The output of the algorithm is the success or failure in transmission of message.

Algorithm 2 : Formulating Neighbouring Table

Input : N
Output : NG
Begin
    Flush NG_s
    Set i to 0
    S checks nodes within range
        Repeat
            If found within range from Node ni
            S updates NG_s
                $NG_s(i,1)$ = N IDni
                $NG_s(i,2)$ = C N Pni
            Increment i
        Until all nodes are processed
    Return NG
End

Explanation :

In Algorithm2 N, the number of nodes in the simulation area is taken as input. This algorithm works for discovering neighbouring nodes and formulating a table of neighbour nodes of a particular node. If a node is found in the communication range of the source node then it adds that particular node to its neighbouring table. Details of node added to the neighbouring table include id of node,current position of node. In Algorithm N IDn represents id of the node and C N Pn represents the current position of node.The algorithm returns the computed neighbouring table.

Algorithm3 : Best Forwarder Discovery

Input : S, D, NG_s
Output : Bf
Begin
    For i=0 **to** NG_s-1
        Δi = L1-norm (NGsi, S)

```
Hi = H(NGsi,S)

    If H(S,D)==1 Then
        If S==Rear and D=front node Then
            δ i = Δi / Hi
        Else
            δ i = Δi * Hi
        End If
    Else
        If S and D are moving towards each other Then
            δ i = Δi / Hi
        Else
            δ i = Δi * Hi
        Endif
    Endif
End for
Bf=Min(δ)
Return Bf
End
```

Explanation :

We know that due to frequent topology changes, the route may change at every instant of time. So, finding the best forwarder at every instant is very important. The L1-norm of every node in the neighbouring table with source node is calculated. This protocol takes the relative position of a source and destination node as the main parameters. If the source and destination nodes are moving in the same direction i.e., if the Hamming distance of source and destination equals 1 then we check whether the source is at the rear position or front position. If source is at rear position then δi = Δi / Hi else δi = Δi * Hi. If the hamming distance of source and destination equals 0 i.eThe source and destination are moving in the opposite direction then we check whether the source and destination are moving towards each other. If source and destination are moving towards each other then δi equals Δi / Hi else δi equals Δi * Hi. In this way δ is computed for every node in neighbouring table. The neighbouring node which has small value of δ is selected as the best forwarder.

The inputs are S, D, and NG. First, we calculate the relative position of S and D along with their movement direction in addition to their distance parameter. This enables us to calculate the best route for transmitting the warning message to the destination node. The output of this algorithm gives us the minimum distance Bf and best route for reaching our destination node D. By this way, it becomes easy for us to find the best route for transmitting the warning message as soon as possible by considering various parameters.

To identify the importance of additional parameters in DABFS routing protocol, five cases are considered. These five cases include all possible scenarios on the bidirectional highway.

Case 1:

Source is at rear position and destination node is in front position. They are moving in the same direction.

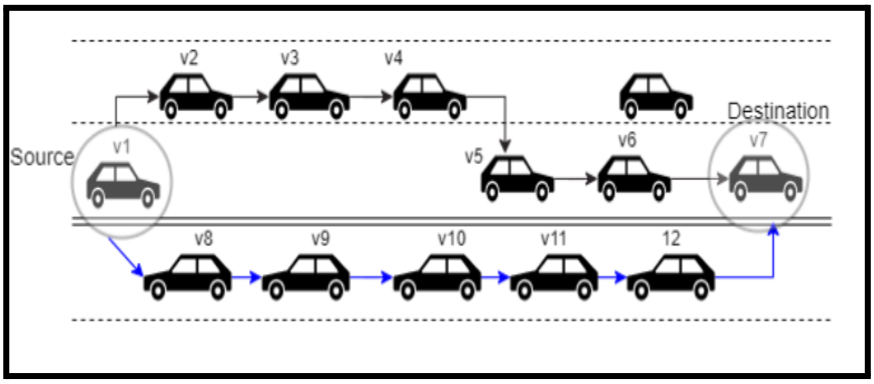

Fig. 1.3.2.Highway scenario for case 1

In this case, consider the scenario shown in figure 1.3.2.

- Source Node is v1
- The Destination Node is v7
- v1 is at the rear position and destination
- v7 is at the front position
- Both the nodes are moving in same direction

There are two routes possible from v1 to v7

1) v1→v2→v3→v4→v5→v6→v7

2) v1→v8→v9→v10→v11→v12→v7

All the direction-based approaches and greedy approaches select v8 as the forwarder which means route 2 is selected but this kind of selection takes only the direction into consideration. The performance of such kinds of protocols will not reach the expected results in VANET's, because when the message reaches node v12 the destination node v7 will not be available because it will be out of communication range of node v12. So, considering the relative positions of nodes and direction of nodes is very important. Node v12 has two possibilities in this case either the node v12 will not be able to forward which means node v12 will not find a node to forward or it finds a node to forward and starts to reconstruct the path. At the time message reaches node v12 the node v2 will be in the communication range of v12. Hence, to transmit a message from v12 to v7, v12 takes the help of v2, v3, v4, v5, v6 nodes. This may increase the additional number of nodes on the second route. But, DABFS selects the first route where intermediate nodes are moving in the same direction towards the destination node. This leads to a high chance of message delivery and a decreased number of nodes comparatively. This proves that the direction of intermediate nodes is crucial in the route discovery. Best forwarder (Bf) in DABFS, when source and destination are in the same direction and source is at rear position is computed by

δi = Δ / H(NGsi,S) - (1.1)

δi represents the final distance of NGsi from the destination node. Δ represents the L1 norm distance of the neighbour node and destination node. H is the function that represents Hamming distance. S represents the source node.

Case 2:

Source is at front position and destination node is in rear position. They are moving in the same direction.

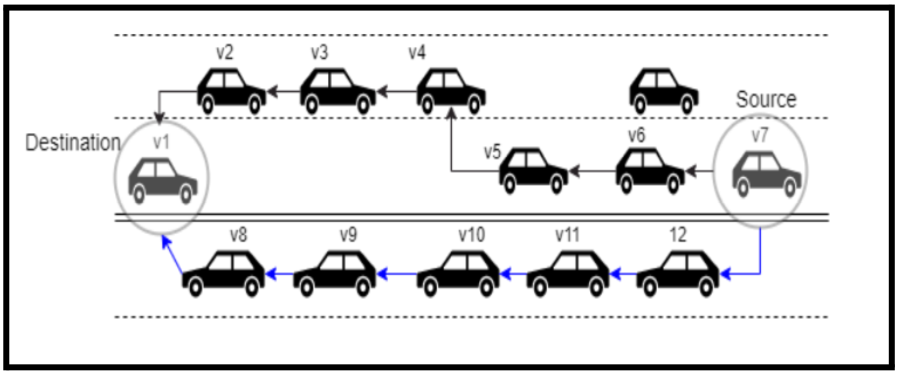

Fig. 1.3.3.Highway scenario for case 2

Consider the scenario in figure 1.3.3.

- The Source node is v7
- The Destination node is v1
- The v7 is at the front position and the v1 is at the rear position.
- Both the nodes are moving in the same direction.

