VEHICULAR AD HOC NETWORK

BY

KHANDAKER SHAHNUR RAHMAN
ID: 101-15-900

AND

UMAM MUSTAIN

ID: 093-15-848

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By
ANISUR RAHMAN
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITYDHAKA, BANGLADESH

JUNE 2013

APPROVAL

This Project titled **"Vehicular Ad hoc Network**", submitted by Khandaker Shahnur Rahman and Umam Mustain to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 23 June 2013.

BOARD OF EXAMINERS

Dr Syed Akhter Hossain — **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr Yousuf Mahbubul Islam — **Internal Examiner**
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr Md Kabirul Islam — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr Muhammad Shorif Uddin — **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Anisur Rahman, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Anisur Rahman
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Submitted by:

Khandaker Shahnur Rahman
ID: 101-15-900
Department of CSE
Daffodil International University

Umam Mustain
ID: 093-15-848
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete this project successfully.

We fell grateful to and wish our profound our indebtedness to **Anisur Rahman, Assistant Professor,** Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of wireless network influenced us to carry out this project .His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr Syed Akhter Hossain**, Professor and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

(Khandaker Shahnur Rahman)
ID: 101-15-900
Department of CSE
Daffodil International University

(Umam Mustain)
ID: 093-15-848
Department of CSE
Daffodil International University

ABSTRACT

Routing in a vehicular ad hoc network is a challenge because of the high mobility of the nodes of the network. Nodes that are situated in vehicles move in different directions at different speeds. In a vehicular ad-hoc network speed of movement is therefore an issue of major concern. As nodes may also include stationary towers, scaling all the nodes in a fixed pattern is not possible. The speed of each node has to be considered. The speed of the nodes should be the scale to determine the hold-on time & update interval time as vehicles may move in and out of range very quickly. In this paper we present the existing routing protocols of wireless ad-hoc network. Also demonstrated the performance of proactive & reactive method through simulation results to show how efficiently they work with high speed nodes. The simulation results show that reactive protocol has a low overhead of control messages but latency is high on the other hand proactive protocol latency is low and routing happens on time. Given the advantages and disadvantages of each type of protocol when speed is taken as a factor we propose a hybrid protocol where proactive and reactive procedure will work in parallel to benefit from the advantages of each. We mainly emphasize on the amount of time a node will stay inside another nodes network zone and speed of every individual node. We have named of our proposed protocol as Speed Based Routing Protocol. The hybrid protocol would be able to track high-speed node and reduce connection drop. The next step would be constructing a simulator that can test the proposed hybrid protocol.

TABLE OF CONTENTS

CHAPTERS

LIST OF FIGURES

Chapter 1: INTRODUCTION

1.1 Introduction:

In the recent years the number of high speed of vehicles has increased exponentially. But the drivers have not stopped driving recklessly. Instead they have become more violent. These high speed vehicles made drivers more careless about traffic laws. Roads of today are saturated, safety distances and speed limits are hardly upheld, and drivers often lack attention given the use of personal mobile devices. Without an improvement in the traffic system, we can never rest assured about the future of transportation system. And thus drivers cause accidents and traffic jams.

Vehicular ad-hoc network is such a technology that can ensure the maintenance of traffic rules and regulation. By applying this technology we can save life, save time, corruption, vehicle security, avoid collision and so on.

In vehicular ad-hoc networks the vehicles will act as wireless nodes and these nodes would form a network enabling communicating with each other without the help of any form stationary infrastructure. Nodes within network range can help others communicate by using each other's communication channels. They would do this by using other nodes as intermediate routers to forward packets through multiple-hop routing. The Vehicular Ad Hoc Network (VANET) is a part of Mobile Ad Hoc Network (MANET). Every node or vehicle can move and freely communicate with each other by wireless technology. The communication may be of various types such as node-to-node (N2N), node to multi node (N2MN), Node To Road Side Unit (N2RSU) and road side unit to node (RSU2N).

1.2 Motivation:

The VANET is a revolutionary system for the modern era. A system that can help controlling the traffic & help the driver to avoid accident and also save time & resource can be a miracle of modern technology.

In modern life communication is a must & a prime media of communication is transportation. Without the means of transportation life becomes impossible. To reach our destination vehicle is a must. But with the convenience that the vehicles offer us it causes some mischief i.e. traffic collision, traffic jam, unauthorized races and so on. To control these phenomenon traffic rules have been put into place, traffic police are enforced. But just because there are rules doesn't mean that everyone will abide by it & in many cases traffic police aren’t enough to control these menace.

To enhance the traffic system VANET can be really helpful. It is able to predict traffic density, predict the best route, give warnings, send information, provide speed control & traffic control, provide emergency contacts and provide many more facilities. The possibility VANET offers is beyond imagination. It can add a new dimension to the vehicular control system.

VANET is new and interesting system. It is still under development but the features it may contain and the facilities it may provide to the future generation and the possibilities it reveals is just amazing. And that’s why we were interested to do a research on this topic.

1.3 Research Challenges:

VANET create some unique characteristics according to the behavior of driving, high speed vehicle and network communication between two nodes and RSU. These characteristics can be distinguished from other technologies.

1. **High speed mobility and rapid network changing topology:** Vehicles move fast on the road especially on the highway it moves faster as it possible. So definitely it will be quite difficult to keep communication with the high speed node. In this circumstance the communication link brake fast and it stay for few second. As a result the communication links have to build again. It harms for frequent communication.
2. **Network topology:** Network topology is the critical part in this technology. We need to involve different type of network topology to ensure maximum accuracy in data communication.
3. **Vehicular moving path prediction:** All over the world public vehicle path are pre defined and most of the private vehicle move to same path every day. According to their moving path a prediction can be made for those vehicle what is so important for traffic system. As a result we can know the possible traffic for a road with respect to time. There are few vehicles that move different path for different day after prediction we can find those vehicle that move different path. This prediction can be made by using VANET technology.
4. **Available geographic position and direction:** A digital map should be integrated to the vehicle to locate its position. For example, Global Positioning System(GPS) finds our

location with respect to the map. In VANET system, vehicle must be equipped with accurate positioning systems integrated by electronic maps.

5. **Communication delay:** In VANET system the data communication must be fast. A notification must be sent to all the nodes if there is a collision in the network. If the pre-collision or collision and other signal make any delay to reach to others vehicle it will harm to the system.
6. **Security issue:** Security is a major challenge for any technology. There are several security challenges in this technology. VANET can handle those challenges easily. And VANET ensures some important security for vehicle and system.
7. **Equipment and Bandwidth:** This is not major concern for developed country but this is most important concern for developing & poor country.
8. **Power supply on the network node:** For uninterrupted communication, power supply is a must requirement.

Chapter 2: VEHICULAR AD-HOC NETWORK (VANET)

2.1 VANET [1]:

A Vehicular Ad-Hoc Network or VANET is a technology that uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of each other to connect and, in turn, create a network with a wide range. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is created. It is estimated that the first systems that will integrate this technology are police and fire vehicles to communicate with each other for safety purposes.

2.2 Characteristics [38]:

VANET has some unique characteristics which make it different from MANET as well as challenging for designing VANET applications.

1. **High dynamic topology:** The topology of VANET changes because of the movement of vehicles at high speed. Suppose two vehicles are moving at the speed of 20m/sec and the radio range between them is 160 m. Then the link between the two vehicles will last 160/20 = 8 sec.
2. **Frequent disconnected network:** From the highly dynamic topology results we observe that frequent disconnection occur between two vehicles when they are exchanging information. This disconnection will occur most in sparse network.
3. **Mobility modeling:** The mobility pattern of vehicles depends on traffic environment, roads structure, the speed of vehicles, driver’s driving behavior and so on.
4. **Battery power and storage capacity:** In modern vehicles battery power and storage is unlimited. Thus it has enough computing power which is unavailable in MANET. It is helpful for effective communication & making routing decisions.
5. **Communication environment:** The communication environment between vehicles is different in sparse network & dense network. In dense network building, trees & other objects behave as obstacles and in sparse network like high-way this things are absent. So the routing approach of sparse & dense network will be different.

6. **Interaction with onboard sensors:** The current position & the movement of nodes can easily be sensed by onboard sensors like GPS device. It helps for effective communication & routing decisions.

2.3 Application: List of application:

VANET will provide the user with various application and features.

a) **Data transfer:** As VANET is an ad-hoc network data transfer is a common feature. If we connect the VENAT with the internet backbone then the user will be able to download & upload files from the internet while remaining in the vehicle.

b) **Warning:** There are various warning message the VANET will have to produce in order to provide security. Such as

- **Clash:** If the distance between two vehicles decreases or any scenario arises where a collision may take place, a warning message is sent to those vehicles.

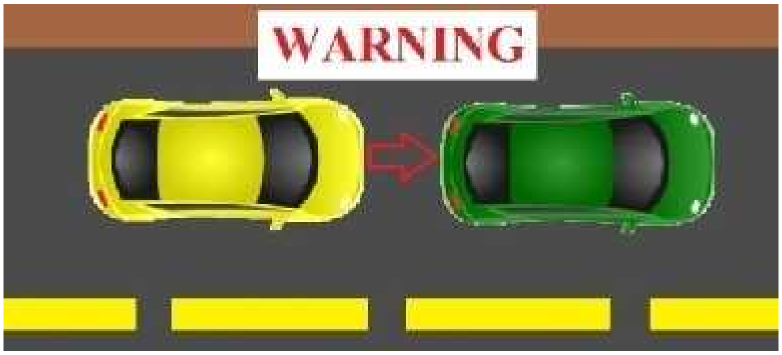


Figure 2.1: Clash warning

- **Traffic jam:** If a vehicle is moving towards traffic jam then that vehicle will be waned and if there is an alternative path, it will be suggested (figure).

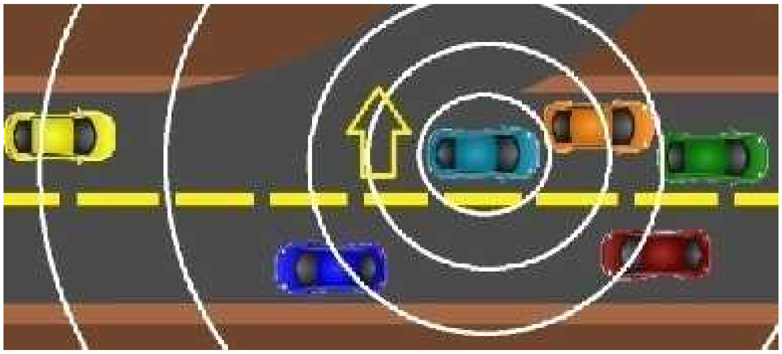


Figure 2.2: Warning about traffic jam

- **Accident:** If an accident occurs then the VANET broadcasts a warning and help request message to the neighboring vehicles, hospitals and police stations.

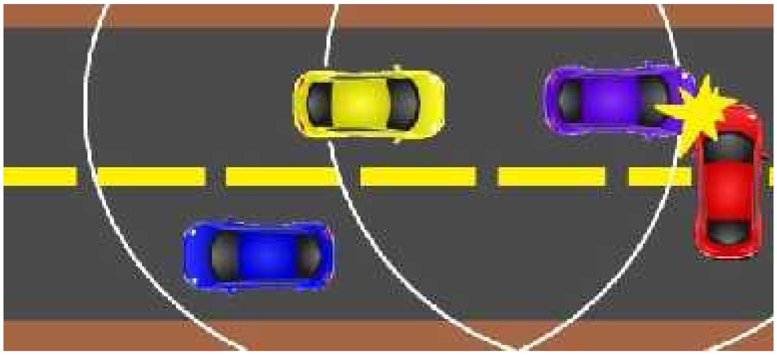


Figure 2.3: Accident warning

- **Speed:** If a vehicles' speed exceed the speed limit then it warns the driver and also warns the nearby police control rooms.

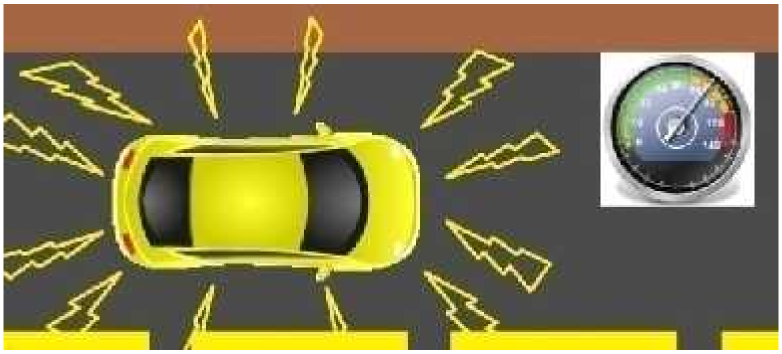


Figure 2.4: Warning on exceeding speed limit

c) **Path prediction:** VANET can predict the better route so that one can save time and avoid traffic jam.

d) **Traffic Control:** The flow of traffic and its density can be controlled using VANET

e) **Violence Control:** Any type of violent act using any vehicle can be controlled by VANET. Police vehicle can warn, mark or even stop a violent vehicle.

f) **Speed Control:** If a vehicle is speeding out of its limit amy police vehicle can stop it with the help of VANET.

g) **Traffic Density Control:** if a road is becoming densely populated with vehicles and might cause traffic jam then VANET directs the vehicles that are moving towards that route to an alternative route.

h) **Emergency Contact:** Incase of any accidents or any other emergencies the system can send help requests to nearby hospitals, fire-services, police control room etc accordingly.

Chapter 3: STUDY ON VANET

3.1 **Routing protocols:**

The characteristic of highly dynamic topology makes the design of efficient routing protocols for VANET is challenging. Here is a figure of routing protocol list that has been discussed in this paper.

Fig 3.0: Types of routing protocol.

3.2 Topology Based Routing:

This routing protocol use link information that exist in the network to perform packet forwarding.

There is three type of topology based routing

1. Reactive
2. Proactive
3. Hybrid

3.2.1 Reactive:

Reactive routing protocol is called on demand routing because it starts route discovery when a node needs to communicate with another node thus it reduces network traffic.

Characteristics:

- Route happens on demand.
- No background routing table.
- Route discovery is here.
- Don’t share link information when routing happened.
- No node maintain routing table.
- Don’t need to update routing table.
- Route discovery packet is used to find the destination.
- Route is not always available on request.
- Storage requirement is less then proactive.
- Delay is more than proactive.
- Require bandwidth depends on route request.

It is not suitable for real time traffic because of high latency

a) AODV: Ad Hoc on Demand Distance Vector [1, 2]

In AODV [1] the network remains still until routing is in order. When route is wanted the source node broadcasts request message. When the destination receives that request, it sends back a reply message through a temporary path to the source node. The source then starts linking using the route that has the smallest amount of hops. The unused routes of the routing tables are discarded after a while. If a link falls, then a routing error message is passed back to the source, and the process starts again.

Each request for route has a different sequence number [2]. Nodes use this sequence number to avoid repeating a route requests that they have already passed on. Every route requests has a "time to live" number that limits the time that they can be retransmitted. If a route request fails, another request may not be sent instantly, it may wait until twice as much time has passed since the timeout of the previous request.

Characteristics [2]

- Reactive protocol
- Descendant of DSDV
- Route discovery(RD) packets are used for route finding
- Active routing maintained
- Sequence number used for loop prevention and
- Provides uni-cast and multi-cast communication

Routing procedure [2]:

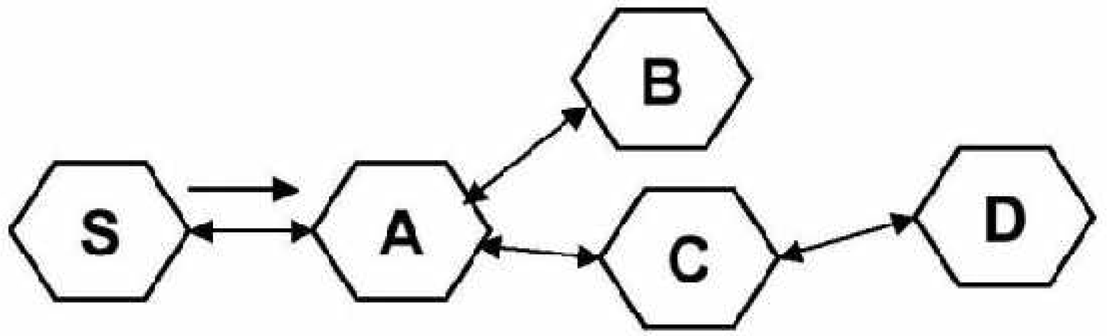


Fig 3.1: AODV routing [2]

1. Node S needs a route to D
2. Create a route request (RREQ)
    - Enters D’s IP address, sequence number, S’s IP address, sequence number
    - Broadcasts RREQ to neighbors
3. Node A receives RREQ
    - Makes reverse route entry for S
        - Dest = S, nexthop = S, hopcount = 0
    - It has no route to D, so it broadcasts RREQ
4. Node C receives RREQ
    - Makes reverse route entry for S
        - Dest = S, nexthop = A, hopcount = 1

    - It has route to D && seq# for route D > seq# in RREQ
        - Creates a route reply (RREP)
    - Enters D’s IP address, sequence number, S’s IP address, hopcount
        - Unicasts RREP to A

5. Node A receives RREP
    - Unicasts RREP to S
    - Makes forward route entry to D
        - Dest = D, nexthop = C hopcount = 2

6. Node S receives RREP
    - Makes forward route entry to D
        - Dest = D, nexthop = A hopcount = 3
    - Sends data packets on route to D

Goals [2]

- Quick adaptation under dynamic link conditions
- Lower transmission latency
- Consume less network bandwidth (less broadcast)
- Loop-free property
- Scalable to large network

b) TORA: Temporally Ordered Routing Algorithm [1, 3, 72]

The Temporally-Ordered Routing Algorithm [1] is an algorithm for routing data across Wireless Mesh Networks or Mobile ad-hoc networks. TORA is a protocol based on the concept of link reversal method.