The possible routes for this scenario are

1) v7→v12→v11→v10→v9→v8→v1
2) v7→v6→v5→v4→v3→v2→v1

Here DABFS selects v12 as the first forwarder which is in the opposite direction to the direction of the destination node. The selection of v12 as forwarder is efficient because when the message reaches from v12 to v11, the destination node v1 falls within the communication range of v11 which leads to direct forwarding of a message from v11 to v1. So, the route chosen by DABFS will be v7→v12→v11→v1. The destination is found within less time when compared to the predefined route. So, the route is not predefined in DABFS instead the next forwarder is selected at every instant. If H (S, D) =1 i.e., the Source and destination are moving in the same direction and the source is at the front position then the selection of the best forwarder (Bf) is computed by

$\delta i = \Delta * H(NGs, S)$ - (1.2)

δi represents the final distance of NGsi from destination node. Δ represents the L1 norm distance of the neighbour node and destination node. H is the Hamming distance function. S represents the source node.

Case 3:

Source is at rear position and destination node is in front position. They are moving in opposite directions and moving towards each other.

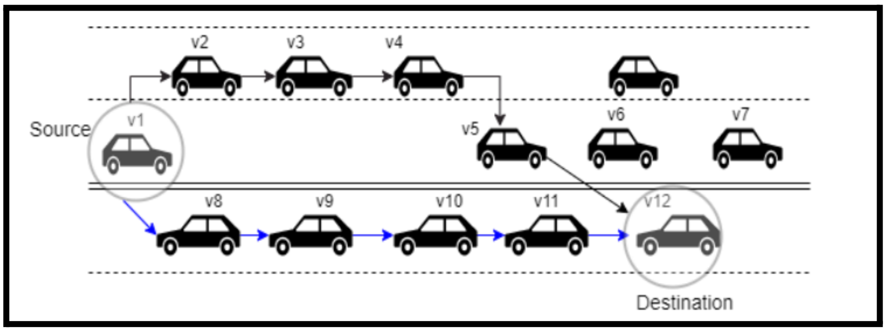

Fig.1.3.4. Highway scenario for case 3

Consider the scenario in figure 1.3.4.

- Source node is v1
- The Destination node is v12
- Source is at rear and the destination is at the front position
- v1 and v12 are moving towards each other
- Source and destination nodes are in opposite direction

By observing source node and destination node's position it is close to case1 but as node v12 lies in the opposite direction of the road this situation is completely different. The two possible routes initially are:

1) v1→v8→v9→v10→v11→v12

2) v1→v2→v3→v4→v5→v12

All the greedy protocols select v8 as the forwarder because the distance between node v1 and node v8 is less when compared to the distance between node v1 and node v2. But, the DABFS protocol selects v2 as the forwarder. It uses (1) to select a node among all possible neighbouring nodes. when the message reaches node v3, v12 will come into the communication range of v3. Therefore, without forwarding to node v4 and node v5 directly, node v3 forwards to node v12. The route will be v1→v2→v3→v12. This reduces the number of intermediate nodes. In this kind of situation the hamming distance function returns 0 indicating that both the nodes are in opposite directions. In this case, for the selection of the best forwarder (Bf) DABFS uses the equation in (1.1).

Case 4:

Source is at front position and destination node is at rear position. They are moving in opposite directions and moving away from each other.

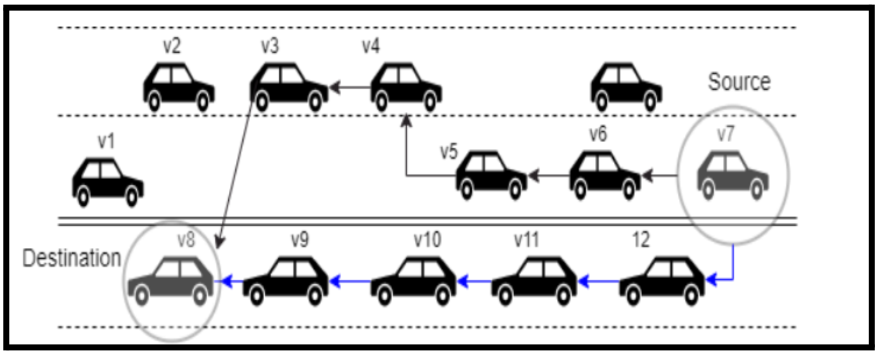

Fig.1.3.5. Highway scenario for case 4

Consider the scenario in figure 1.3.5.

- Source node is v7
- The Destination node is v8
- v7 is at front and v8 is at rear position
- Source and destination are in the opposite direction
- Source and destination nodes are moving away from each other

The two possible routes from v7 to v8 are

1) v7→v12→v11→v10→v9→v8

2) v7→v6→v5→v4→v3→v8

All the normal greedy approach - based protocols chooses v6 as the forwarder. But this is not efficient because as the message reaches node v3, the destination node v8 will be not there in the communication range of v3. Again, from node v3 reconstruction of the path should start. As part of the path reconstruction process, the message should be forwarded to v2, v1, v11, v10, v9 to reach v8. Because of this, there will be an increase in the number of intermediate nodes. Suppose if node v3 doesn't find any nodes to forward then it is impossible to send the message. To overcome this situation, the DABFS selects nodes v12 as the best forwarder. Hence, the message follows route 1 to reach the destination. This reduces the number of intermediate nodes. If source and destination are moving in opposite directions and they are moving away from each other, then for the selection of the best forwarder (Bf) the DABFS uses equation (1.2).

Case 5:

Source is at rear position and destination node is in front position. They are moving in opposite directions and moving towards each other.

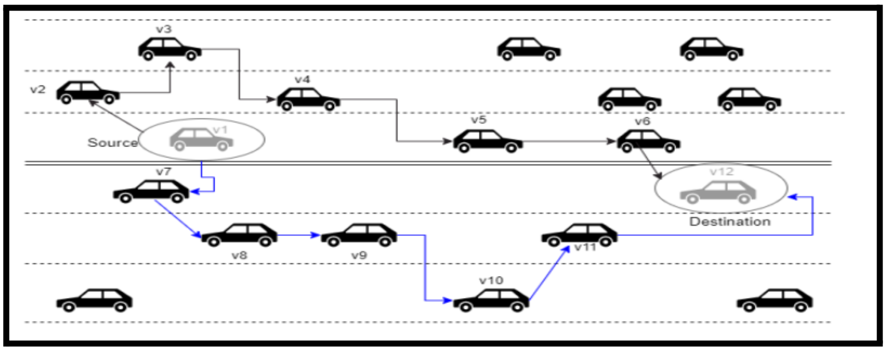

Fig.1.3.6. Highway scenario for case 5

Consider the scenario in figure 1.3.6.

- Source node is v1
- The destination node is v12
- v1 is at the rear position and v12 is at the front position.
- Both v1 and v12 are in the opposite direction.v1 and v2 are moving towards each other.

There is a chance of availability of intermediate nodes that are in the range of source nodes but they will be at a larger distance to reach the destination node when compared to the source node. This type of situation refers to recovery mode. This is very common in greedy approach-based protocols. In this case, node v1 wants to transmit data to node v12 there are two possible routes:

1) v1→v7→v8→v9→v10→v11→v12

2) v1→v2→v3→v4→v5→v6→v12

If route 1 is chosen when the message reaches v7 node v8 will be out of the communication range of v7. This leads to message drop. So, similar to case 3, DABFS protocol chooses node v2 as the forwarder. As node v2 is in the opposite direction, node v12 comes within the communication range of node v4 by the time node v4 receives the message. The route followed by DABFS is v1→ v2→v3→v4→v12. This decreases the number of intermediate nodes and improves the message reliability.

1.5 MERITS AND DEMERITS

DABFS protocol works well in bidirectional highway scenarios where most existing protocols failed to produce good results. DABFS produces better throughput and thus leads to efficient transmission of messages. Throughput improves further in dense networks. Even if the initial routes are broken, this DABFS transmits messages by transferring to other possible routes. As the route is not fixed in DABFS sometimes this will deliver messages in less time than expected.

DABFS is not suitable in urban environments where there are a lot of intersections and it works if vehicles have the capacity to track the positions of other vehicles using some technologies like GPS. The transmission is not guaranteed where there are signal and network issues.

CHAPTER 2
Source Code

Vanet.m

%This code is used to create gui using matlab

```
function varargout = vanet(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @vanet_OpeningFcn, ...
                   'gui_OutputFcn',  @vanet_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
function vanet_OpeningFcn(hObject, eventdata, handles, varargin)
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
function varargout = anish_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
% --- Executes on button press in Simulate.
function Simulate_Callback(hObject, eventdata, handles)
%get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a double
NumOfNodes=str2double(get(handles.edit1,'string')); %edit is the name of the edit box using
GUIDE the names of the editboxes are done consecutively (edit1,edit2,edit3...)
src_node=str2double(get(handles.edit2,'string'));
```

```
if isnan(src_node) %returns a logical array containing 1 ( true ) where the elements of A are NaN(Not a Number) , and 0 ( false ) where they are not else it will consider the entered integer value
  src_node=round(1+(NumOfNodes-1).*rand); %if source is not a number generate a random number as source
end
if src_node>NumOfNodes
  errordlg('Source ID is more than Number of nodes','Index Exceeds') %If source node index given is greater cmp to total raise an error using errordlg
  return
end
dst_node=str2double(get(handles.edit3,'string'));
if isnan(dst_node)
  dst_node=round(1+(NumOfNodes-1).*rand);
end
if dst_node>NumOfNodes
  errordlg('Destination ID is more than Number of nodes','Index Exceeds')
  return
end
%code for plotting data in graphs about distance b/w src and dst,energy req
%by a packet to be transmitted nd lifetime of packet along with its
%throughput
[distance,energy,nw_liftime,throughput]=UrbanCitySimu(NumOfNodes,src_node,dst_node);
function listbox3_Callback(hObject, eventdata, handles)
%A list box is created to list out all the possible paths for message trans.
function listbox3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
  set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Clear. Used to clear  data of gui
function Clear_Callback(hObject, eventdata, handles)
cla
function edit1_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
  set(hObject,'BackgroundColor','white');
end
function edit2_Callback(hObject, eventdata, handles)
```

```
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
  set(hObject,'BackgroundColor','white');
end
function edit3_Callback(hObject, eventdata, handles)
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
  set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in pushbutton3.
%This MATLAB function creates a push button in a new figure and returns the Button object.
function pushbutton3_Callback(hObject, eventdata, handles)
close all
```

Equal.m

%This code is used to check if the input source variable leads to destination or not

```
function log=equal(inpu,dst) %returns the result of comparision of dst and inpu to a variable
called log
% dst=35;
if ismember(dst,inpu) %input is a member of dst set log=1
  log=1;
else
  log=0;
end
end
```

Evaluation.m

%This code is used to evaluate packet loss,performance and throughput

```
function [E,pcktlossrate,total_dist,pcktloss,thrgput]=evaluation(nodtble,node_rsu)
% take out the distance of nodes in routing table from each other
distnc = zeros(1,1000000);
for ii=1:numel(nodtble)-1

distnc(ii)=sqrt((node_rsu(nodtble(ii+1),3)-node_rsu(nodtble(ii),3))^2+(node_rsu(nodtble(ii+1),4)
-node_rsu(nodtble(ii),4))^2);
end
```

```
% take out the distance of nodes in routing table from each other
for ii=1:numel(nodtble)-1 %numel-count the number of array elements
distnc(ii)=sqrt((node_rsu(nodtble(ii+1),3)-node_rsu(nodtble(ii),3))^2+(node_rsu(nodtble(ii+1),4)
-node_rsu(nodtble(ii),4))^2); %sqrt((x2-x1)^2+(y2-y1)^2)
end
total_dist=sum(distnc); % total distance from source to destination
time_consumed=total_dist/(3*10e9); %aeb=a*10^b
%% Performance Evolution
pkt size=64;% in bytes
data rate=[4,6,8,10,12,14]; % packets/sec
Etx=1;% in joules
Eini=Etx;
Elec=50e-9; %amount of Energy consumption per bit in the transmitter or receiver circuitry
Emp=0.0015e-12;%Amount of energy consumption for multipath fading-path changes
EDA=5e-9; %Data aggregation energy-summarise data from diff multiple sources
% parameters for energy calculation using ratio model of message transmission
alpha1=50e-9; %J/bit
alpha2=0.1e-9; %J/bit/m2
alpha=2;
Ebit=0.3e-3; % energy assigned to each bit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%radio Model for energy consumption is
% E=alpha1+alpha2*(dist)^alpha
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hop=numel(nodtble);
for ff=1:length(datarate)
E(ff)=(alpha1*datarate(ff)*pktsize*8)+(alpha2*datarate(ff)*pktsize*8)*(total_dist)^alpha;%
radio model energy consumption calculation at data rate
  Edata(ff)=Ebit*datarate(ff)*pktsize*8; %energy consumption of data wrt bits
  for ll=1:datarate(ff)
     Etx=Etx-(Elec*8*pktsize+Emp*8*pktsize);%1-(egy/bit+egy/mpath)-totaldata
     Erx=Eini-Etx;
     Erx=Erx-(Elec+EDA)*8*pktsize;
     Eini=Etx;
     if Etx<0.98
       pcktloss(ll)=1; %some data is lost
     else
       pcktloss(ll)=0;
     end
```

```
end
        if hop>4 && datarate(ff)> 6 % No of hops greater than 4 and data rate greater than 6
           pcktlossrate(1,ff)=(datarate(ff)-7)/datarate(ff); %packet loss decreases
        else
           pcktlossrate(1,ff)=0;
        end
        thrgput(1,ff)= (datarate(ff)*pktsize)/time_consumed;%amount of info that can be
processed in given time.
end
```