Characteristics:

- Loop free.
- Establish routes quickly.
- Establishes multiple routes
- Minimizes algorithmic communication overhead.

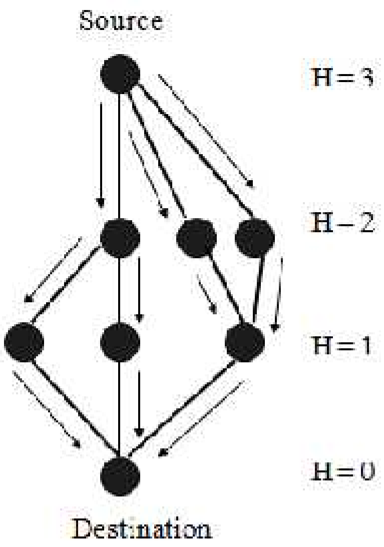


Fig 3.2: TORA routing [3]

The protocol performs three basic functions [72]:

- Route creation: Demand driven “query/reply”.
    - A query packet is flooded throughout the network.
    - An update packet propagates back if routes exist.
- Route maintenance: Update packets re-orient the structure.
- Route erasure: A clear is flooded throughout the network to erase invalid routes.

Link reversal algorithm [72]:

When a node has no downstream links it reverses the direction of one or more links.

Links are directed based on a metric, maintained by the nodes in the network that can conceptually viewed as a “height”.

Height metric [72]:

Each node i has a height: a quintuple (⊤, oid, r, 𝒹, i) where

- ⊤: logical time of a failure.
- oid: unique ID of the node that defined the reference level.
- r: “reflection” indicator bit.
- 𝒹:”propagation” ordering parameter.
- i: unique ID of the node.

Heights can be ordered lexicographically. The ordering of non-null height forms a Directed Acyclic Graph (DAG).

c) DSR: Dynamic Source Routing [1, 4]

Dynamic Source Routing [1] is a routing protocol for wireless mesh networks. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device.

The protocol consists of two major phases [4]:

- Route Discovery,
- Route Maintenance.

When a mobile node wants to send a packet to some destination, it consults its route cache to check whether it has a route to that destination or not. If there is any unexpired route, it will use this route. If not, it initiates route discovery by broadcasting a Route Request packet. Route Request contains the address of the destination and the source address.

Each node that receives the packet checks if it has any route to the destination. If it does not, it adds its own address to the route record of the packet and forwards it to the next node. When the request reaches either the destination or an intermediate node that has an unexpired route to that destination in its cache, a route reply is generated. If the node generating the route reply is the destination, it places the route record contained in the route request into the route reply.

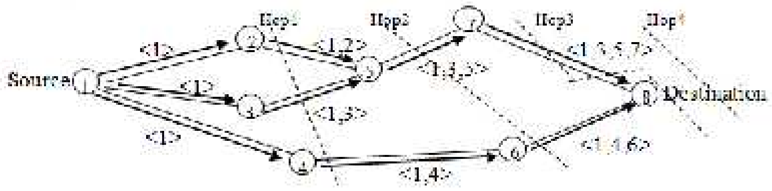


Fig 3.3: Building Record Route during Route Discovery [4]

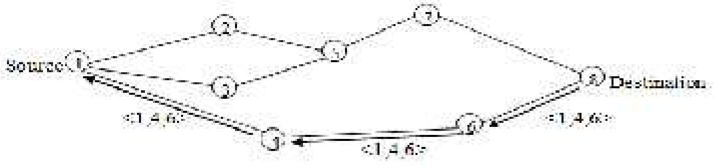


Fig 3.4: Propagation of Route Reply with the Route Record [4]

3.2.2 Proactive:

Proactive routing protocols are mostly based on shortest path algorithms. They keep information of all connected nodes in form of tables because these protocols are table based. Furthermore, these tables are also shared with their neighbors. Whenever any change occurs in network topology, every node updates its routing table.

Characteristics:

1. Route happens according to the background routing table.
2. Possible routing destination stored in the routing table.
3. No need to route discovery.
4. Share link information when routing happened.
5. Each and every node maintain routing table.
6. Routing table update itself in every specified time.
7. To update route information every node needs to exchange topology information on a regular basis and it makes high overhead on the network.
8. Route is always available on request.
9. Storage requirement is high.
10. Delay is less then reactive.
11. Scalability label generates up to 100 node OLSR and TBRPF may scale higher.

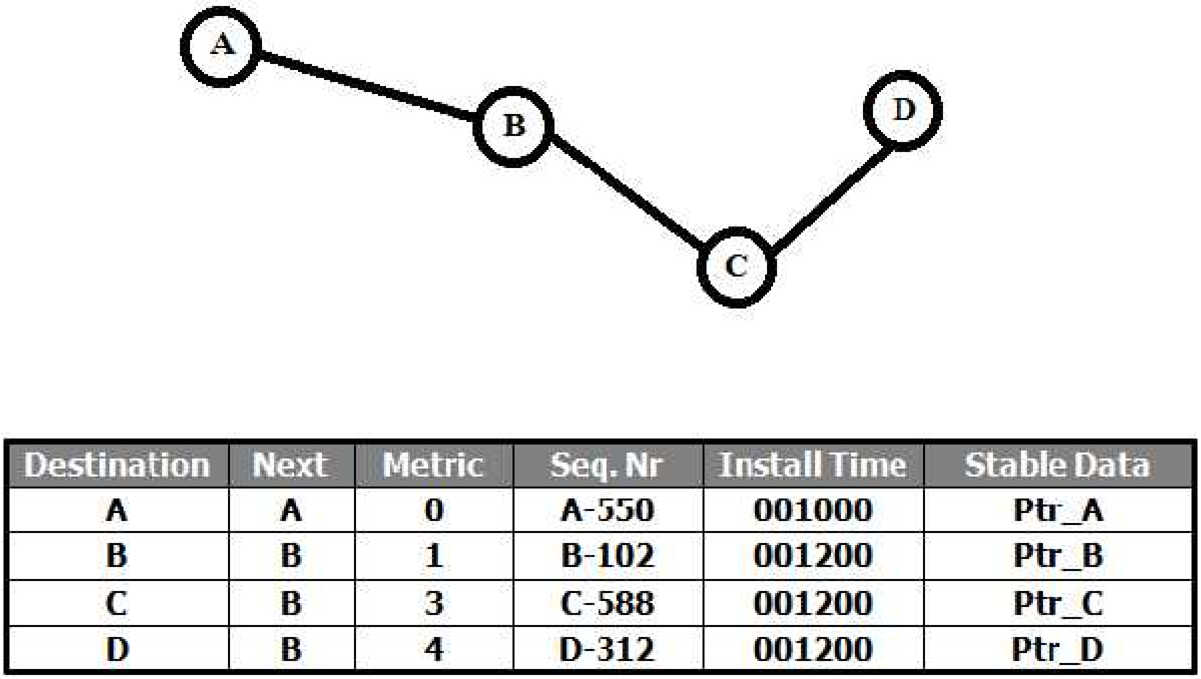
a) DSDV: Destination-Sequenced Distance-Vector [1, 2, 73]

DSDV [1] is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm.

Each entry in the routing table has a sequence number [2], if a link is present then the sequence numbers are even; otherwise an odd number is used.

The sequence number is generated by the destination, and the emitter must send out the next update with this number. Routing information is distributed between nodes through sending full dumps infrequently and smaller incremental updates more frequently. This routing adds three things to distance-vector routing [73]:

- **Sequence number** originated from destination. Ensures loop freeness.
- **Install Time** when entry was made (used to delete stale entries from table)
- **Stable Data** Pointer to a table holding information on how stable a route is. Used to damp fluctuations in network.



Destination	Next	Metric	Seq. Nr	Install Time	Stable Data
A	A	0	A-550	001000	Ptr_A
B	B	1	B-102	001200	Ptr_B
C	B	3	C-588	001200	Ptr_C
D	B	4	D-312	001200	Ptr_D

Fig 3.5: DSDV routing [73]

Update technique [2]:

- Each node periodically transmits the updates, which includes its own sequences number, routing table updates
- It also send the routing table updates for important link changes
- When two routes to a destination is received from two different neighbors it chooses the one with greatest destination sequence number but if sequence numbers are equal, choose the smaller metric (hop count).

Routing update is done in two ways [2]:

- Full dump
    - Entire routing table is sent to the neighbor.
    - Transmits relatively infrequently when no movement of node occurs.
    - Appropriate when the network change is more frequent.

- Incremental
    - The entries that require changes are sent.
    - Transmitted more frequently.
    - Appropriate when the network is relatively stable

b) WRP: Wireless Routing Protocol [1, 5]

Wireless Routing Protocol [1] is a proactive uni-cast routing protocol. WRP is a table-based protocol that tries to maintaining routing information among all nodes in the network. WRP used an enhanced version of the distance-vector routing protocol, which used the Bellman-Ford algorithm to calculate paths.

Each node in the network is responsible for maintaining four tables [5]:

- **Distance Table:** It contains the network view of the neighbor of a node. It contains matrix where each element contains the distance and the penultimate node reported by a neighbor for a particular destination.
- **Routing Table:** It contains the up-to-date view of the network for all destinations. It keeps the shortest distance, the predecessor node, the successor node, and a flag indicating the status of the path.
    - The path status may be a simple path, or a loop, or the destination node not marked.
- **Link-Cost Table:** It contains the cost of relaying messages through each link. The cost of a broken link is infinity. It also contains the number of update periods passed since the last successful update was received from that link. This is done to detect link breaks.
- **Message Retransmission List:** It contains the sequence number of the update message, a retransmission counter, an acknowledgement-required flag vector with each entry per neighbor, and a list of updates sent in the previous message.
    - It records which updates in an update message need to be retransmitted and which neighbors should acknowledge the retransmission.

Method [5]:

- Nodes inform each other of link changes through the use of update messages.
- An update message is sent only between neighboring nodes and contains a list of updates, as well as a list of responses indicating which mobiles should acknowledge the update.
- Nodes send update messages after processing updates from neighbors or detecting a change in a link to a neighbor.
- In the event of loss of a link between two nodes, the nodes send update messages to their neighbors.
- The neighbors then modify their distance table entries and check for new possible paths through other nodes. Any new paths are relayed back to the original nodes so that they can update their tables accordingly.
- If a node is not sending message, it must send a HELLO message within the specified time period to ensure connectivity.
- Lack of messages from the node indicate the failure of that link, this may cause a false alarm.

c) CGSR: Cluster-head Gateway Switch Routing [1, 3, 4]

CGSR [1, 4] is a clustered multi-hop mobile wireless network with several heuristic routing schemes. A distributed cluster-head (CH) selection algorithm is used to elect a node as the cluster head. It modifies DSDV by using a hierarchical CH to route traffic.

Gateway nodes serve as bridge nodes between two or more clusters. A packet sent by a node is first routed to its CH and then the packet is routed from the CH to a gateway of another cluster and then to the CH and so on, until the destination cluster head is reached. Frequent changes in the CH may affect the performance of the routing protocol.

The source of the packet transmits the packet to its cluster-head. From this cluster-head, the packet is sent to the gateway node that connects this cluster-head and the next cluster-head along the route to the destination. The gateway sends it to that cluster-head and so on till the destination cluster-head is reached in this way. The destination cluster-head then transmits the packet to the destination.

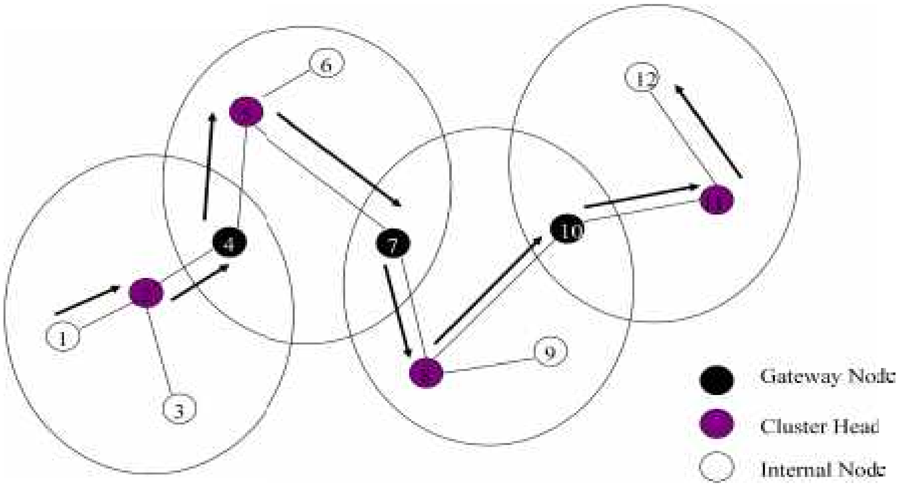


Fig 3.6: CGSR routing [3]

d) OLSR: Optimized Link State Routing Protocol [1, 6, 7]

The OLSR [1, 6] is an IP routing protocol optimized for mobile ad-hoc networks, which can also be used on other wireless ad-hoc networks. OLSR is a proactive link-state routing protocol, which uses hello and topology control (TC) messages to discover and then disseminate link state information throughout the mobile ad-hoc network. Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths.

The protocol achieves two main goals [7]:

- Optimized performance
- Full internet legacy

Characteristics [7]:

- Symmetric neighbors detection
- Multipoint relays
- Optimized flooding
- Partial link state
- Optimal route calculation

Main modules [7]

- Neighbor detection (link, MPR)
- Multipoint relay selection
- Optimized flooding
- Topology control
- Routing table

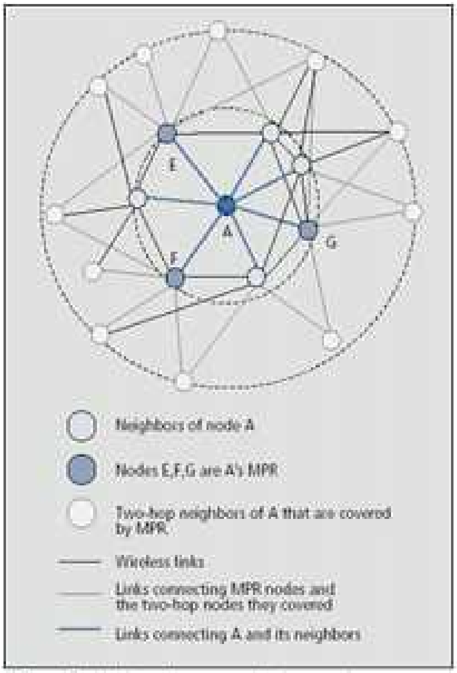

E
A
F
G
Neighbors of node A
Nodes E,F,G are A's MPR
Two-hop neighbors of A that are covered by MPR
Wireless links
Links connecting MPR nodes and the two-hop nodes they covered
Links connecting A and its neighbors

Fig 3.7: OLSR routing [6]

Route creation [7]

- Neighbor detection: Periodically broadcasted hello packets
  - Hellos advertize the heard nodes set
- Multipoint relay selection: Every node selects its multipoint relay set
- Optimized flooding: only multipoint relays forward broadcasts
  - All network nodes receive
  - A subset retransmits
- Topology control: Periodic broadcast of TC pack
  - TCs advertize MPR selector set
  - Much smaller than neighbor set

- Route calculation: Nodes know their neighbors and all the network MPR links
  - Route calculation on partial topology knowledge
  - Routes are optimal for the full topology

e) FSR: Fisheye State Routing [3, 8, 9]

Fisheye State Routing [8] is an implicit hierarchical routing protocol. Also considered a proactive protocol and is a link state based routing protocol that has been adapted to the wireless ad hoc environment. Relays on link state protocol as a base, and it has the ability to provide route information instantly by maintaining a topology map at each node. Thus it will maintain updated information from the neighbor node through a link state table. In each node the network, a full topology map is stored then utilized.

According to Kleinrock and Stevens, FSR uses the "fisheye" technique where the technique was used to reduce the size of information required to represent graphical data. The eye of a fish captures with high detail the pixels near the focal point. The detail decreases as the distance from the focal point increases.

In routing [9], the fisheye approach translates to maintaining accurate distance and path quality information about the immediate neighborhood of a node, with progressively less detail as the distance increases.