UrbanCitySimu.m

%This code is used to create an urban city environment with vehicles and rsus represented using nodes and lists out the paths that can be taken by source node to reach the destination using a neighbour table and routing tables.

```
function
[distance,energy,nw_liftime,throughput]=UrbanCitySimu(NumOfNodes,src_node,dst_node)
%UrbanCity is the function that takes NumOfNodes,src_node,dst_node as input
%arguments and returns the results using output variables distance,energy,nw_liftime,throughpu
%paths are stored in form of table
citysize=100;
axis([0 citysize+1 0 citysize+1]); %draws the plot with x-axis limit limiting from 0 to citysize+1
and y-axis from 0 to citysize+1
hold on %hold on retains plots in the current axes so that new plots added to the axes do not
delete existing plots
blksiz=30; %BLKSIZE=value Specifies the maximum length of a block for auxiliary trace data
sets.
Eini=1;% in joules
Range=20;
breadth = 0;
display_node_numbers = 1;
src_node1=src_node;
%%c = uicontrol creates a push button (the default user interface control)
            H = uicontrol('Style', 'listbox', ...
               'Units', 'normalized', ...
               'Position', [0.6 0.2 0.3 0.68], ...
```

```
          'String', {'Path Establishing...'});
        drawnow;
%Above mentioned code creates an object of style listbox with normalised
%units and string positioned at specified units and including  a string Path
%Establishing
%position Location and size, specified as a four-element vector of the form [left bottom width
height].
%%
%%%%%%%%%%%%%creating road network%%%%%%%%%
%Designing the plot
for len = 0:citysize
  if(rem(len,10)~=0) %returns the remainder when a is divided by b !=0
     for breadth = 0:citysize
        if(rem(breadth,10)~=0)
           h1 = plot(len,breadth,':g');
        end
     end
  end
  breadth = breadth+1;
end
%%%%%%%%%%%%%%%%%%%END1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for node_index = 1:NumOfNodes
  TempX = randi([0,citysize],1,1);
  if (rem(TempX,10)==0)
    %sprintf('TempX = %d\n',TempX);
    Node(node_index,1) = TempX;      %X co-ordinate in 1st column
    Node(node_index,2) = randi([0,citysize],1,1); %Y co-ordinate in 2nd column
    %sprintf('%d IF: X=%d Y=%d',node_index, Node(node_index,1),Node(node_index,2))
  else
    Node(node_index,2) = 10*(randi([0,citysize/10],1,1)); %Y co-ordinate in 2nd column
    Node(node_index,1) = randi([0,citysize],1,1); %X co-ordinate
    %sprintf('%d ELSE: X= %d Y= %d',node_index, Node(node_index,1),Node(node_index,2))
  end
end
%% Assign Positions to RSUs
m=1;
temp=1472014;
for ii=blksiz/2:2*blksiz:citysize
```

```
  n=1;
  for jj=blksiz/2:2*blksiz:citysize
     rsu.position{m,n}=[ii,jj]; % RSU's Position
     rsu.ID{m,n} =  temp;% RSU's ID
     plot(ii,jj,'xr','Linewidth',2)
     text(ii+1,jj, num2str(rsu.ID{m,n}))
     n=n+1;
     temp=temp+1;
  end
  m=m+1;
end
m=round((citysize/(2*blksiz))+1);
for ii=blksiz/2+blksiz:2*blksiz:citysize
  n=1;
  for jj=blksiz/2+blksiz:2*blksiz:citysize
     rsu.position{m,n}=[ii,jj];% RSU's Position
     rsu.ID{m,n} =  temp;% RSU's ID
     plot(ii,jj,'xr','Linewidth',2)
     text(ii+1,jj, num2str(rsu.ID{m,n}))
     n=n+1;
     temp=temp+1;
  end
  m=m+1;
end
rsu.origID=rsu.ID;
% combine nodes position and RSU positions in a single matrix
temp=reshape(rsu.position,numel(rsu.position),1);
temp=temp(~cellfun(@isempty, temp)); % delete empty cell in the matrix
for ii=1:numel(temp)
  temp1(ii,:)=temp{ii};
node_rsu=[Node;repmat(temp1,1,3)]; % combined matrix for rsu and nodes location
clear temp temp1
h2 = ones(NumOfNodes,1);
h4 = ones(1,1);
counter=1;
for n = 0:citysize
  for node_index = 1:NumOfNodes

     if(rem(Node(node_index,1),10)~=0)
        h2(node_index) = plot(Node(node_index,1)+n*(2*(rem(node_index,2))-1),
Node(node_index,2),'.k');

        node_rsu(node_index,3) = Node(node_index,1)+n*(2*(rem(node_index,2))-1);
```

```
  node_rsu(node_index,4) = Node(node_index,2);
            node_rsu(node_index,5) = rem(node_index,2)+2;
            if node_index==src_node1
                plot(node_rsu(node_index,3), node_rsu(node_index,4),'og');
                h7=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(src_node1));
            end
            if node_index==dst_node
                plot(node_rsu(node_index,3), node_rsu(node_index,4),'dm');
                h9=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(dst_node));
            end
        else
            h2(node_index) =
  plot(Node(node_index,1),Node(node_index,2)+n*(2*(rem(node_index,2))-1),'.k');

            node_rsu(node_index,3) = Node(node_index,1);
            node_rsu(node_index,4) = Node(node_index,2)+n*(2*(rem(node_index,2))-1);
            node_rsu(node_index,5) = rem(node_index,2);
            if node_index==src_node1
                plot(node_rsu(node_index,3), node_rsu(node_index,4),'og');
                h7=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(src_node1));
            end
             if node_index==dst_node
                plot(node_rsu(node_index,3), node_rsu(node_index,4),'dm');
                 h9=text(node_rsu(node_index,3), node_rsu(node_index,4)+1,num2str(dst_node));
            end
        end
    end

%%%%%%%%%NEIGHBORING TABLE%%%%%%%%%%%%%5
  % find all nodes which are in range of each other
  for p = 1:size(node_rsu,1)
    for q = 1:size(node_rsu,1)
        dist=sqrt((node_rsu(p,3)-node_rsu(q,3))^2+(node_rsu(p,4)-node_rsu(q,4))^2);
        if dist<=Range
            inrange(p,q)=1;
        else
            inrange(p,q)=0;
        end


    end
  end
  src_node=src_node1; % to reset teh src_node to original source node after every iteration of
n=1:citysize
```

```
rtngtble=src_node;% initialize
  tble1=src_node;% initialize
  tble=src_node;% initialize
  cnt=1;% initialise
  cnt1=1;% initialise
  dimnsn(cnt)=numel(rtngtble);
  while rtngtble~=dst_node

      for ii=1:numel(tble1)
          src_node=tble1(ii);
          temp=find(inrange(src_node,:));
          temp=temp(find(ismember(temp,tble)==0));
          str{cnt1}=[src_node,temp];
    tble=[tble, temp];
              cnt1=cnt1+1;
          end
          tble1=tble(find(ismember(tble,rtngtble)==0));% separate nodes which are not present in
    routing table
          rtngtble=[rtngtble,tble];
          % remove the repeated node in table
          [any,index]=unique( rtngtble,'first');
          rtngtble=rtngtble(sort(index));

          if ismember(dst_node,rtngtble)
              dst_cell=find(cellfun(@equal, str,repmat({dst_node},1,length(str)))); % find out which
    structre cell has destination node
              dst=dst_cell;
              nodtble=dst_node;
              frst_node=dst;
              while frst_node~=src_node1
                  frst_node=str{dst(1)}(1);
                  dst=find(cellfun(@equal, str,repmat({frst_node},1,length(str))));
                  nodtble=[nodtble, frst_node];
              end
    %           msgbox('path found')
              nodtble=fliplr(nodtble) % final routing table
              set(H, 'String', cat(1, get(H, 'String'), {['Path ' num2str(nodtble)]}));
              drawnow;
              pause(0.25);
```

```
            route{counter}=nodtble; % save all AODV paths for each change in vehicle position
into a structure
            h4= plot(node_rsu(nodtble,3),node_rsu(nodtble,4));
            pause(0.01);
             set(h4,'Visible','off');
             [E,pcktlossrate,total_dist,pcktloss,thrgput]=evaluation(nodtble,node_rsu); %
parameters calculation
             energy(counter,:)=E; % energy consumption
             distance(counter)=total_dist;% Total Distance between hops in AODV path
             throughput(counter,:)=thrgput; % throughput

                 counter=counter+1;
               end
              cnt=cnt+1;
              dimnsn(cnt)=numel(rtngtble);
              if numel(rtngtble)==1
                msgbox('1-No Node in range, Execute again')
                return
              end
              if cnt>=5
%                 h8=msgbox('No path found');
                break
              end

          end
          pause(0.0001);
          set(h2,'Visible','off');
          set(h7,'Visible','off');
          set(h9,'Visible','off');
        end
        %% plot results
        figure(2)
        plot(distance,'r','linewidth',2)
        xlabel('Number of times path found during simulation of VANET')
        ylabel('Dsiatnce in a path for each source and destination vehicles position')
        title(['Total Distnace in each linked path with hops=', num2str(cellfun('ndims',route(1)))])
        grid on
        figure(3)
        plot(energy,'Linewidth',1.5)
        xlabel('Number of times path found during simulation of VANET')
        ylabel('Energy in Joules')
        title('Energy Consumption')
```

```
legend('Data Rate=4 pckts/sec','Data Rate=6 pckts/sec','Data Rate=8 pckts/sec','Data Rate=10
pckts/sec','Data Rate=12 pckts/sec','Data Rate=14 pckts/sec')
grid on
figure(5)
plot(throughput/10e6,'Linewidth',1.5)
xlabel('Number of times path found during simulation of VANET')
ylabel('Throughput in MBps ')
title('Throughput plot for different data rates')
legend('Data Rate=4 pckts/sec','Data Rate=6 pckts/sec','Data Rate=8 pckts/sec','Data Rate=10
pckts/sec','Data Rate=12 pckts/sec','Data Rate=14 pckts/sec')
grid on
```