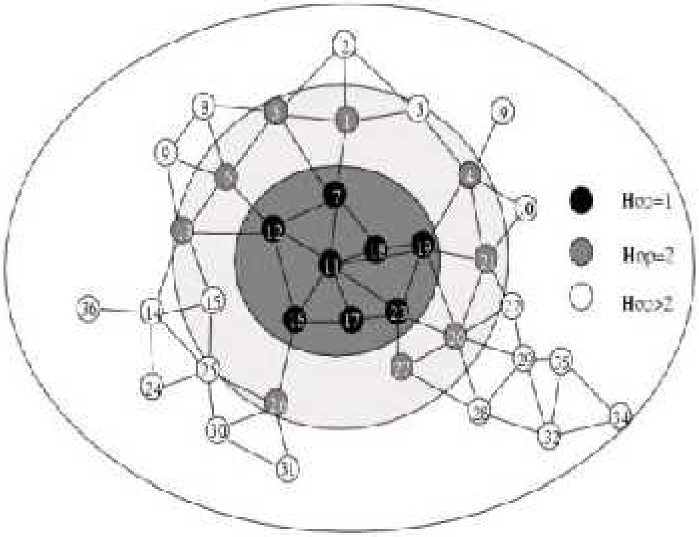
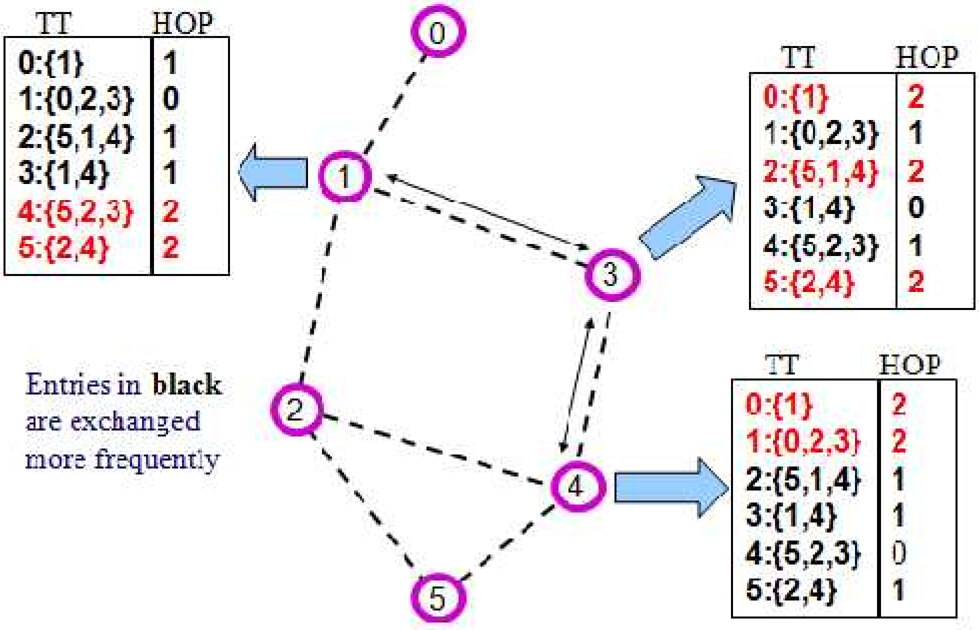


Fig 3.8: FSR routing [3]

Routing [9]

- For routing this approach translates into an accurate information in the immediate neighborhood of a node and less detail as the distance increases.
- FSR is similar to link state (LS) routing in that each node maintains a view of the network topology with a cost for each link.
- In LS routing link state packets are flooded into the network whenever a node detects a topology change.
- In FSR nodes maintain a topology table (TT) based on the up-to-date information received from neighboring nodes and periodically exchange it with their local neighbors.
- For large networks in order to reduce the size of the routing update messages the FSR technique uses different exchange periods for different entries in the routing table.
- Relative to each node the network is divided in different scopes.



TT	HOP
0:{1}	1
1:{0,2,3}	0
2:{5,1,4}	1
3:{1,4}	1
4:{5,2,3}	2
5:{2,4}	2

TT	HOP
0:{1}	2
1:{0,2,3}	1
2:{5,1,4}	2
3:{1,4}	0
4:{5,2,3}	1
5:{2,4}	2

TT	HOP
0:{1}	2
1:{0,2,3}	2
2:{5,1,4}	1
3:{1,4}	1
4:{5,2,3}	0
5:{2,4}	1

Entries in **black** are exchanged more frequently

Fig 3.9: Table update in FSR [9].

3.2.3 Hybrid: This protocol is the combination of Proactive and reactive protocol.

ZRP: Zone Routing Protocol [1, 10, 11]

In wireless networking, Zone Routing Protocol [1] was the first hybrid routing protocol with both a proactive and a reactive routing component. ZRP was proposed to reduce the control overhead of proactive routing protocols and decrease the latency caused by route discovery in reactive routing protocols. ZRP defines a zone around each node consisting of the node's *k*-neighborhood. A proactive routing protocol, Intra-zone Routing Protocol (IARP), is used inside routing zones, and a reactive routing protocol, Inter-zone Routing Protocol (IERP), is used between routing zones. A route to a destination within the local zone can be established from the source's proactively cached routing table by IARP. Therefore, if the source and destination of a packet are in the same zone, the packet can be delivered immediately. Most of the existing proactive routing algorithms can be used as the IARP for ZRP.

For routes beyond the local zone, route discovery happens reactively. The source node sends a route request to the border nodes of its zone, containing its own address, the destination address and a unique sequence number. Border nodes are nodes which are exactly *k* hops away from the source. Each border node checks its local zone for the destination. If the destination is not a member of this local zone, the border node adds its own address to the route request packet and forwards the packet to its own border nodes. If the destination is a member of the local zone, it sends a route reply on the reverse path back to the source. The source node uses the path saved in the route reply packet to send data packets to the destination.

Routing [10, 11]:

- The routing in ZRP is divided into two parts
  - Intrazone routing : First, the packet is sent within the routing zone of the source node to reach the peripheral nodes.
  - Interzone routing : Then the packet is sent from the peripheral nodes towards the destination node.

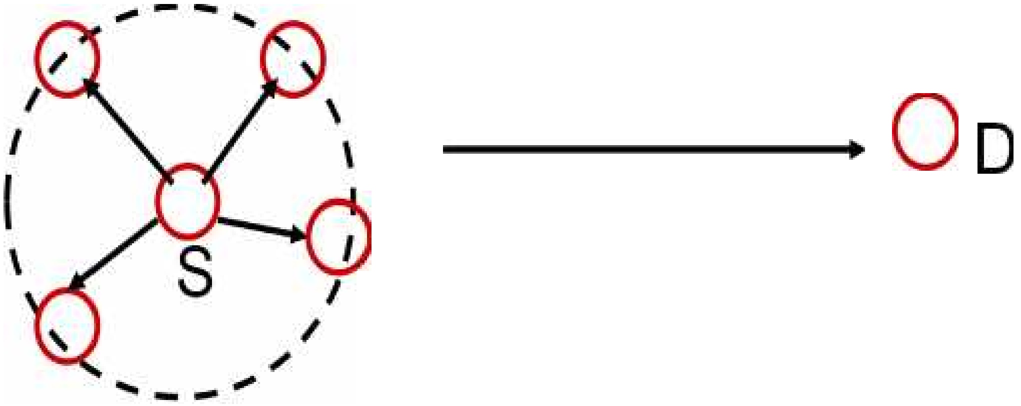


Fig 3.10: ZRP routing [11]

- Each node collects information about all the nodes in its routing zone proactively. This strategy is similar to a proactive protocol like DSDV.
- Each node maintains a routing table for its routing zone, so that it can find a route to any node in the routing zone from this table.
- In the original ZRP proposal, intrazone routing is done by maintaining a link state table at each node.
- Each node periodically broadcasts a message similar to a hello message kwon as a zone notification message.
- The zone radius is k for k>1
- A zone notification mesage dies after k hops, so after reaching the node´s neighbours at a distance of k hops the notification mesage dies.
- Each node receiving this message decreases the hop count of the message by 1 and forwards the message to its neighbours.
- The message is not forwarded any more when the hop count is 0.
- Each node P keeps track of its neighbour Q from whom it received the message through an entry in its link state table.
- P can keep track of all the nodes in its routing zone through its link state table.
- The interzone routing discovers routes to the destination reactively.
- Consider a source (S) and a destination (D). If D is within the routing zone of S, the routing is completed in the intrazone routing phase.

- Otherwise, S sends the packet to the peripheral nodes of its zone through bordercasting.
- The bordercasting to peripheral nodes can be done mainly in two ways :
  - By maintaining a multicast tree for the peripheral nodes. S is the root of this tree.
  - Otherwise, S maintains complete routing table for its zone and routes the packet to the peripheral nodes by consulting this routing table.
- S sends a route request (RREQ) message to the peripheral nodes of its zone through bordercasting.
- Each peripheral node P executes the same algorithm.
  - First, P checks whether the destination D is within its routing zone and if so, sends the packet to D.
  - Otherwise, P sends the packet to the peripheral nodes of its routing zone through bordercasting.

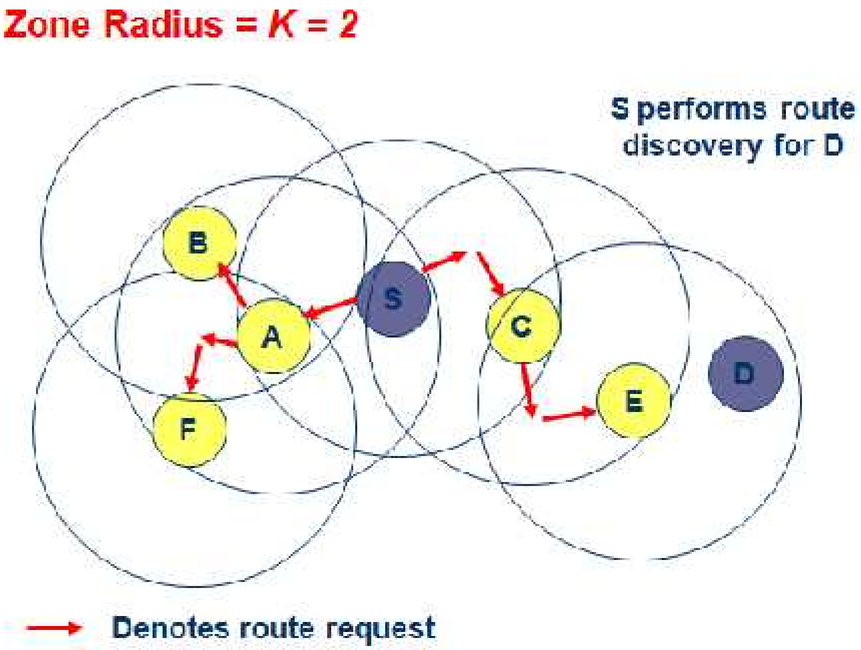


Fig 3.11: Zone to zone communication [10].

- If a node P finds that the destination D is within its routing zone, P can initiate a route reply.
- Each node appends its address to the RREQ message during the route request phase. This is similar to route request phase in DSR.
- This accumulated address can be used to send the route reply (RREP) back to the source node S.
- An alternative strategy is to keep forward and backward links at every node´s routing table similar to the AODV protocol. This helps in keeping the packet size constant.
- A RREQ usually results in more than one RREP and ZRP keeps track of more than one path between S and D. An alternative path is chosen in case one path is broken.
- When there is a broken link along an active path between S and D, a local path repair procedure is initiated.
- A broken link is always within the routing zone of some node.
- Hence, repairing a broken link requires establishing a new path between two nodes within a routing zone.
- The repair is done by the starting node of the link (node A in the previous diagram) by sending a route repair message to node B within its routing zone.
- This is like a RREQ message from A with B as the destination.
- When a node P receives a RREQ message, P records the message in its list of RREQ messages that it has received.
- If P receives the same RREQ more than once, it does not forward the RREQ the second time onwards.
- Also P can keep track of passing RREQ messages in several different ways.

3.3 Position based routing:

Position based routing consists of class of routing algorithm. They share the property of using geographic positioning information in order to select the next forwarding hops. Position based routing is broadly divided into two types. One position based greedy V2V protocol Another Delay tolerant protocol.

3.3.1 NON-DTN

3.3.1.1 Beacon

a) NON-Overlay

i) GPSR: Greedy Perimeter Stateless Routing [12, 13, 14]

The GPSR [12] algorithm belongs to the category of position based routing, where an intermediate node forwards a packet to an immediate neighbor which is geographically closer to the destination node. This approach is called greedy forwarding. For that matter, each node needs to be aware of its own position, the position of its neighbors as well as the position of the destination node. Each node is able to obtain its own position using positioning devices, exchange it with neighboring nodes by beacon messages and obtain the position of the destination node by a separate location service.

Characteristics [14]

- Greedy Perimeter Stateless Routing (GPSR) proposes the aggressive use of geography to achieve scalability.
- In GPSR, the packets follow the perimeter of the planar graph to find their routes.
- Although the GPSR approach reduces the number of states a node should keep, it has been designed for general mobile ad hoc networks and requires a location service to map locations and node identifiers.
- The algorithm consists of two methods:
    - greedy forwarding + perimeter forwarding
        - Greedy forwarding, which is used wherever possible.
        - Perimeter forwarding is used in the regions where greedy forwarding cannot be done.
- Under GPSR, packets are marked by their originator with their destinations' locations. As a result, a forwarding node can make a locally optimal, greedy choice in choosing a packet's next hop.
- Specifically, if a node knows its radio neighbors' positions, the locally optimal choice of next hop is the neighbor geographically closest to the packet's destination.
- Forwarding in this scheme follows successively closer geographic hops, until the destination is reached.

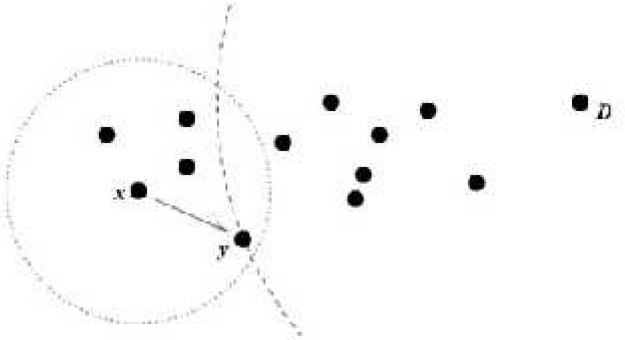


Fig 3.12 GPSR routing [14]

Routing [14]:

- A simple beaconing algorithm provides all nodes with their neighbors' positions: periodically, each node transmits a beacon to broadcast MAC address, containing its own identifier (e.g., IP address) and position.
- Position is encoded as two four-byte floating point quantities, for *x* and *y* coordinate values
- Upon not receiving a beacon from a neighbor for longer than timeout interval *T*, a GPSR router assumes that the neighbor has failed or gone out-of-range, and deletes the neighbor from its neighbor table.

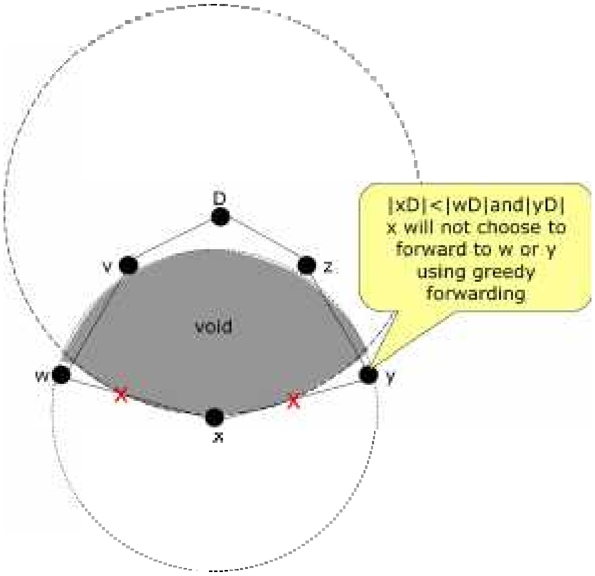
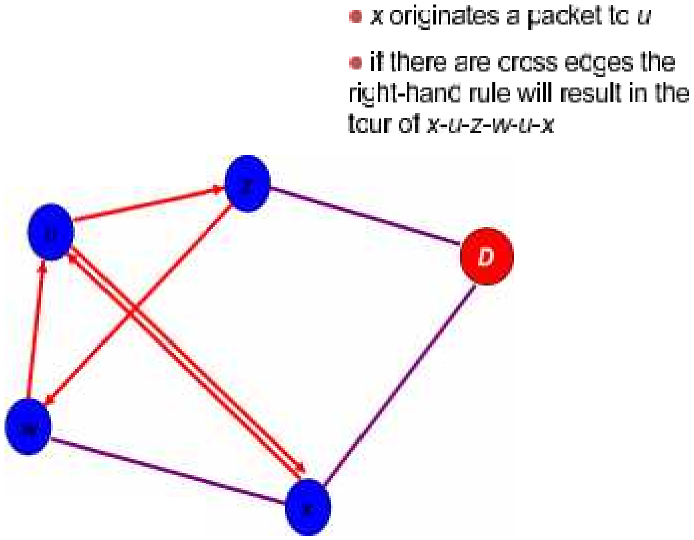

D
v
z
void
w
y
x
|xD|<|wD|and|yD| x will not choose to forward to w or y using greedy forwarding

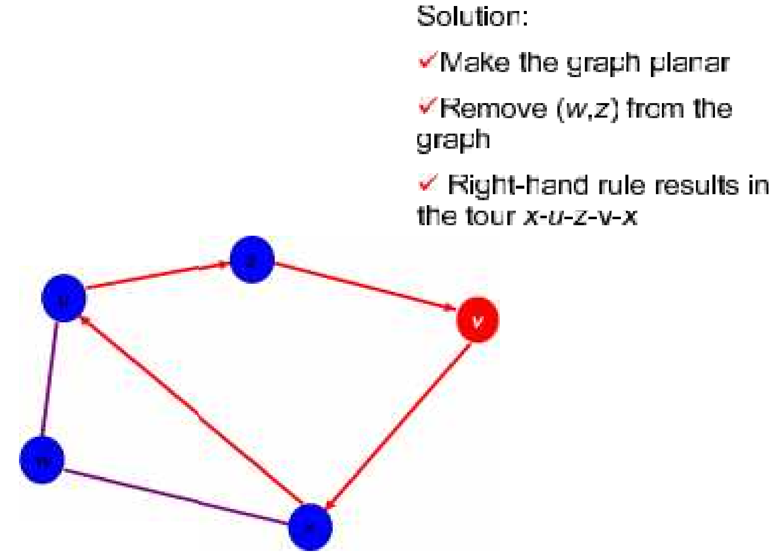
Fig 3.13: Greedy approach [14]

- GPSR recovers from a local maximum using a Perimeter mode, where the right-hand rule is used. The state accumulated in these packets is cached by nodes, which recover from local maxima in greedy forwarding by routing to a node on a cached perimeter closer to the destination.
- This approach requires a heuristic, the no-crossing heuristic, to force the right-hand rule to find perimeters that enclose voids in regions where edges of the graph cross.