CHAPTER 3
SNAPSHOTS

3.1 EXPERIMENTAL ANALYSIS :

GUI of Simulation will be as follows :

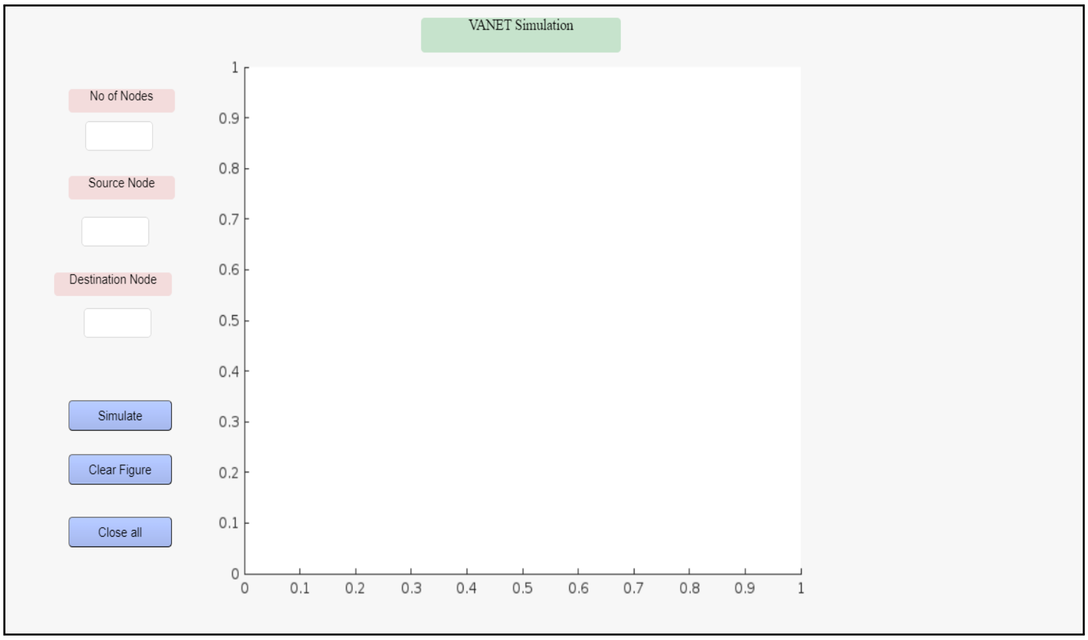

Fig 3.1.Initial Output with GUI

Entering the values of No of nodes,Source node number and destination node we start simulation. The results are as follows :