- *x* originates a packet to *u*
- if there are cross edges the right-hand rule will result in the tour of *x-u-z-w-u-x*

Fig 3.14: Crossing over problem [14]



Solution:

✓Make the graph planar

✓Remove (*w*,*z*) from the graph

✓ Right-hand rule results in the tour *x-u-z-v-x*

Fig 3.15: Crossing over solution [14]

Making a planner graph [14]

- A graph in which no two edges cross is known as planar. A set of nodes with radios, where all radios have identical, circular radio range *r*, can be seen as a graph: each node is a vertex, and edge (*n*, *m*) exists between nodes *n* and *m* if the distance between *n* and *m*, $d(n, m) \leqq r$.
- Convert a connectivity graph to planar non-crossing graph by removing “bad” edges
- Ensure the original graph will not be disconnected
- Two types of planar graphs:
    - Relative Neighborhood Graph (RNG)
    - Gabriel Graph (GG)

ii) PBR-DV: Position-Based Routing with Distance Vector [15]

Position-Based Routing with Distance Vector Recovery (PBR-DV) [15] uses AODV-Style recovery as packets fall into a local maximum. The node at the local maximum would broadcast a request packet for the node’s position and destination’s location.

After receiving the request packet, it will check the node and determine which one is closer to the destination or closer to local maximum?

If it is not closer to destination it records the node from which it receives the request packet and then broadcasts the request. In other respect, it sends a reply to the node from which it receives the request.

Similarly when the reply packet travels back to the local max node, each intermediate node will record the previous node which receives the reply packet.

So the local max node can maintain a route to a closer node than itself. The disadvantage of this method is its extra flooding which is necessary to discover the non-greedy part of the route.

There is no reappraisal for comparing PRB-DV to neither GPSR nor AODV; therefore the execution in packet delivery and from above is unconvincing.

iii) Grant: Greedy Routing with Abstract Neighbor Table [15]

GRANT [15] uses the idea of extended greedy routing which each node known its “x” hop local area. Each node is far view envision of the best route to avoid local maximum.

For selecting the next forwarding neighbor “E” which is based on the multiplication of distance between the nodes, hop “x” in another side forms “E” and the destination, the shortest path from “N” to “E”, and the Charge per hop for multi-hop neighbors. The smallest metric is chosen to the next hop by neighbor “E”. GRANT sets apart the plane into areas and involved a typical neighbor for every area, because transmitting x-hop neighbors in the beacon is more than enough from above.

Upon receiving a beacon, a node computes the area that the broadcasting node and its neighbors belong to, thus categorizing them into different hops from the current node. On receiving a beacon, a node calculate the area which includes the broadcasting node and its neighbors, therefore classification of them is in the different hopes from and the current node. The estimate is based on snapshots of appointment of cars from a uniform distribution.

The spreading model is based on a significant property of a city script that many radio obstructions are there, for example buildings and trees. A simple supposition that model makes nodes on different streets unable to hear each other because of radio obstructions. Consequence gives us that most of the routes in GRANT have shorter path length than traditional greedy routing. And also, the number of times the packet is recovered per route is also less in GRANT than in traditional greedy routing.

GRANT with Face routing similarly the recovery strategy is compared to GRANT with Distance Vector which is based on the recovery (like PRG-DV).The total of hops for each recovery is way less in GRANT with Distance Vector-which is based on recovery than GRANT with Face routing.

In spite of the problematic condition of short-range overwhelm, Distance Vector recovery is strong to radio obstruction that epidemic Face routing. Because the estimate is done base on the static traces and the x-hop neighbors are supposed to be available, the beacon above and possible erroneousness are not measured and well comprehended. Also, additional ways are which have smaller path length than traditional greedy routing on a normalized percentage basis; there is no performance metric such as packet delivery ratio that can confirm its correct performance.

b) Overlay

i) GPCR: Greedy Perimeter Coordinator Routing Algorithm [12, 15]

GPCR [15] is an enhancement of the GPSR protocol. It is also based on the fact that streets and junctions naturally form a planar graph and thus does not require any planarization algorithm. Moreover, GPCR does not need an urban map. As GPSR, it contains two phases:

A restricted greedy forwarding and a recovery phase. The restricted greedy for-warding part only uses nodes on the same road segment as potential relays, as building and other obstacles block radio signal between adjacent road segments.

An important point is that, since junctions are the only places where routing decisions are made, packet must always be sent to a node that is at a junction. Forwarding a packet across any junction may risk bringing GPCR to a local maximum.

At junctions, a greedy decision is also made, and the neighboring. An edge is here defined as a wireless bi-directional link between two neighbor nodes. A virtual edge is a geometric construction connecting two nodes that are not in connection range node which brings the maximum progress towards the destination is chosen. If a local maximum is reached, the recovery mode is used.

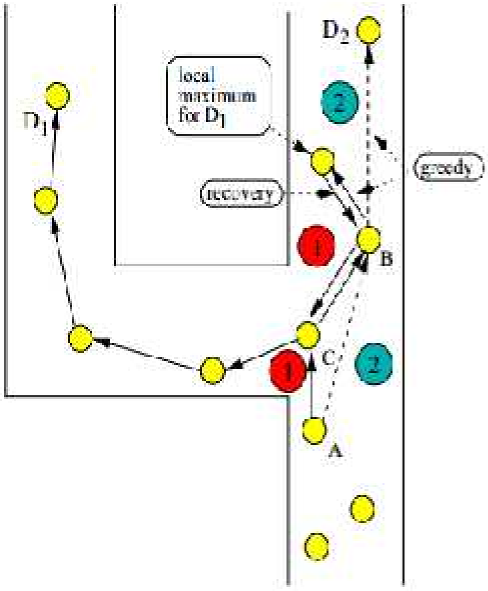


Fig 3.16: Routing with GPCR and the critical junction node. [12]

When GPCR is in recovery mode, packets are backtracked in a greedy fashion (i.e. bringing maximum progress) to a junction node in order to find an alternate solution to return to the greedy mode. At the junction node, the right-hand rule is used to find the next road segment to forward the packets.

The major weak points of GPCR are threefold [12]:

- First, junction nodes need to be determined and advertised which might bring some overhead to the protocol.
- Second, recognizing a junction node, which is faulty in GPCR, is extremely crucial to avoid local maximums and consequent hop reduction.
- Third, even if the junction node detection algorithm is perfect, forwarding to a node at a junction is often unnecessary and counter-productive as most of junctions are not critical.

As shown in Figure where the destination node is D1 and the routing starts from A, if a packet faces a critical junction and that the protocol fails to provide a valid junction node, GPCR will surely encounter a local maximum. Moreover, when a packet does not face a critical junction (the destination node is D2 in Figure), routing to a junction node is counter-productive as crossing the junction to the relay bringing the maximum distance would have been preferred. In that perspective, it would be better if the observation of a critical junction be made by nodes before the junction.

ii) **GpsrJ+ [12]**

GpsrJ+ [12] removes the unnecessary stop at a junction while keeping the efficient planarity of topological maps. It uses two-hop neighbor beaconing to predict which road segment its neighboring junction node will take. If the prediction indicates that its neighboring junction will forward the packet onto a road with a different direction, it forwards to the junction node; otherwise, it bypasses the junction and forwards the packet to its furthest neighboring node.

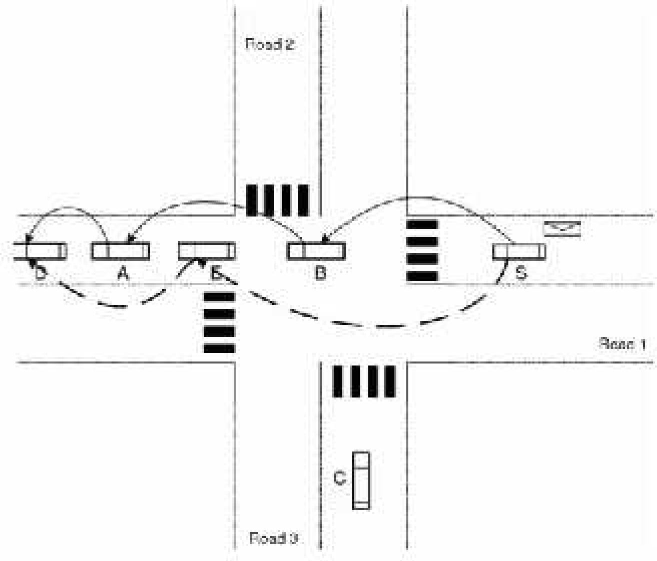


Fig 3.17: Dashed arrows are GpsrJ+ and solid arrows are GPCR [12]

The figure shows that GpsrJ+ can bypass the junction area and forward the packet to node. E directly, yet GPCR forwards it to the junction node B, thus causing more transmissions. In the perimeter mode, GpsrJ+ uses the right-hand rule to determine the best direction (as opposed to final destination direction) and thereby the best forwarding node. That is, if the furthest node is in the same direction as the best direction, the best forwarding node is the furthest node; otherwise, the best forwarding node is a junction node. GpsrJ+ manages to increase packet delivery ratio of GPCR and reduces the number of hops in the recovery mode by 200% compared to GPSR.

iii) CAR: Connection-Aware Routing [15]

CAR [15] utilizes AODV which is based on path discovery to encounter routes with restricted broadcast from Preferred Group Broadcasting (PGB). In any event, nodes which is formed the route don‘t record their previous node from backward aware and their premature node that forwards the path reply packet from the destination. Somewhat, anchor points, which are nodes in the vicinity a crossing or road curve, are recorded in the path discovery packet.

A node decides itself similarly an anchor point if its vector is not parallel to the velocity vector of the preceding node in the packet. The destination allows accepting many times path discovery packets. It selects the route which is provides better connection and lower delays. Advanced

Greedy Forwarding (AGF) is utilized to the front the route reply back to the source via the recorded anchor points. When the source receives the route respond, it records the route to the destination and beginning to transmit.

Data packets which forwarded in a greedy manner for the destination inward the set of anchor points utilize the AGF. Additionally to manage mobility by AGF, CAR insert "guards" to aid to tack the prevalent condition of a destination. A keeping node can be filtered or directed again packets or summed information to a packet that delivers finally this information to the destination of packet. In result illustrates CAR possesses important packet delivery ratio (PDR) than GPSR and GPSR+AGF.

The reason that CAR's PDR is higher than GPSR+AGF is that CAR guarantees to recognize the shortest connected route when GPSR+AGF permit from sub-optimality of greedy mode in terms of determining like a route. CAR's route discovery above which is checked by PGB. The above of supply protection is not in the data packets except in the beacons.

iv) GSR: Geographic Source Routing [17]

Geographic Source Routing [17] relies on the availability of a map and computes a Dijkstra shortest path on the overlaid graph where the vertices are junction nodes and the edges are streets that connect those vertices. The sequence of junctions establishes the route to the destination. Packets are then forwarded greedily between junctions. GSR does not consider the connectivity between two junctions; therefore, the route might not be connected through. Recovery when such a case happens is greedy forwarding. The major difference between GSR and CAR is that CAR does not use a map and it uses proactive discovery of anchor points that indicate a turn at a junction.

The evaluation also considers a basic form of obstacle modeling as the propagation model. GSR performs better than AODV and DSR in packet delivery ratio. In a densely populated network, most roads are connected that GSR forwards most of the packets. Scalability is not a problem to GSR as to AODV and DSR. However, GSR is not compared with other position-based routing protocols. Its performance in sparse networks is not verified.

v) A-STAR: Anchor-Based Street and Traffic Aware Routing [15]

Anchor-Based Street and Traffic Aware Routing (A-STAR) [15] is similar to GSR in that packets are routed through anchor points of the overlay. However, A-STAR is traffic aware: the traffic on the road determines whether the anchor points of the road will be considered in the shortest path.

A-STAR routes based on two kinds of overlaid maps: a statically rated map and a dynamically rated map. A statistically rated map is a graph that displays bus routes that typically imply stable amount of traffic. Dijkstra paths computed over the statistically rated map are in general connected because of the extra knowledge.

A dynamically rated map is a map that is generated based on the real-time traffic condition on the roads. Road-side deployment units can monitor the city traffic condition and distribute this information to every vehicle. Thus, the difference between a statically rated map and a dynamically rated map is accuracy of road traffic; while a statically rated map is based on bus routes that typically have high traffic volume; a dynamically rated map is based on the traffic monitored dynamically by road-side units.

A-STAR also proposes a different recovery algorithm when the packet gets stuck due to dis-connectivity of the current path to the destination. The node will re-compute a new anchor path and the road segment where the packet is currently located will be marked as “out of service” temporarily to prevent other packets from entering into the same problem. The notification of “out of service” is piggybacked in the recovered packets. Nodes that receive the recovered packets update their map and recomputed anchor paths accordingly.

A-STAR also proposes a different recovery algorithm when the packet gets stuck due to dis-connectivity of the current path to the destination. The node will re-compute a new anchor path and the road segment where the packet is currently located will be marked as “out of service” temporarily to prevent other packets from entering into the same problem. The notification of “out of service” is piggybacked in the recovered packets. Nodes that receive the recovered packets update their map and recomputed anchor paths accordingly.

vi) STBR: Street Topology Based Routing [1, 4]

Street Topology Based Routing (STAR) [1, 4] went further than A-STAR by computing the road connectivity at junction nodes. One of the nodes at a junction is selected as a master that is responsible for checking if links to the next junctions are up or down. Within the broadcast from every master, there is also link information to all neighboring links. This is because every master will receive every other master's link information. Thus, every master contains a two-level junction neighbor table. The first level is through neighboring links to its direct junction nodes. The second level is its direct junction nodes through their neighboring links to their own junction nodes. In STBR, packets are routed based on their geographic distance to the street where the destination is on. This is different from GSR or A-STAR where routes are computed through Dijkstra shortest path.

vii) GyTAR: Greedy Traffic Aware Routing [15]

Greedy Traffic Aware Routing protocol (GyTAR) [15] is an overlaid approach similar to the approaches mentioned above in that packets are forwarded greedily toward the next junction which will then determine the best junction to forward next. GyTAR assumes that the number of cars is given per each road from roadside units and determines the connectivity of roads. A score is given to each neighboring junction considering the traffic density and their distance to the destination. The weights to traffic density and their distance to the destination are configurable parameters. GyTAR tries to mimic the shortest path routing by taking into account the road connectivity.

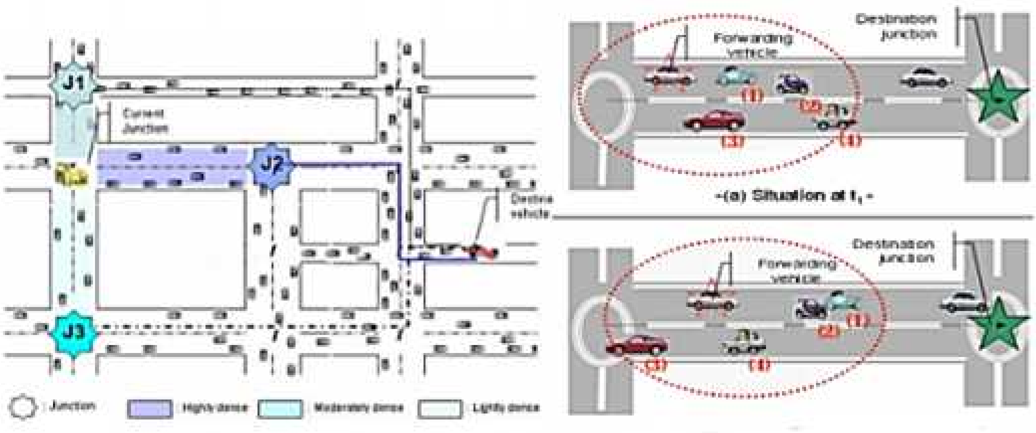


Fig 3.18: GyTAR Routing [15]

viii) LOUVRE: Landmark Overlays for Urban Vehicular Routing Environments [15]

There are two types of camp in geographic greedy overlay routing into [15]:

The first camp is geo-reactive overlay routing where the next overlaid node is determined based on their neighboring nodes' distance to the destination (STBR) or a combination of it and traffic density (GyTAR).

The second camp is geo-proactive overlay routing where the sequence of overlaid nodes is determined a-priori (GSR and A-STAR).

Landmark Overlays for Urban Vehicular Routing Environments (LOUVRE) belongs to the second camp. It takes note of the fact that above a given vehicular density threshold, an overlay link remains connected regardless of the vehicular spatiotemporal distribution on the link. Thus, by only considering overlay links based on such density threshold when establishing overlay routes, most routes would partially use the same overlay links. With these considerations, geo-proactive overlay routing becomes attractive as it guarantees global route optimality and reduces the delay for establishing overlay routes. The drawback of this approach is obviously its scalability.