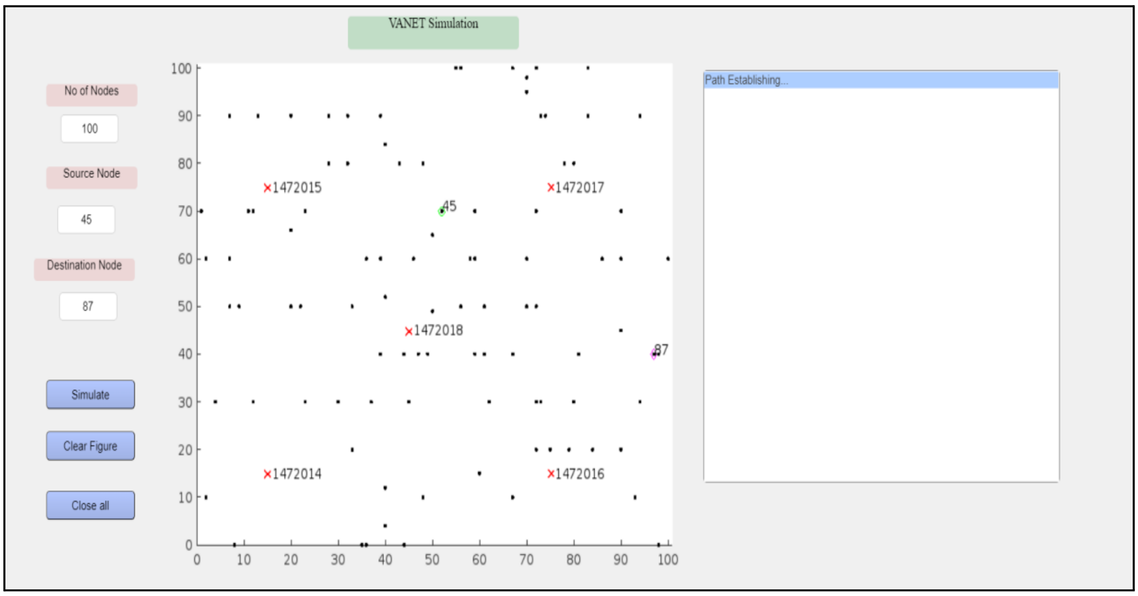

Fig 3.2.Initial Output with GUI

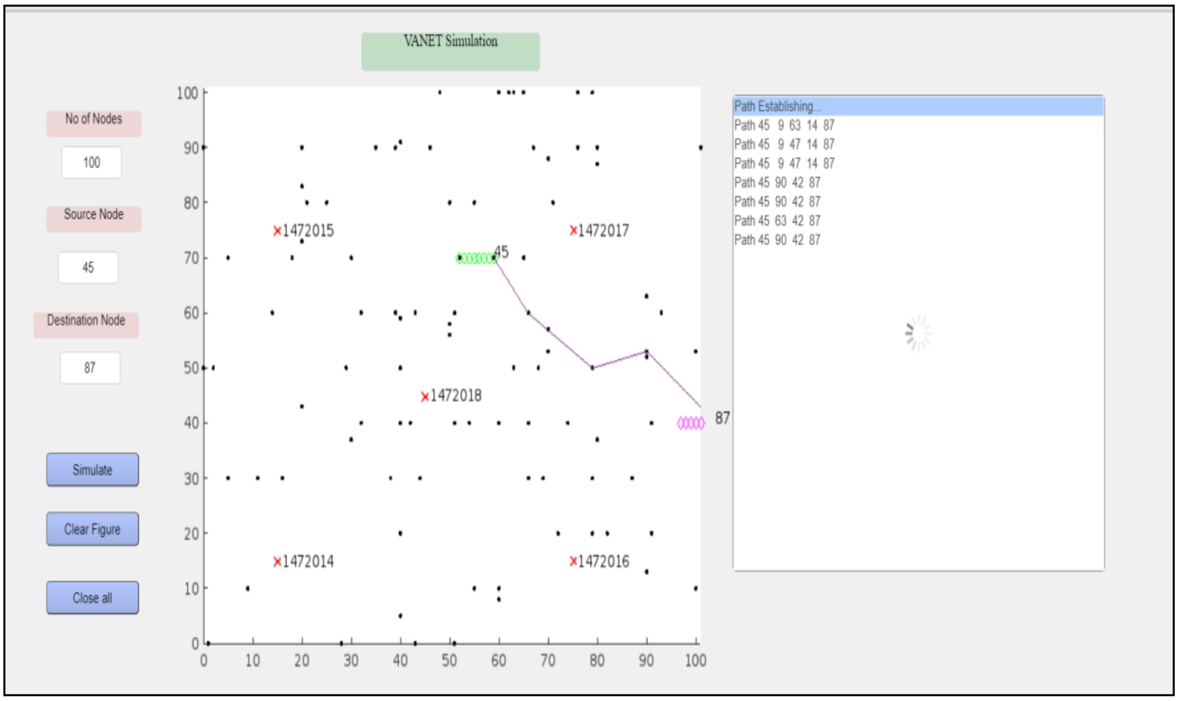

Fig 3.3. Line connecting S and D

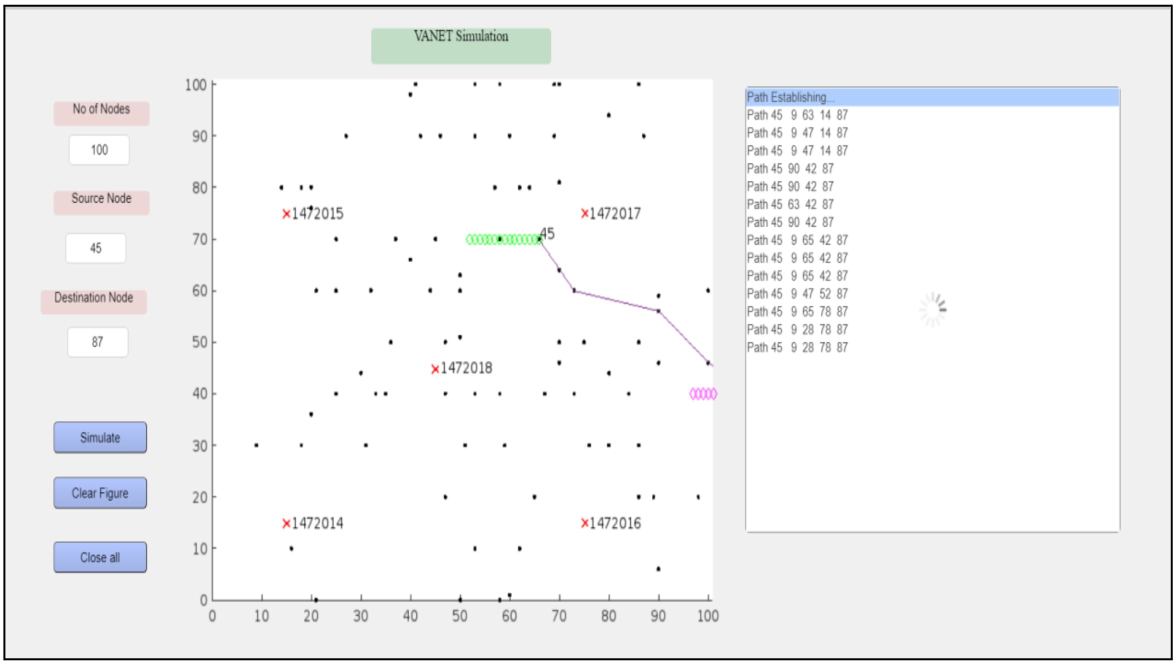

Fig 3.4.Intermediate Path

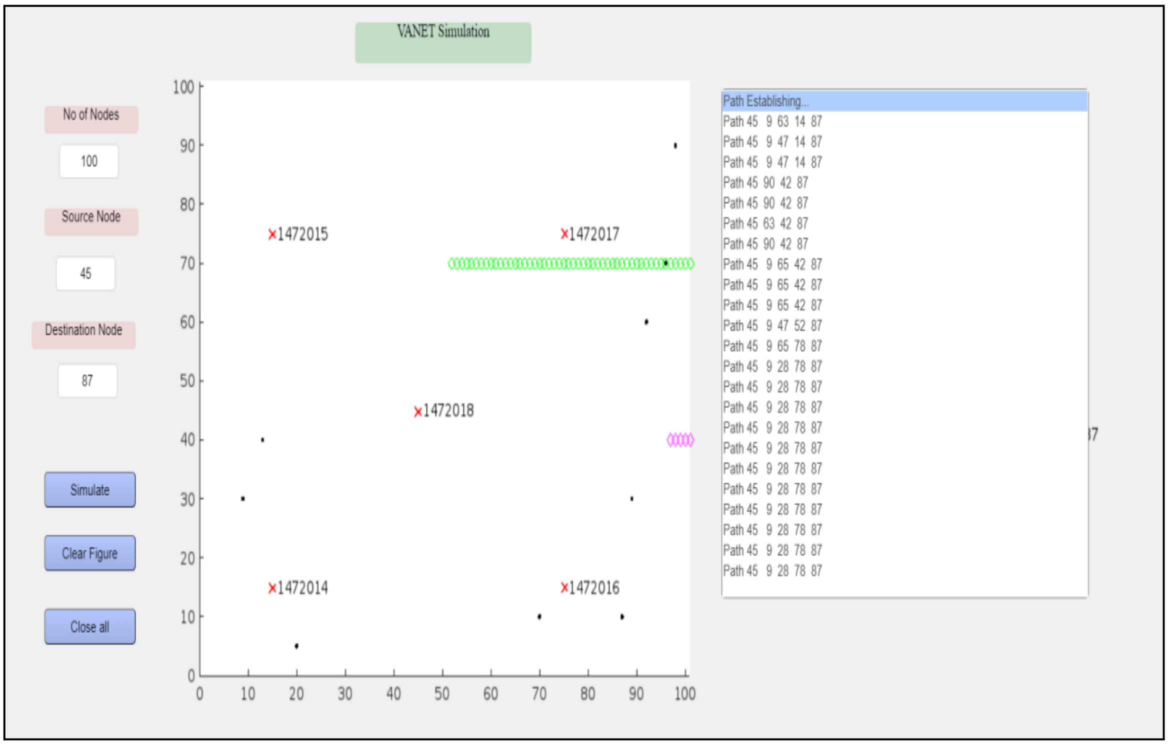

Fig 3.5.Final Path Establishment

Figure 3.3 shows connection between source and destination node.As we have already seen in the previous sections,the path between two nodes keeps changing because of frequent topology changes as it approaches towards the destination node because it works on finding the safest,reliable, and efficient path/route. After final establishment of the path, simulation comes to an end.

The final path established by DABFS is shown in Fig 3.4 on the right hand side of the image.The starting number(45) indicates the source node,the last number(79) indicates the destination node and the numbers in between indicate the path taken by the respective node to reach the next hop.We can see that the numbers in between source and destination node changes as we move closer to D.

The final path established between source and destination nodes is 45→ 9→ 28→78→ 87. Thus we can conclude the data sent by the source node travels through a variety of paths to reach its destination and DABFS protocol helps in establishing a path that results in less latency and packet loss.