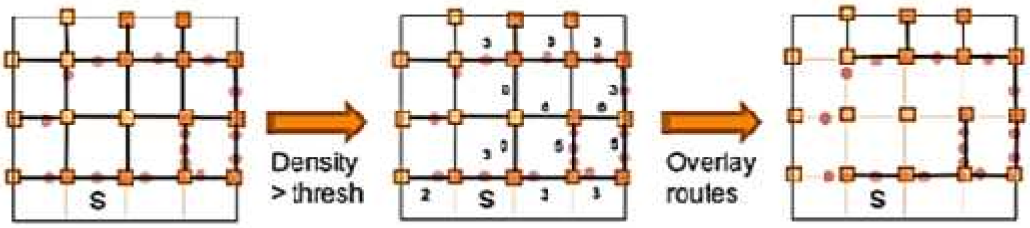


Fig 3.19: LOUVRE Routing [15]

Figure shows the procedure in which LOU-VRE obtains routes to nodes from node S. From the peer-to-peer density scheme, LOURVE first filters out roads that do not have density over the threshold, determined by the road length and radio range. Then the overlaid routes are built on top of roads whose density is above the threshold. This forms the graph the Dijkstra shortest path algorithm runs on. The algorithm automatically obtains the shortest path between sender and its destination.

The novelty of LOUVRE is that road density which correlates to road connectivity are computed in a peer-to-peer fashion to remove reliance on deployment of roadside units. Thus, each node has the density of all the “connected” roads in the network. The Dijkstra shortest path is then built by roads with density above a certain density threshold, correlating closely to road connectivity.

LOUVRE performs better than GPCR and GPSR due to LOUVRE’s global knowledge of the density distribution on road segments and on local maxima, typical information that is not available to GPSR and GPCR. The hop count and delay are also significantly reduced as LOUVRE does rarely encounter local maxima and therefore mostly does not use a recovery mode [15].

c) DIR: Diagonal Intersection-based Routing [12, 16]

To improve the CAR protocol, Chen et al developed a diagonal-intersection- based routing (DIR) [16] protocol. The key difference of CAR and DIR protocols is that DIR protocol constructs a series of diagonal intersections between the source and destination vehicles.

The DIR [12] protocol is a geographic routing protocol. Based on the geographic routing protocol, source vehicle geographically forwards the data packet toward the first diagonal intersection, the second diagonal intersection, and so on, until the last diagonal intersection, and finally geographically reaches to the destination vehicle.

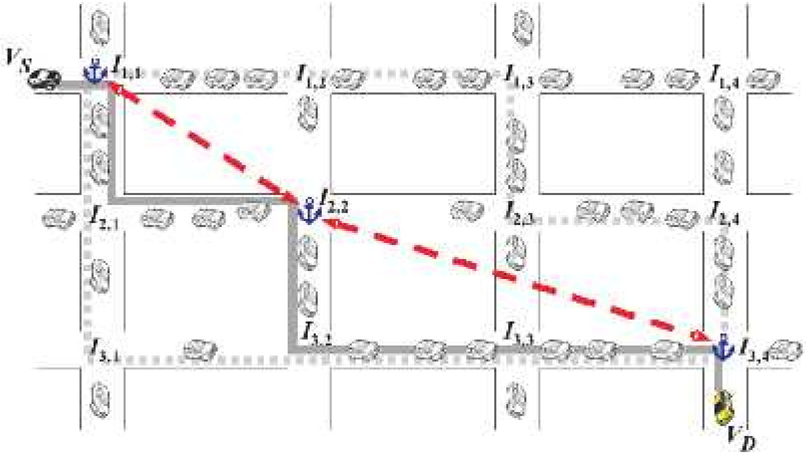


Fig 3.20: DIR routing [12]

For given a pair of neighboring diagonal intersections, two or more disjoint sub-paths exist between them. The novel property of DIR protocol is the auto-adjustability; while the auto-adjustability is achieved that one sub-path with low data packet delay, between two neighboring diagonal intersections, is dynamically selected to forward data packets. To reduce the data packet delay, the route is automatically re-routed by the selected sub-path with lowest delay. Fig shows that DIR protocol constructs a series of diagonal intersections between vehicles VS and VD. Observe that, DIR protocol may set the fewer number of anchors than CAR protocol. DIR protocol can automatically adjust routing path for keeping the lower packet delay, compared to CAR protocol

Characteristics [16]

- To improve the CAR protocol.
- DIR protocol constructs a series of *diagonal* intersections between the source and destination vehicles.
- Auto-adjustability is achieved that one sub-path with low data packet delay, between two neighboring diagonal intersections, is dynamically selected to forward data packets.
- To reduce the data packet delay, the route is automatically re-routed by the selected sub-path with lowest delay.
- DIR protocol constructs a series of diagonal intersections between vehicles V_S *and* V_D.
- DIR protocol may set the fewer number of anchors than CAR protocol.
- DIR protocol can automatically adjust routing path for keeping the lower packet delay, compared to CAR protocol.

d) AMAR: Adaptive movement aware routing [17]

The AMAR [17] protocol can be used to solve the above described problem of BMFR. This protocol makes use of additional information about vehicle movement to select an appropriate packet's next-hop that ensures the data delivery. In this scheme, a border node is selected out of the two conflicting nodes by making use of mobility awareness i.e. by using some parameters like speed and direction. Based on the position, speed and direction, weighted score W*i* for border bode *i* is calculated as follows:

$Wi = \alpha Pm + \beta Dm + \gamma Sm$

Where α, β and γ are the weight of the three used metrics Pm, Dm, Sm representing the position, the direction and the speed factors respectively with α+ β+ γ= 1.

A sorted list of next hop candidates can be defined on the computed score W*i*; the node with the highest weighted score among all the border nodes of the current forwarder will be selected as the best candidate for next forwarding node. It also improves the data delivery.

Problem in AMAR

AMAR protocol solves the problem of BMFR but there is still some problem in it. Suppose that *i* and *j* are two border nodes and W*i* and W*j* are their respective calculated weighted score. If the weighted score of two border nodes *i* and *j* i.e. W*i* and W*j* are equal, again a dilemma will occur. Now to resolve this conflict, we use an attribute named probability.

Novel Solution

On the basis our study, we assign probability to the node that changes its direction on the intersection as Pc and to the node that does not change its direction on the intersection as Pnc where Pc is higher than Pnc.

It is assumed that all the nodes have a digital map. The source node or the current forwarding node will look on the route of both the conflicting nodes. Now the current forwarding node will take into account the probability factor and discard the node having an intersection in its route since it may change its direction and leading the packet to be forwarded in the wrong direction.

Finally the packet is forwarded to the other node i.e. to the node that does not have an intersection in it route in order to accomplish successful delivery to the destination.

e) EBGR: Edge node based greedy routing protocol [18, 19]

EBGR [18] is a reliable greedy position base routing algorithm designed for sending messages from any node to any other node (unicast) or from one node to all other nodes (broadcast/multicast) in a vehicular ad hoc network. The general design goals of the EBGR algorithm are to optimize the packet behavior for ad hoc networks with high mobility and to deliver messages with high reliability.

The EBGR algorithm has six basic functional units:

- Neighbor Node Identification (NNI),
- Distance Calculation (DC),
- Direction of Motion Identification (DMI),
- Reckoning Link Stability (RLS),
- Potential score calculation (PS)
- Edge Node Selection (ENS).

Neighbor Node Identification (NNI) [19]

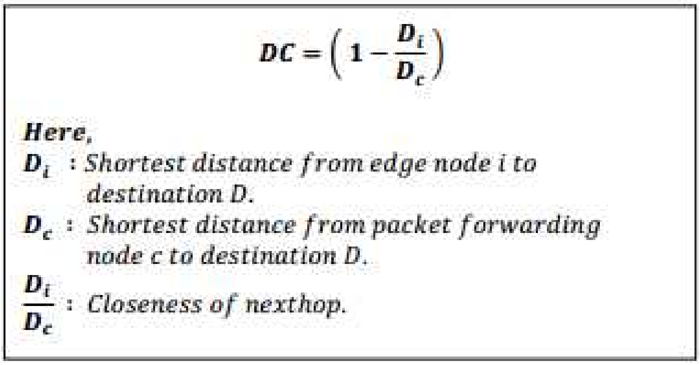
Neighbor node identification is the process whereby a vehicle/node identifies its current neighbors within its transmission range. For a particular vehicle, any other vehicle that is within its radio transmission range is called a neighbor. All vehicles consist of neighbor set which holds details of its neighbor vehicles. Since all nodes might be moving, the neighbors for a particular mobile node are always changing. The neighbor set is dynamic and needs to be updated frequently.

Generally, neighbor node identification is realized by using periodic beacon messages. The beacon message consists of node ID, node location and timestamp. Each node informs other nodes of its existence by sending out beacon message periodically. All nodes within the transmission range of source/packet forwarding node will intimate its presence by sending a beacon message every µ second. After the reception of a beacon, each node will update its neighbor set table.

If a node position is changed, then it will update its position of all neighbors by sending beacon signal. If a known neighbor, times out after α*µ seconds without having received a beacon (α is the number of beacons that a node is allowed to miss) and it will be removed from the neighbor set table.

Distance calculation (DC) [19]

The location and distance information of all vehicles/nodes can be identified with the help of GPS receivers. It can be communicated to neighbor vehicles using periodic beacon messages. The neighbor node which is closer to the destination node is calculated. The closeness of next hop is identified by the mathematical model and it is shown in Fig.



$$DC = \left(1 - \frac{D_i}{D_c}\right)$$

Here,

D_i : *Shortest distance from edge node i to destination D.*

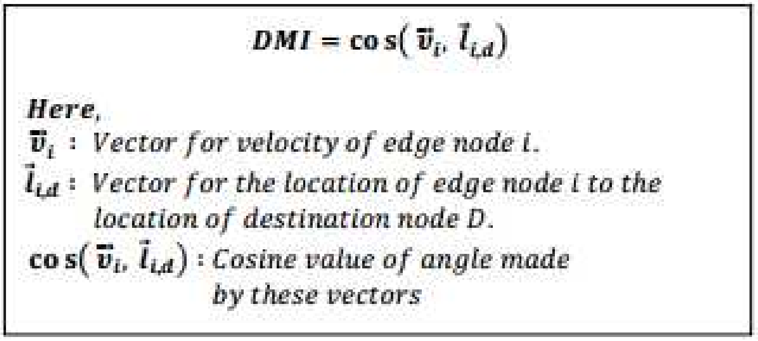
D_c : *Shortest distance from packet forwarding node c to destination D.*

$\frac{D_i}{D_c}$: *Closeness of nexthop.*

Fig 3.21: Distance Calculation in EBGR [19]

Direction of Motion Identification (DMI) [19]

The appropriate neighbor node which is moving towards the direction of destination node is identified using the mathematical model and it is shown in Fig.



$$DMI = \cos(\vec{v}_i, \vec{l}_{i,d})$$

Here,

$\vec{v}_i$: *Vector for velocity of edge node i.*

$\vec{l}_{i,d}$: *Vector for the location of edge node i to the location of destination node D.*

$\cos(\vec{v}_i, \vec{l}_{i,d})$: *Cosine value of angle made by these vectors*

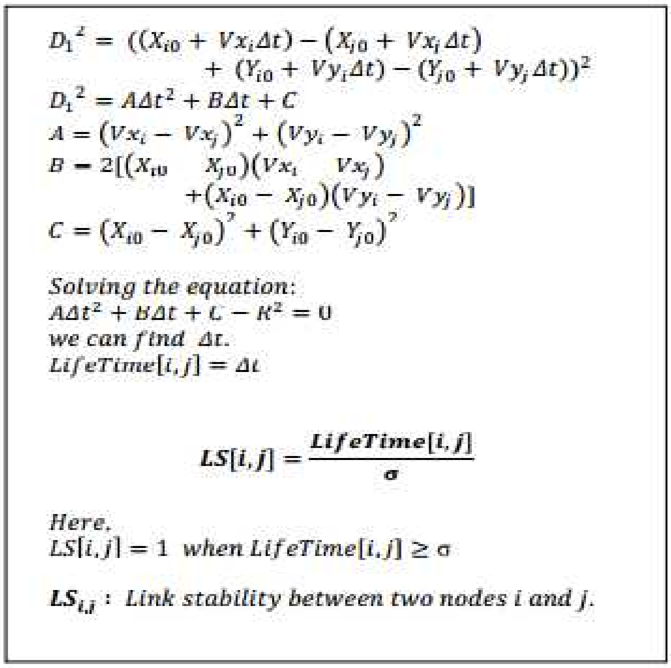
Fig 3.22: Direction of Movement Identification in EBGR [19]

The cosine value of vector for velocity of edge node i and vector for location of edge node i to the location of destination node D is measured. A large cosine value implies a vehicle/node can still approach the destination closer and closer along its current direction.

Reckoning Link Stability (RLS) [19]

Each vehicle estimates the Link Stability (LS) for each neighboring vehicle before selecting the next hop for the data forwarding/sending. The LS is a relation between the link communication lifetime and a constant value which represents in general cases the routing route validity time, and it depends on the used routing protocol. Figure shows how link lifetimes are estimated based on neighbors' movement information.

The lifetime of the link (i, j) lifetime [i, j] corresponds to the estimated time $\Delta t = t_1 - t_0$ with t_1 is the time when D_1 becomes equal or bigger than the communication range R, D_1 and Δt are estimated using the initial positions of i and j.



$$D_1^2 = ((X_{i0} + Vx_i\Delta t) - (X_{j0} + Vx_j\Delta t) + (Y_{i0} + Vy_i\Delta t) - (Y_{j0} + Vy_j\Delta t))^2$$

$$D_1^2 = A\Delta t^2 + B\Delta t + C$$

$$A = (Vx_i - Vx_j)^2 + (Vy_i - Vy_j)^2$$

$$B = 2[(X_{i0} - X_{j0})(Vx_i - Vx_j) + (X_{i0} - X_{j0})(Vy_i - Vy_j)]$$

$$C = (X_{i0} - X_{j0})^2 + (Y_{i0} - Y_{j0})^2$$

Solving the equation:

$$A\Delta t^2 + B\Delta t + C - R^2 = 0$$

we can find Δt.

$$LifeTime[i,j] = \Delta t$$

$$\boldsymbol{LS[i,j]} = \frac{\boldsymbol{LifeTime[i,j]}}{\boldsymbol{\sigma}}$$

Here,

$LS[i,j] = 1$ *when* $LifeTime[i,j] \geq \sigma$

$\boldsymbol{LS_{i,j}}$: *Link stability between two nodes i and j.*

Fig 3.23: Reckoning Link Stability in EBGR [19]

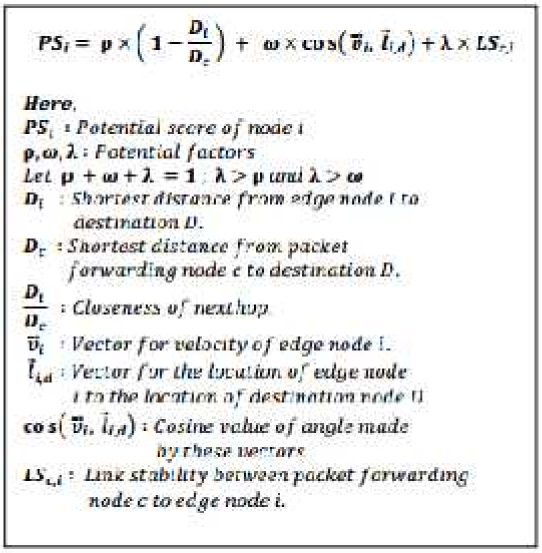
Once LS is calculated for each neighboring vehicle, EBGR selects the node corresponding to the highest LS as next hop for data forwarding. This approach should help as well in minimizing the risk of broken links and in reducing packet loss.

Potential Score Calculation (PS) [19]

The potential score (PS) of all nodes present within the different levels of transmission range of source/packet forwarding node is calculated. The potential score (PS) is calculated to identify the closeness of next hop to destination, direction of motion of nodes and reliability of neighbor nodes.

The appropriate edge node with largest potential score will be considered as having higher potential to reach the destination node and that particular node can be chosen as next hop to forward the packet to the destination node. Potential score is calculated by addition of DC, DMI and LS and that mathematical model represented in Figure.

$$PS_i = \rho \times \mathrm{DC} + \omega \times \mathrm{DMI} + \lambda \times \mathrm{LS}$$



$$PS_i = \rho \times \left(1 - \frac{D_i}{D_c}\right) + \omega \times \cos(\vec{v}_i, \vec{l}_{i,d}) + \lambda \times LS_{c,i}$$

Here,

PS_i : *Potential score of node i*

ρ, ω, λ : *Potential factors*

Let $\rho + \omega + \lambda = 1$; $\lambda > \rho$ *and* $\lambda > \omega$

D_i : *Shortest distance from edge node i to destination D.*

D_c : *Shortest distance from packet forwarding node c to destination D.*

$\frac{D_i}{D_c}$: *Closeness of next hop*

$\vec{v}_i$: *Vector for velocity of edge node i.*

$\vec{l}_{i,d}$: *Vector for the location of edge node i to the location of destination node D*

$\cos(\vec{v}_i, \vec{l}_{i,d})$: *Cosine value of angle made by these vectors*

$LS_{c,i}$: *Link stability between packet forwarding node c to edge node i.*

Fig 3.24: Potential Score Calculation in EBGR [19]

Edge Node Selection (ENS) [19]

In the Edge Node Selection, edge nodes are selected for packet forwarding event. An edge node is a node which has shortest distance to the destination D compared to all other nodes within the different levels of transmission range of source/packet forwarding node.

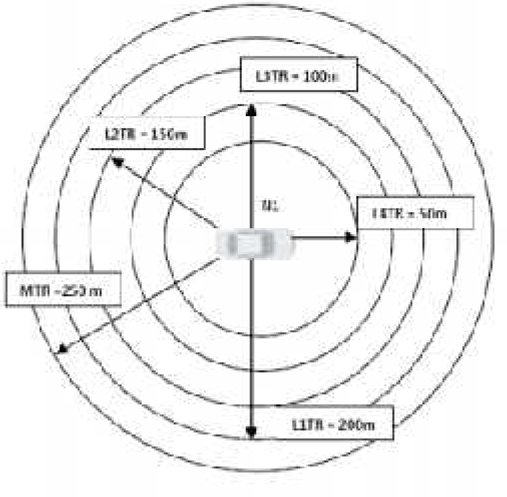


Fig 3.25: The different levels of transmission range [19]

The different levels of transmission range are considered to avoid packet loss due to high speed mobility of vehicles. An edge node has the responsibility of saving received data packets in forwarding table and transfers it later when those nodes meet new neighbors. The overall objective of the algorithm is to forward the packet as soon as possible to increase packet delivery ratio, minimize the end to end delay and avoid packet loss. The MTR of a vehicle/node is 250m.The other levels of transmission range is considerably less than MTR. The different levels of transmission range are shown in Fig.

f) Abr: Associativity Based Routing [20, 21]

Associativity Based Routing (ABR) [21] is a bandwidth efficiently distributed routing protocol used in Ad Hoc networks. ABR is a source initiated On-Demand routing protocol. ABR uses both point-to-point and broadcast routing. In ABR, the destination node takes the decision of choosing a route basing on the property of "Associativity", the selected route is used and all other routes are discarded. This results in long-lived routes because the decision is made on the property of "Associativity". ABR consists of three phases:

1. Route Discovery phase
2. Route Re-Construction (RRC) phase.
3. Route deletion phase

1) **The route discovery phase in ABR:** The route discovery phase in ABR is accomplished by a Broadcast Query (BQ) and a wait-reply (BQ-REPLY) cycle. A node desiring a route broadcasts a BQ message in search of mobiles that have a route to the destination. All nodes receiving the query (that are not the destination) append their addresses and their associativity ticks with their neighbors along with QoS information to the query packet. A successor node erases its upstream node neighbors' associativity tick entries and retains only the entry concerned with itself and its upstream node. In this way, each consequential packet arriving at the destination will contain the associativity ticks of the nodes along the route to the destination. The destination is then able to select the best route by examining the associativity ticks along each of the paths. In the case where multiple paths have the same overall degree of association stability, the route with the minimum number of hops is selected. The destination then sends a REPLY packet back to the source along this path. Nodes propagating the REPLY mark their routes as valid. All other routes remain inactive and the possibility of duplicate packets arriving at the destination is avoided.

2) **The route re-construction phase in ABR:** Movement by the source results in a new BQ-REPLY process. The RN message is a Route Notification that is used to erase the route entries associated with downstream nodes. When the destination node moves, the immediate upstream node erases its route and determines if the node is still reachable by a Localized Query (LQ [H]) process, where H refers to the hop count from the upstream node to the destination. If the destination receives the LQ packet, it replies with the best partial route; otherwise, the initiating node times out and the process backtracks to the next upstream node. Here an RN message is sent to the next upstream node to erase the invalid routes and inform this node it should invoke the LQ [H] process. If this process results in backtracking more than halfway to the source, the LQ process is discontinued and a new BQ process is initiated at the source.

3) **Route deletion phase in ABR:** When a discovered route is no longer desired, the source node initiates a Route Delete (RD) broadcast so that all nodes along the route update their routing tables. The RD message is propagated by a full broadcast, as opposed to a directed broadcast, because the source node may not be aware of any route node changes that have occurred during route re-constructions.

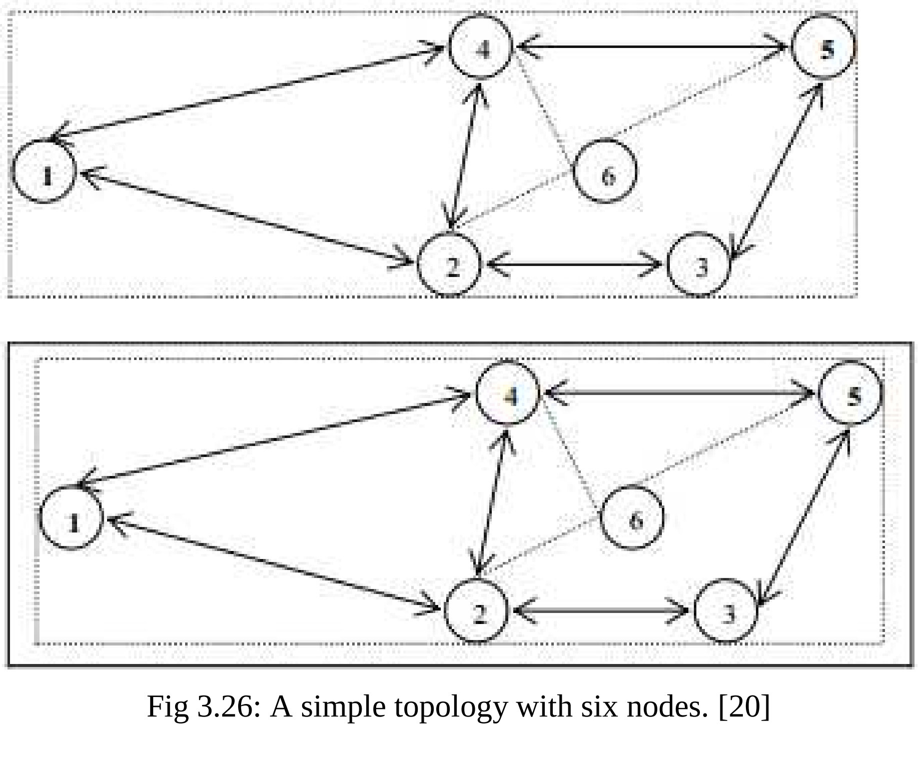


Fig 3.26: A simple topology with six nodes. [20]

Description of the Protocol [20]

- A double arrow indicates that the two nodes are in high association with each other. It means both nodes can send messages.
- Dotted line indicates low association between nodes. It means that there is no communication between these two nodes due to low associativity ticks.
- Source node 1 wants to establish a route to destination node 5. Node 1 broadcast a BQ request which is received by node 2 and 4.
- While receiving the BQ request, node 2 and node 4 will check that is it desired destination node or not. Now these nodes will create an entry in their routing table specifying a route back to the originator node 1. These nodes also save the route quality information in BQ request with hop count value.
- Since node 2 and node 4 are not the target of BQ request. So both nodes will broadcast this BQ request to their neighboring nodes. Node 4 broadcast to node 2, 1 and 5. Node 1 and node 2 will drop this BQ request because both have processed this same BQ request earlier. Only node 5 checks that it is destination node desired by source node and will create the routing table entry.
- Node 5 is the destination node, but it waits before creating the REPLY message because it will choose best path.
- Now node 2 broadcast to nodes 3, 4 and 1. Node 4 and node 1 will drop this request from node 2. Only node 3 will create the entries in its routing table and check that it is destination node or not.
- Again it will broadcast this request to its neighboring node. Only node 5 will receive this request. Node 5 enters its routing entry in its table and check that is desired destination node.
- Now node 5 chooses best path among them. It should be noted that Node 6 will not receive any broadcast messages because it does not have association with the broadcasting nodes. It will create the REPLY message to source node by using the 5-4-1 path because it is having shortest path and hightest associative ticks.
- However another path is available 5-3-2-1. But this path having longer hop count so it will be discarded. Now REPLY message unicast to path 5-4-1.

g) MORA: Movement-based routing algorithm [22, 23]

Movement-based routing algorithm [22] is completely distributed, since nodes need to communicate only with direct neighbors within their transmission range, and it exploits a specific metric, which exploits not only the position, but also the direction of movement of nodes.

In a position-based routing algorithm, each node makes a decision to which neighbor to forward the message based only on the location of itself, its neighboring nodes, and the intended destination. Therefore, the system can be decentralized, more robust and easier to set up and operate. In our approach, considering the specific automotive scenario, this decision is taken considering also which direction neighbors are moving in.

The metric used in MORA is a linear combination of the number of hops and a target functional, which can be independently calculated by each node.

The core idea of the approach is to develop a function which depends on the distance of forwarding car from the line connecting the source and the destination and on the node's movement direction. This functional is required to be implemented in a distributed way allowing any vehicle to calculate it.

The target functional should reach its absolute maxima in the case the node is moving towards destination and it should decrease as the distance from destination increases.

Let d_0 be a reference distance metric [23], chosen on the basis of the application context. Let $x = d/d_0$ be the dimensional distance of the current node from destination and $y = l/d_0$ the dimensional distance from the destination of the intersection point between destination and its perpendicular starting from the node's current position. The functional F is a function of $x \in [0, \infty]$ and $\alpha \in [\pi, -\pi]$ where α represents an angle between the line of the movement direction and the perpendicular line to destination.

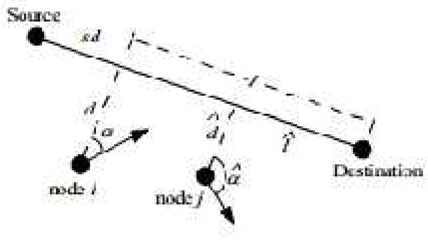


Fig 3.27: Graphical representation of parameters used in MORA. [23]

In order to ensure the targeted properties, we choose the functional F as follows:

$$F_{\delta,\gamma}(x,\alpha) = \sin\frac{|\alpha|}{3}e^{-|x|} + \cos\frac{\alpha}{3}e^{-\frac{(x-\delta)^2}{\gamma}}$$

Where δ and γ are two parameters set on the basis of the application, which simply vary the curvature of F, adjusting the weight associated with node's movement direction, δ defines the value of x corresponding to the relative maximum along the x axis and γ leads to a smoother or steeper behavior down to zero.

3.3.1.2 Non Beacon

Contention-Based Forwarding: SSFR: Spray Select Focus Routing [24]

Routing the packets efficiently in mobile ad hoc network does not have end to end paths. Multiple copies are forwarded from the source to the destination. To deal with such networks, researches introduced flooding based routing schemes which leads to high probability of delivery. But the flooding based routing schemes suffered with contention and large delays. In Spray Select Focus Routing [24], sprays a few message copies into the network, neighbors receives a copy and by that relay nodes we are choosing the shortest route and then route that copy towards the destination.

In this algorithm multiple copies sprayed to the neighbors, from those neighbors a route is searched. If the copy is reached other copies are discarded. Otherwise we see the other copies for transmission. So we are minimizing the route to avoid contention in the network. Our routing algorithm has three phases:

Spray: For every message originating at a source node, L message copies are initially spread-forwarded by the source and possibly other nodes receiving a copy-to L distinct relays.

Select: Selects a node; from that node find the shortest route by hop distances to the destination.

Focus: Let U_X (Y) denote the utility of node X for destination Y; a node A carrying a copy for destination D, forwards its copy to a new node B it encounters, if and only if U_B (D) > U_A (D).

Dead end occurs when the node gets struck with hardware failure or power failure. So no packets can be transmitted through the dead end. We cannot pass through the dead end and the copy on that route gets struck.

Routing

First the dynamic nodes hop distance, is found out by the node coverage.

Then the source and the destination are identified. By this all the moving nodes between the source node and the destination node are marked.

Multiple copies are sprayed into the network by spray select focus routing algorithms. A dead end is overcome by a Bypass recovery within the coverage.

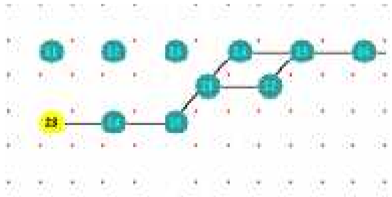


Fig 3.28: Bypass Recovery [24]

In case of without coverage the dead end is overcome by the Focus phase.

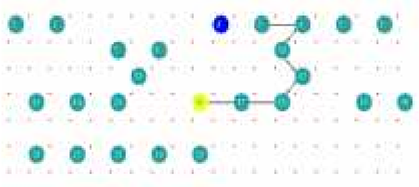


Fig 3.29: Spray Select Focus without dead ends. [24]

3.3.1.3 Hybrid

TO-GO: Topology-assist Geo-Opportunistic [26]

Topology-assist Geo-Opportunistic [26] Routing is a geographic rout-ing protocol that exploits topology knowledge acquired via 2-hop beaconing to select the best target forwarder and incorporates opportunistic forwarding with the best chance to reach it. It is different from CBF in three main aspects. First, rather than picking the next forwarding node that makes the best progress to the destination; it picks the next forwarding node that makes the best progress to a target node. A target node is defined to be the node that greedy algorithm or recovery algorithm would normally pick except at the junction where optimization in choosing the target node either beyond the junction or at the junction is based upon whether the routing is in greedy mode or recovery mode. The reason for choosing the target node instead of the destination as the frame of reference is to take care of the city topology where roads intersect and destination usually does not lie on the same street as the source as in the highway. Packets have to make multiple turns into different streets before arriving at the destination. The data is then broadcast to all direct neighbors. Whoever's distance is closer to the target node gets picked to be the next forwarding node.

The second difference is that unlike CBF, there is still the need of beacons, which are used for nodes to pick the target node. The fact that the data is broadcast and only the node that makes the furthest progress toward the target is chosen is to account for wireless channel errors and low packet delivery rate arising from multi-path fading, shadowing, and mobility – the furthest node (the target node) usually does not receive the data packet. Packets are therefore "opportunistically" making their best progress toward the target node and thus the destination. TO-GO uses a novel way to choose the forwarding set of nodes that are candidates for the next forwarding node. The set is chosen so that all nodes can hear one another (no hidden terminals) and make a progress toward the target node.

Lastly, TO-GO differs from CBF by providing routing decision for recovery. CBF on the highway works because the destination is always straight ahead. Thus, local maximum never occurs on the highway. Thus, the selection of the next forwarding node is always one that's closest to the destination. However, in city environments, streets cross each other and destination does not lie on the same street as the source. Thus, local maximum frequently occurs. TO-GO adapts the concept of CBF that packets are opportunistically sent to the target node, calculated by the routing decision in both the greedy and recovery mode.

3.3.2 DTN

a) VADD: Vehicle-Assisted Data Delivery [27]

Vehicle-Assisted Data Delivery [27] is a vehicular routing strategy aimed at improving routing in disconnected vehicular networks by the idea of carry-and-forward based on the use of predictable vehicle mobility. A vehicle makes a decision at a junction and selects the next forwarding path with the smallest packet delivery delay. A path is simply a branched road from an intersection. The expected packet delivery delay of a path can be modeled and expressed by parameters such as road density, average vehicle velocity, and the road distance. The minimum delay can be solved by a set of linear system equations.

- VADD is based on the idea of carry and forward
- Although geographical forwarding approaches such as GPSR which always chooses the next hop closer to the destination, are very efficient for data delivery in ad hoc networks, they may not be suitable for sparsely connected vehicular networks
- Transmit through wireless channels as much as possible
- If the packet has to be carried through certain roads, the road with higher speed should be chosen
- Due to the unpredictable nature of vehicular ad-hoc networks, we cannot expect the packet to be successfully routed along the pre-computed optimal path, so dynamic path selection should continuously be executed throughout the packet forwarding process

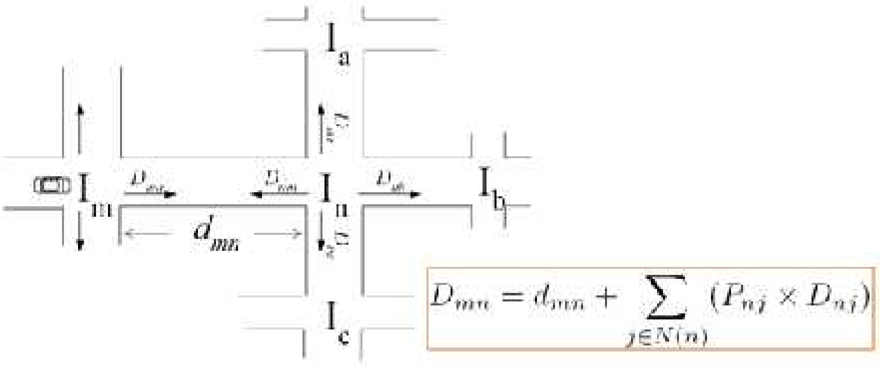


Fig 3.30: VADD routing [27]

Where

D_{mn}: the expected packet deliver delay from I_m to the destination if the packer carrier at I_m chooses to deliver the pack following road r_{mn}

P_{mn}: the probability that the packet is forwarded through road r_{mn} at I_m

N(n): the set of neighboring intersections of I_n

b) GeOpps: Geographical Opportunistic Routing [4]

Geographical Opportunistic Routing (GeOpps) [4] takes advantage of the suggested routes of vehicles' navigation system to select vehicles that are likely to move closer to the final destination of a packet. It calculates the shortest distance from packet's destination to the nearest point (NP) of vehicles' path, and estimates the arrival of time of a packet to destination. Figure shows Node A in computing the NP of its neighbors N1and N2. Since N2 offers closer NP to the destination, Node A picks N1to forward its packets.

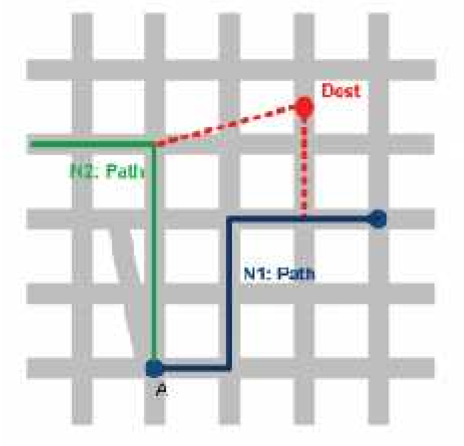


Fig 3.31: Calculation of the Nearest Point from packet’s Destination for N1 and N2 [4]

During the travel of vehicles, if there is another vehicle that has a shorter estimated arrival time, the packet will be forwarded to that vehicle. The process repeats until the packet reaches destination. The minimum delay used by VADD is indirectly obtained by selecting the next forwarding node whose path’s nearest point is closest to the destination. GeOpps requires navigation information to be exposed to the network, thus, privacy such as vehicle’s whereabouts might be an issue.