3.2 PERFORMANCE EVALUATION

After the simulation is successfully implemented, graphs are plotted using various parameters in order to compare the working efficiency and reliability of DABFS Protocol. The results are as follows-

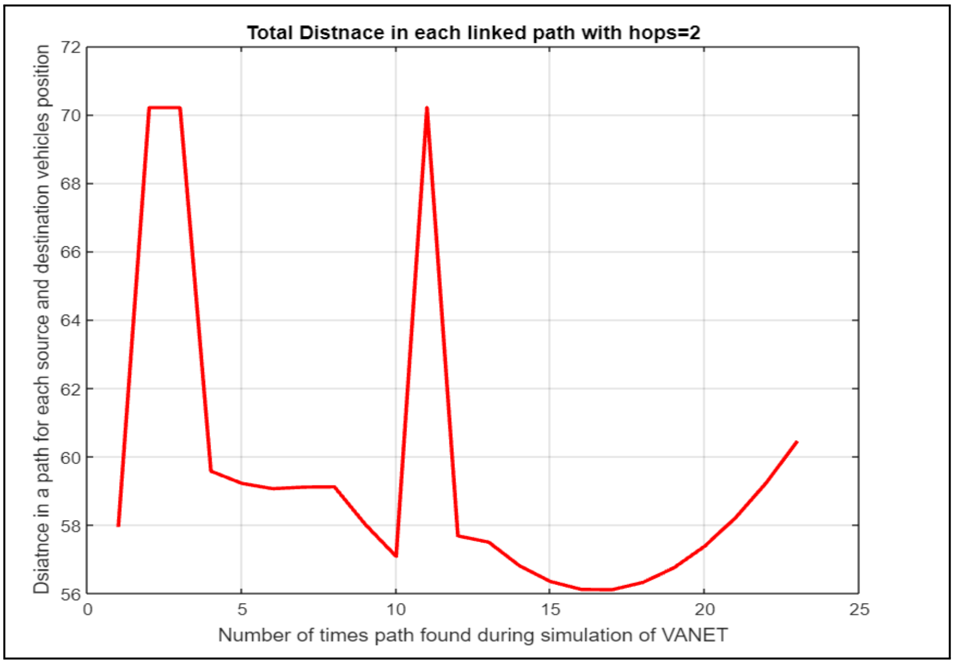

Fig 4.1.Distance Graph

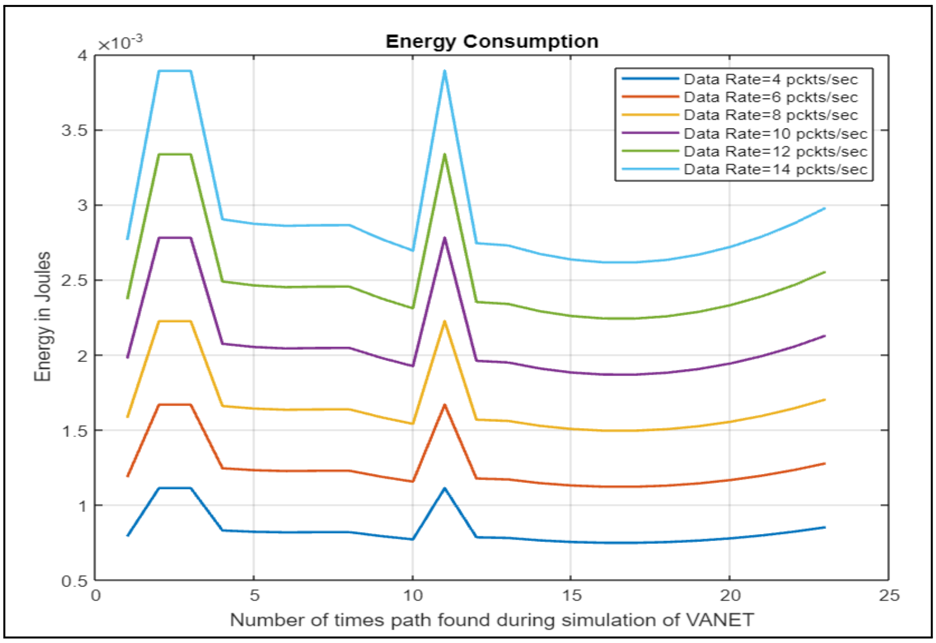

Fig 4.2.Energy Graph

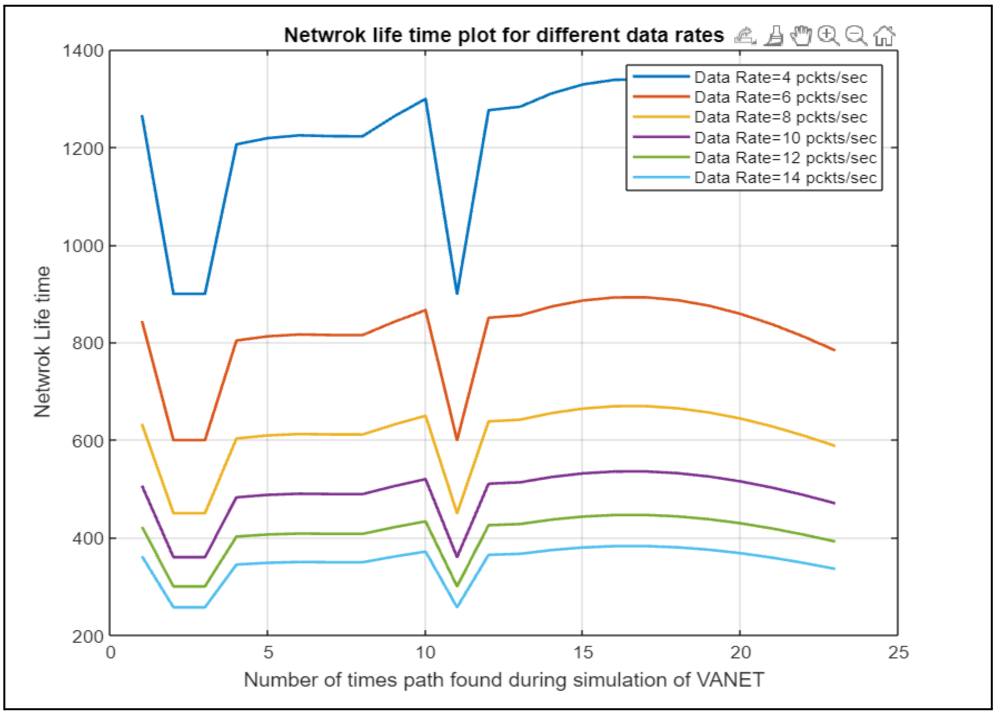

Fig 4.3.Network Life Time Graph

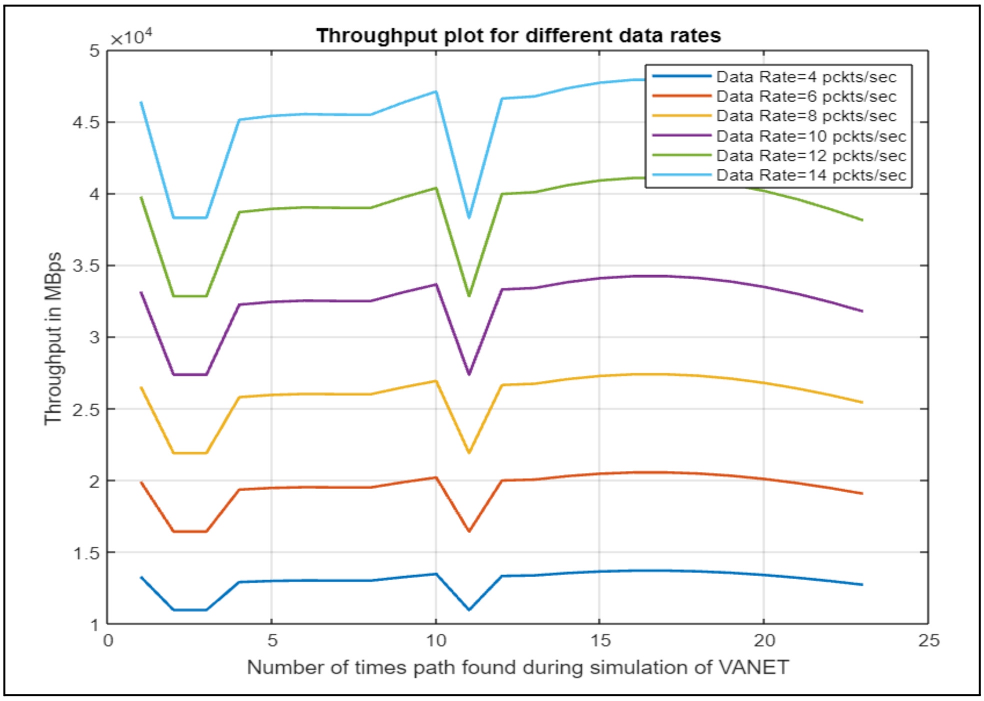

Fig 4.4.Throughput Graph

Distance Graph :

- Distance graph represents the distance in a path between source and destination on the y-axis and the number of times that particular path is found during simulation on the x-axis.
- From Fig 4.1 we can analyse the average number of times the message transmission sticks to a particular path and we can analyse how the path changes frequently because of frequent topology changes within the network at different data rates.

Energy Graph :

- Energy Graph plotted above represents total energy depleted during data transmission of packets from source to destination on the y-axis and number of times path found during simulation on x-axis.
- From Fig 4.2 we can analyse the energy consumed by each path while transmitting the data packets between source and destination at different data rates.

Network LifeTime Graph :

- Generally Network LifeTime is defined as the operational time of the network during which the network is able to perform the dedicated task.
- The graph is plotted between the number of paths found during simulation of vanets on the x-axis and network lifetime of each path on the y-axis at different data rates.

Throughput Graph :

- Throughput is the amount of data transmitted in unit time.Throughput graph above represents the throughput with the number of times path found during simulation on x-axis and throughput on y-axis.
- From Fig 4.4 we can analyse the throughput of each path while transmitting the data packets between source and destination at different data rates.

CHAPTER 4

4.1 CONCLUSION :

Avoiding collision has been our main aim throughout the completion of this project by transmitting warning messages from each node to the next node. We have introduced DABFS for a reason as it considers not only the distance parameter but also other two parameters as discussed above which is important for bi-directional highway environments and also provide us with the best route. These parameters help us to select the appropriate next hop. This leads to increased throughput and reduced packet loss and delay. Simulation results establish the path taken by the nodes and corresponding graphs using various parameters is plotted.

4.2. FUTURE WORKS :

Future works include the extension of DABFS in urban environments with appropriate use of GPS localization. Also, ensuring secure transmission of the set warning messages can be one of the extensions in the near future.

CHAPTER 5

REFERENCES

[1] Tanwee Kausar, Priyanka Gupta, Deepesh Arora, Rishabh Kumar(April 2012) - A VANET based Cooperative Collision Avoidance System for a 4-Lane Highway.

[2] Hidayatulah Himawan, Aslinda Hassan, **and Nazrul Azhar Bahaman**(2022) - Performance Analysis of Communication Model on Position Based Routing Protocol.

[3] Alzamzami, O., & Mahgoub, I. (2020). Link Utility Aware Geographic Routing for UrbanVANETs using Two-Hop Neighbor Information. Ad Hoc Networks, 102213.