3.4 Cluster Based Routing: Cluster based routing is preferred in clusters. A group of nodes identifies themselves to be a part of cluster and a node is designated as cluster head will broadcast the packet to the cluster. Good scalability can be provide a large networks but network delays and overhead are incurred when formatting clusters in mobile VANET.

a) **CBRP: Cluster Based Routing [28, 29]**

The CBRP [28] uses a variation of the lowest-ID algorithm specified, which is also an identifier-based algorithm in order to support the cluster formation process each node uses a neighbor table, where it stores information about its neighbor nodes, such as their ID’s, their role in the cluster and the status of the link to that node. The neighbor table is maintained by periodically broadcasting HELLO messages. A HELLO message contains information about one node's state, its neighbor table and its cluster adjacency table.

The following states describe the clustering process depending on the current node state. These states are:

Undecided: This means the node does not belong to any cluster: this usually occurs if a new node appears in the network. Thus, if it receives a HELLO message from a cluster-head and there is a bi-directional link between them it changes its state to be member of the cluster indicated by the cluster-head. Otherwise it looks up in its neighbor table if it has any bi-directional links. If so, it becomes itself the cluster-head of a new cluster, if not, it remains in the undecided state and tries again.

Cluster-head: If a cluster-head detects that it has a bi-directional link to an-other cluster-head for a time period, it changes its state to member if the other cluster-head has a lower ID. Otherwise it stays the cluster-head and the other node has to change its state. This is a special case which may result in cluster reorganization.

Member: If a member loses its cluster-head, it looks for bi-directional links to other nodes. If it detects any, it changes its state to cluster-head if it has the lowest ID, otherwise it switches to the undecided state. Each member node belongs at least to one cluster.

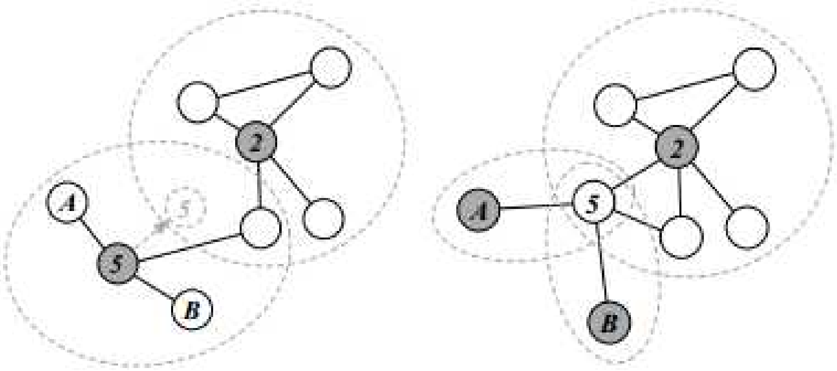


Fig 3.32: Creating clusters [29]

When cluster-head 5 moves into cluster 2 it gives up its role as cluster-head according to its higher ID. Nodes A and B who lost their cluster-head now form new clusters.

Routing [29]:

CBRP uses two data structures to support the routing process:

- The cluster adjacency table (CAT)
- The two-hop topology database.

The CAT stores information about neighboring clusters. This is, whether they are bi-directionally or uni-directionally linked.

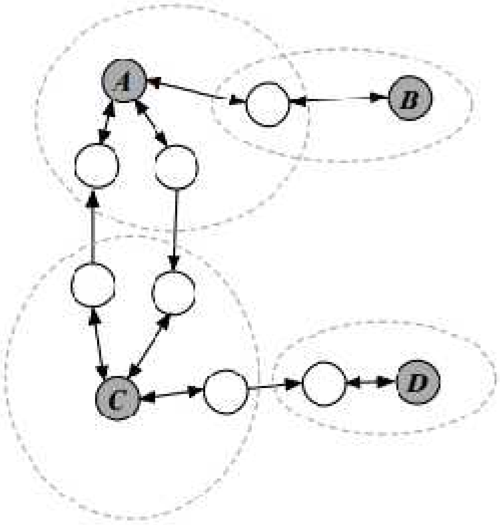


Fig 3.33: Bi-directionally and uni-directionally link [29]

In the figure clusters A, B and A, C are bi-directionally linked, clusters C, D are uni-directionally linked.

Route discovery [29]

Route discovery is done by using source routing. In the CBRP only cluster-heads are flooded with route request package (RREQ). Gateway nodes receive the RREQs as well, but without broadcasting them. They forward them to the next cluster-head.

This strategy reduces the network traffic. Initially, node S broadcasts a RREQ with unique ID containing the destination's address, the neighboring cluster-head including the gateway nodes to reach them and the cluster address list which consists of the addresses of the cluster-heads forming the route.

b) LORA_CBF: Location Routing Algorithm with Cluster-Based Flooding [30, 31]

The algorithm inherits the properties of reactive routing algorithms and has the advantage of acquiring routing information only when a route is needed.

LORA_CBF [30] has the following features:

- Firstly, this protocol improves the traditional routing algorithms, based on non-positional algorithms, by making use of location information provided by GPS.
- Secondly, it minimizes flooding of its Location Request (LREQ) packets. Flooding, therefore, is directive for control traffic as it uses only the selected nodes, called gateways, to diffuse LREQ messages.

The function of gateway nodes is to minimize the flooding of broadcast messages in the network by reducing duplicate retransmissions in the same region. Member nodes are converted into gateways when they receive messages from more than one cluster head. All the members in the cluster read and process the packet, but do not retransmit the broadcast message. This technique significantly reduces the number of retransmissions in a flooding or broadcast procedure in dense networks. Therefore, only gateway nodes retransmit packets between clusters (hierarchical organization). Moreover, gateways only retransmit a packet from one gateway to another in order to minimize unnecessary retransmissions, and only if the gateway belongs to a different cluster head.

Apart from normal Hello messages, the protocol does not generate extra control traffic in response to link failures and additions. Thus, it is suitable for networks with high rates of geographical changes. As the protocol keeps only the location information of the [source, destination] pairs in the network, the protocol is particularly suitable for large and dense networks with very high mobility.

The protocol is also designed to work in a completely distributed manner and does not depend upon any central entity. The protocol does not require reliable transmission for its control messages, because each node sends its control messages periodically and can, therefore, sustain some packet loss. This is, of course, important in radio networks like the one being considered here, where deep fades are possible.

Location Routing Algorithm with Cluster-Based Flooding (LORA_CBF) carries out different functions that are needed to perform the task of routing [31]:

- Neighbor sensing: Each node must detect the neighbor nodes with which it has a direct link. To accomplish this, each node periodically broadcasts a Hello message, containing its location information, address and status. These control messages are transmitted in broadcast mode and received by all one-hop neighbors, but they are not relayed to any further nodes. A Hello message contains the following information:

- Node Address.
- Type of node (Undecided, Member, Gateway or Cluster head).
- Location (Latitude and Longitude).

Cluster-Based Flooding: LORA_CBF must have one cluster head, zero or more members in every cluster, and one or more gateways, in order to communicate with other cluster heads. Each cluster head maintains a "Cluster Table," which is defined as a table that contains the addresses and geographic locations of the member and gateways nodes. We have assumed that all nodes can ascertain their positions via GPS or some local coordinate system.

When a source attempts to send data to a destination, it first checks its routing table to determine if it knows the location of the destination. If it does, it sends the packet to the closest neighbor to the destination. Otherwise, the source stores the data packet in its buffer, starts a timer and broadcasts Location Request (LREQ) packets. Only gateways and cluster heads can retransmit LREQ packets. Gateways only retransmit a packet from one gateway to another in order to minimize unnecessary retransmissions, and only if the gateway belongs to a different cluster head.

Upon receiving a location request, each cluster head checks to see if the destination is a member of its cluster. Success triggers a Location Reply (LREP) packet that returns to the sender using geographic routing, because each node knows the position of the source and the closest neighbor, based on the information received from the LREQ and the neighbor sensing mechanism. Failure triggers retransmissions by the cluster head to adjacent cluster heads, where the destination address is recorded in the packet. Cluster heads and gateways, therefore, discard location request packets they have already seen.

Once the source receives the location of the destination, it acquires the data packet from its buffer and sends it to the closest neighbor to the destination. Essentially, the algorithm consists of four stages:

- Formation of clusters.
- Location discovery (LREQ and LREP).
- Routing of data packets.
- Maintenance of location information.

3.5 Geo Cast Based Routing: Geocast routing is basically a location based multicast routing. Its objective is to deliver the packet from source node to all other node within a specified geographical region (Zone Of Relevance).

a) DTSG: Dynamic Time-Stable Geo-cast [32, 33]

Dynamic Time-Stable Geo-cast [33] Routing is a unique protocol as it also works well in low-density networks.

A highway has significantly variable density. In rush hour, the density is high, but in the middle of the night, the density is too low. A good protocol should work well in both situations. But in case of most other protocols in low-density networks, the network becomes disconnected.

In DTSG the relaying node would store the message in its buffer and then try to move toward the destination until the destination is within its broadcast range. Then, it is time to relay the message to the destination. Based on this general solution, for geo-casting, our proposed protocol relies on opposite lane vehicles to relay the message through the region.

Although the opposite lane vehicles may not be interested in the message, they can help inform the interested vehicles while they pass by them. In addition, these vehicles, depending on the perceived user density, determine and dynamically adapt an extra distance where they continue relaying the message. Therefore, the opposite lane vehicles, before exiting the extra region, deliver the message to at least one of the vehicles coming toward the event.

Thus, the opposite-lane vehicles are useful for two reasons [32]:

First, they increase the probability of the delivery ratio of the message to all vehicles moving toward the event, especially in sparse networks. As shown in Figure, two de-fragmentations occur in the vehicle groups in the lane moving toward the event. The first vehicle in each group is not in the broadcasting range of the last vehicle of the following group. In this case, the opposite-lane vehicles play vital role in message relaying.

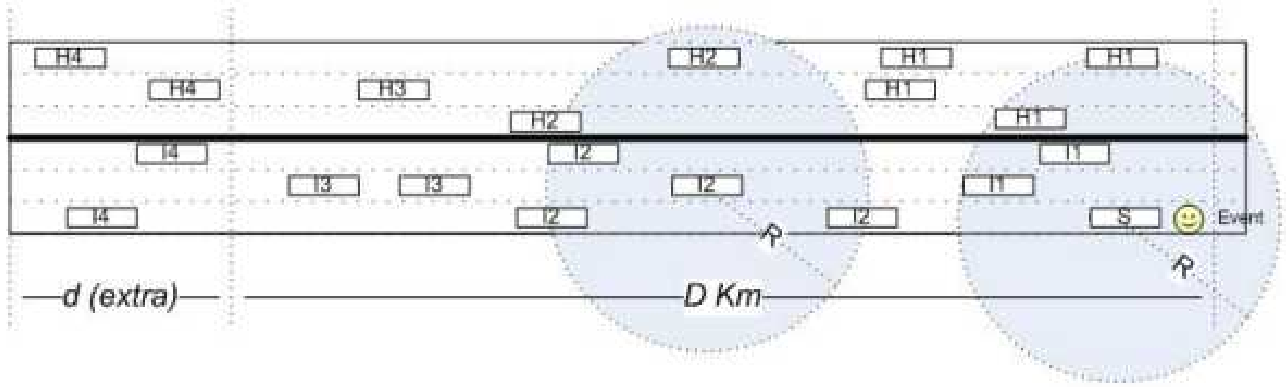


Fig 3.34: DTSG routing [32]

Second, the opposite-lane vehicles can help maintain the message for duration of time T on sparse roads. Thus, at least one vehicle remains in the region for the entire duration of the time T. Another focus of this work is a new idea that we call dynamic time-stable geo-casting. If the disseminated event has been removed before the expiration of the time T, there should be a way to stop disseminating the message. As this relaying continues until the time T expiry, all the vehicles moving toward the removed event are still being informed, an undesirable situation.

There should be a technique to cancel this dissemination. If the problem continues in the region, the stable message should be relayed longer than the expiry of time T. It is also clear that there should be some mechanism to extend the time for relaying of the message within the region.

A simple way to extend the time is to geo-cast another message just after the expiry of the previous one. Then, this new message can also be stable in the region. This method seems satisfactory, but a good protocol should not only guarantee the delivery of the message, but also lower the network cost.

b) IVG: Inter-Vehicle Geo-cast [33]

The Inter-Vehicle Geo-cast protocol [33] uses a timer-based mechanism for message forwarding and periodic broadcasts are used to overcome network fragmentation. The purpose of IVG is to inform vehicles located in a risk area called multicast group about any danger on the highway (e.g., when an accident occurs). To achieve this goal, the risk area is determined considering the precise obstacle location on the road and the driving directions which can be affected. The damaged vehicle broadcasts a message alert to the multicast group (Figure).

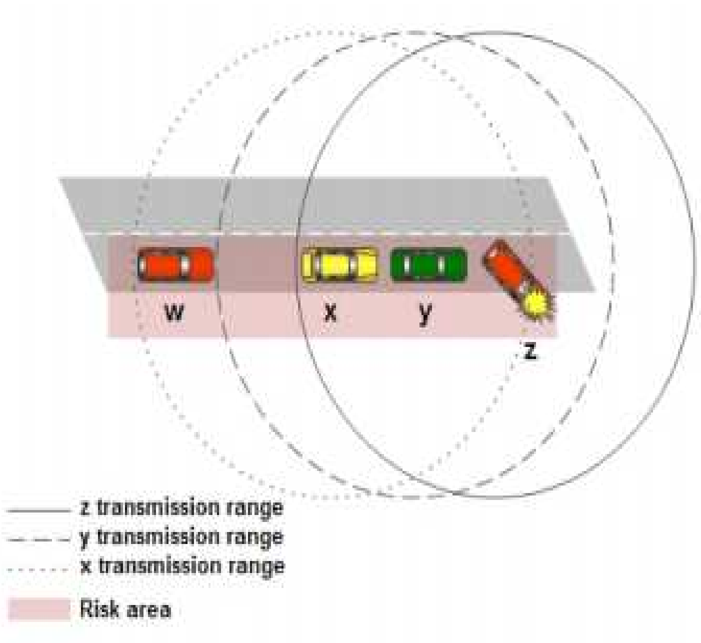


Fig 3.35: IVG Relay selection: x is more distant to z then y. x is a relay. x permits to reach w while y not.

The neighbors receiving the message test its relevance according to their location reporting to the risk area. All neighbors belonging to the risk area calculate a different time back-off that promotes the furthest node in order to be a relay rebroadcasting the message (more distant is more favorable). This relay selection technique makes the use of periodic beacons unnecessary.

c) Rover: Robust Vehicular Routing [34]

Rover [34] is a reliable geographical multicast protocol where only control packets are broadcasted in the network and the data packets are uni-casted.

The objective of the protocol is to send a message to all other vehicles within a specified Zone of Relevance (ZOR). The ZOR is defined as a rectangle specified by its corner coordinates. A message is defined by the triplet [A, M, Z] it indicates specified application, message and identity of a zone respectively. When a vehicle receives a message, it accepts the message if it is within the ZOR. It also defines a Zone of Forwarding (ZOF) which includes the source and the ZOR.

Characteristics:

- Each vehicle is identified by an Identification Number,
- Each vehicle is equipped with a GPS receiver,
- Vehicles have access to a digital map,
- ZOR is a rectangle area,
- ZOF includes the sender and the ZOR.

The goal of ROVER is to deliver an application generated message to all vehicles located into the specified ZOR. ROVER defines a message as a triplet [Application, Message and ZOR]. A vehicle considers a message if it belongs to the message's ZOR.

3.6 Broad Cast Based Routing: Broadcast based routing is frequently used in VANET for sharing, traffic, weather and emergency, road conditions among vehicles and delivering advertisements and announcements.

a) DV-CAST Distributed vehicular broadcast [35, 36]

Distributed vehicular broadcast protocol [36] uses local topology information by using the periodic hello messages for broadcasting the information. Each vehicle uses a flag variable to check whether the packet is redundant or not. This protocol divides the vehicles into three types depending on the local connectivity as:

- Well connected neighborhood,
- Sparsely connected neighborhood,
- Disconnected neighborhood.

In well connected neighborhood it uses persistence scheme (weighted p persistence, slotted 1and p persistence).

In sparsely connected neighborhood after receiving the broadcast message, vehicles can immediately rebroadcast with vehicles moving in the same direction.

In totally disconnected neighborhood vehicles are used to store the broadcast message until another vehicle enters into transmission range, otherwise if the time expires it will discard the packet.

This protocol causes high control overhead and delay in end to end data transfer.

Routing Parameters [35]: The most important parameters for DV-CAST protocol are the local topology information and the Region of Interest. In particular, each vehicle should be able to

- Determine whether it is the intended recipient of the message that is moving in the same direction as the source;
- Determine whether it is the last vehicle in the group/cluster; and
- Determine whether it is connected to at least one vehicle in the opposite direction.

These three parameters are denoted in this paper as Destination Flag (DFlg), Message Direction Connectivity (MDC), and Opposite Direction Connectivity (ODC), respectively.