[4] Bujari, A., Conti, M., De Francesco, C., & Palazzi, C. E. (2019). Fast multi-hop broadcast of alert messages in VANETs: An analytical model. Ad Hoc Networks, 82, 126-133.

[5] Darwish, T. S., Bakar, K. A., & Haseeb, K. (2018).Reliable intersection-based traffic aware routing protocol for urban areas vehicular ad hoc networks.IEEE IntelligentTransportation Systems Magazine, 10(1), 60-73.

[6] Husain, A., Singh, S. P., & Sharma, S. C. (2020). PSO Optimised Geocast Routing In VANET. Wireless Personal Communications, 115(3), 2269-2288.

[7] Hosmani, S., & Mathapati, B. (2020). Efficient Vehicular Ad Hoc Network routing protocol using weighted clustering technique. International Journal of InformationTechnology, 1-5.

[8] Haider, S., Abbas, G., Abbas, Z. H., & Baker, T. (2019). DABFS: A robust routing protocol for warning messages dissemination in VANETs. Computer Communications, 147, 21-34.

[9] Karimi, R., & Shokrollahi, S. (2018). PGRP: Predictive geographic routing protocol for VANETs. Computer Networks, 141, 67-81.

[12] Purkait, R., & Tripathi, S. (2020). Fuzzy logic based multi-criteria intelligent forward routing in VANET. Wireless Personal Communications, 111(3), 1871-1897.

[13] Silva, A., Reza, N., & Oliveira, A. (2019). Improvement and performance evaluation of GPSR-based routing techniques for vehicular ad hoc networks. IEEE Access, 7,21722-21733.

[14] Srivastava, A., Prakash, A., & Tripathi, R. (2020). Location based routing protocols in VANET: Issues and existing solutions. Vehicular Communications, 23, 100231.

[15] Wang, C., Fan, Q., Chen, X., & Xu, W. (2018, March). Prediction based greedy perimeter stateless routing protocol for vehicular self-organising network. In IOPConf. Series: Materials Science and Engineering (Vol. 322, pp. 1-6)

[16] Yang, X., Li, M., Qian, Z., & Di, T. (2018). Improvement of GPSR protocol in vehicular ad hoc network. IEEE Access, 6, 39515-39524

[17] Zahedi, K., Zahedi, Y., & Ismail, A. S. (2019). CJBR: connected junction-based routing protocol for city scenarios of VANETs. Telecommunication Systems, 72(4),567-578.

V. Naumov, T.R. Gross, Connectivity-aware routing (CAR) in vehicular ad-hoc networks, in: Proc. IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, May, 2007, pp. 1919–1927.

Q. Yang, A. Lim, S. Li, F. Jian, P. Agarwal, ACAR: Adaptive connectivity aware routing for vehicular ad hoc networks in city scenarios, Mob. Netw. Appl. 15 (1) (2010) 36–60.

G. Ajay, S. Kuldeep, Position based adaptive routing for VANETs, Int. J. Comput. Netw.Commun. 9 (1) (2017) 55–70.

S. Boussoufa-Lahlah, **F. Semchedine, L. Bouallouche-Medjkoune**, Geographic routing protocols for vehicular ad hoc networks (VANETs): A survey, Veh. Commun. 11 (2018) 20–31.

X. Yang, M. Li, Z. Qian, T. Di, Improvement of GPSR protocol in vehicular ad hoc network,IEEE Access 6 (2018) 39515–39524.

K. Zahedi, Y. Zahedi, A.S. Ismail, CJBR: connected junction-based routing protocol for city scenarios of VANETs, Telecommun. Syst. (2019) 1–12.

T. Lu, S. Chang, W. Li, Fog computing enabling geographic routing for urban area vehicular network, Peer-to-Peer Netw. Appl. 11 (4) (2018) 749–755.

A.Gupta,VANET Simulation in MATLAB, 2019, Available:https://www.mathworks.com/matlabcentral/fileexchange/68829-vanet-simulation-in-matlab.