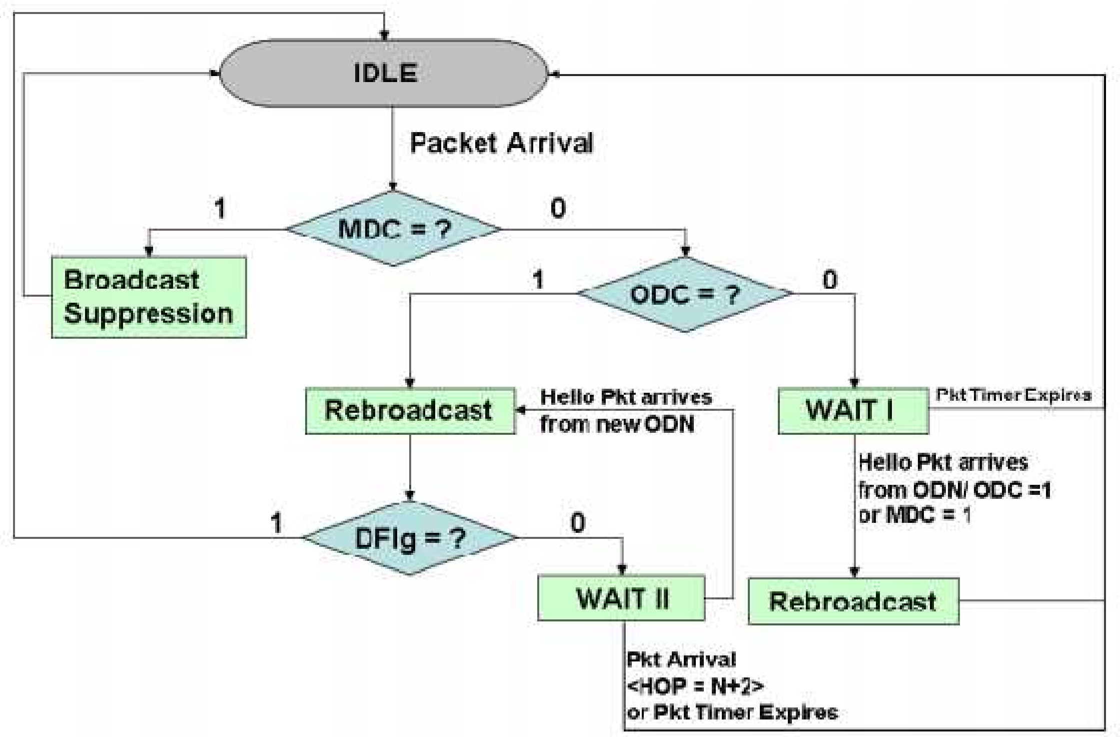

IDLE
Packet Arrival
1
MDC = ?
0
Broadcast Suppression
1
ODC = ?
0
Rebroadcast
Hello Pkt arrives from new ODN
WAIT I
Pkt Timer Expires
Hello Pkt arrives from ODN/ ODC =1 or MDC = 1
1
DFlg = ?
0
WAIT II
Rebroadcast
Pkt Arrival <HOP = N+2> or Pkt Timer Expires

Fig 3.36: Decision tree for DV-CAST protocol [35].

Routing Rules [35]: In order to handle the broadcast message properly, we propose that each vehicle follows two basic routing rules:

- If Destination Flag (DFlg) is set to 1, vehicle should ignore any duplicate broadcast or follow the diagram in Figure if the message is received for the first time.
- If DFlg is set to 0, vehicle is a relay node and should follow the routing diagram.

b) EAEP: Edge-aware epidemic protocol [35, 37]

Edge-aware epidemic protocol [37] is reliable, bandwidth efficient information dissemination based highly dynamic VANET protocol. It reduces control packet overhead by eliminating exchange of additional hello packets for message transfer between different clusters of vehicles and eases cluster maintenance. Each vehicle piggybacks its own geographical position to broadcast messages to eliminate beacon messages.

Upon receiving a new rebroadcast message, EAEP uses number of transmission from front nodes and back nodes in a given period of time to calculate the probability for making decision whether nodes will rebroadcast the message or not.

But EAEP does not address the intermittent connectivity issue. Specifically, a node does not know whether it has missed any messages to its new neighbors or its neighbors have missed some messages [37]. EAEP overcomes the simple flooding problem but it incurs high delay of data dissemination.

Epidemic protocols are probabilistic information dissemination protocols which do not require any knowledge of the local and global network topology.

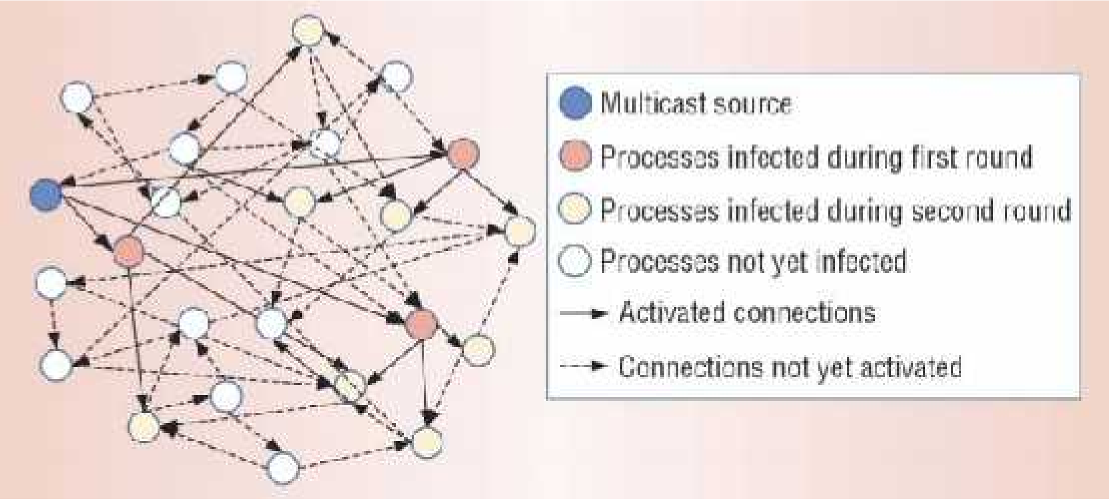

Multicast source
Processes infected during first round
Processes infected during second round
Processes not yet infected
Activated connections
Connections not yet activated

Fig 3.37: EAEP routing [37]

The proposed scheme releases the requirements for:

- Infrastructure support
- Exchange of “Hello” messages
- Cluster maintenance

Chapter 4: SECURITY

4. **VANET Security:**

VANET is such technology what ensure the security of data communication. Vehicular Ad Hoc Networks require a mechanism to help authenticate messages, identify valid vehicles, and remove malevolent vehicles. It ensures the confidentiality, integrity, non-repudiation and Real time guarantee of a message. VANET technology also confirms the entity authentication and security for each vehicle.

4.1 Possible Attacks & Threats:

In each communication, attack is a common affair. VANET face many attacks. We have discussed few of security attack here.

4.1.1 Denial of Service attack:

For this attack an attacker create a network jam in the communication network. As a result, source cannot send data or information to the destination. Attacker may use vehicle information to make a network jam. This network jam may harm for any kind of critical information to arrive. This attack is so dangerous for traffic and any kinds of communication and it's depends on information. If a vehicle filling trouble on highway but it can't send emergency or help message to nearby police station, hospital and fire service for this attack.

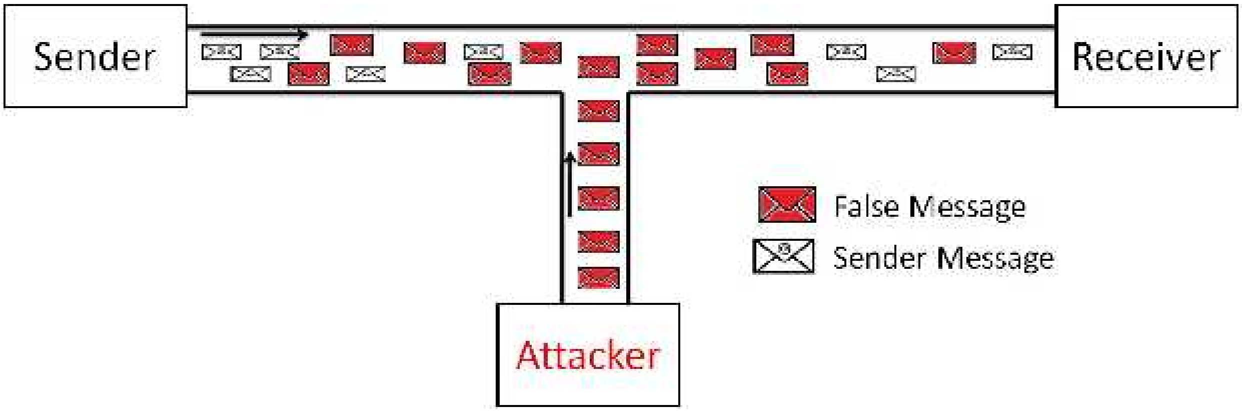


Figure 4.1: DoS attack.

4.1.2 Message Suppression Attack:

For this attack an attacker selectively fetch and drop packets from the communication channel and the packet may be very important. The attacker suppress packets can use for next time. The intention of this type of attacker is to interrupt sequential message sending to the destination and the another goal of such an attacker would be to prevent registration and insurance authorities from learning about collisions involving his vehicle and/or to avoid delivering collision reports to roadside access points[51]. An attacker may suppress a congestion warning, and use it in another time, as a result the vehicle will not receive the warning and forced to wait in the traffic.

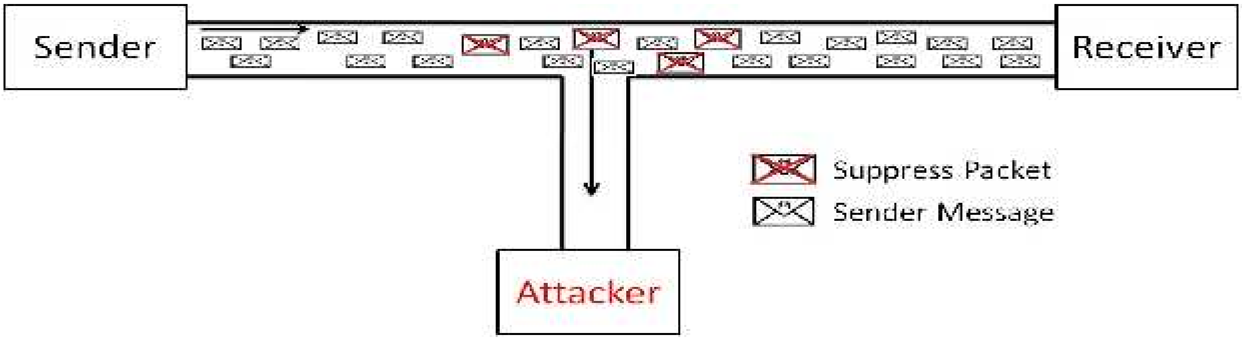


Figure 4.2: Message suppression attack.

4.1.3 Fabrication Attack:

For this attack an attacker make some false message and sends it to the network and claim that its coming from those sender. Victim may get compromised with the false message and return acknowledge for that. Sometimes attacker may fabricate false warning and message what may make trouble for the receiver and sender. Attacker may take benefit for the fabricate message from the destination.

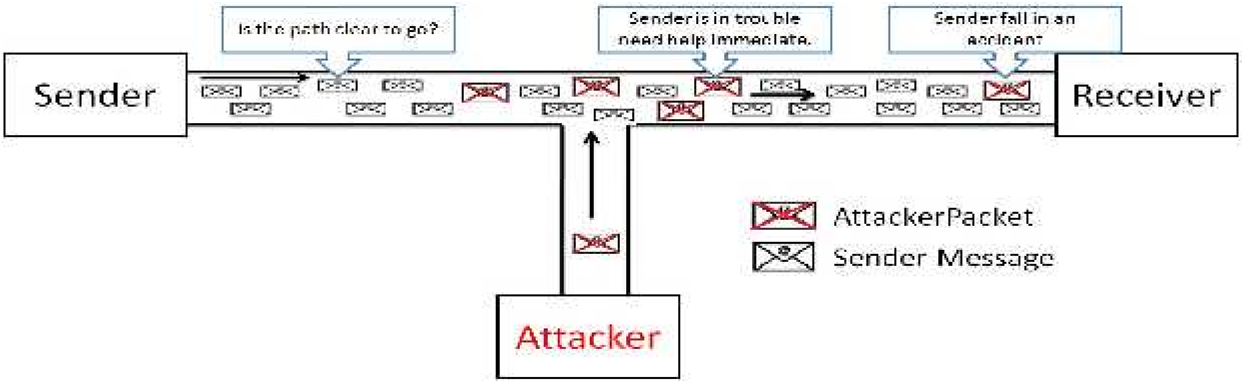


Figure 4.3: Fabrication attack.

4.1.4 Alteration Attack:

For this attack, an attacker exchange real information with the false information in the communication network. This attack happens when attacker alters an existing data. It includes delaying the transmission of the information, replaying earlier transmission, or altering the actual entry of the data transmitted [51],[52]. An attacker can alter a message telling other vehicles that is the path clear but it is altered by sender is in trouble.

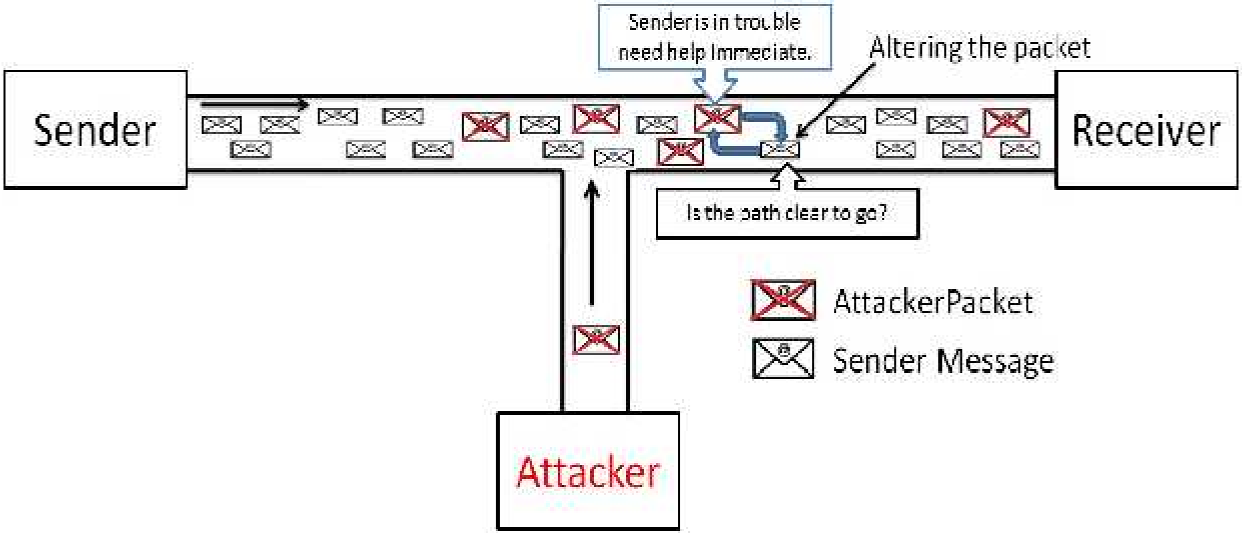


Figure 4.4: Alteration attack.

4.1.5 Replay Attack:

For this attack an attacker copy packet from the communication channel and the packet may be really important. Attacker uses the packet for the next time to get benefit of that situation. Basic 802.11 security has no protection against replay [51]. It does not contain sequence numbers or timestamps. Because of keys can be reused, it is possible to replay stored messages with the same key without detection to insert bogus messages into the system [51]. Each packets must be authenticated, not just encrypted. Packets must have timestamps. The motive of this type of attacker is to confuse others.

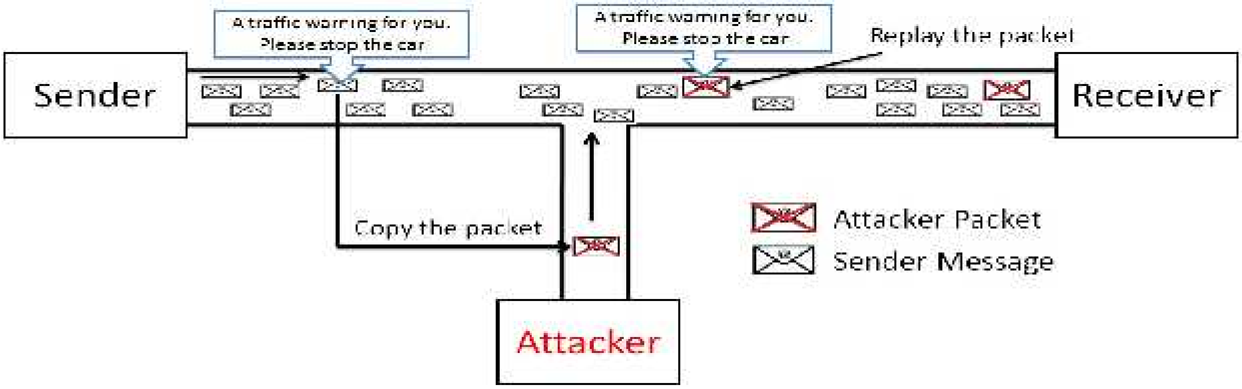

A traffic worning for you. Please stop the car
A traffic waming for you. Please stop the car
Replay the packet
Sender
Receiver
Copy the packet
Attacker Packet
Sender Message
Attacker

Figure 4.5: Replay attack.

4.1.6 Sybil Attack:

For this attack an attacker make a large number of false traffic and claim that there is many vehicle and they makes a jam. The attacker suggests other vehicle to use alternative path. A Sybil attack depends on how cheaply identities can be generated, the degree to which the system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the system treats all entities identically[51].

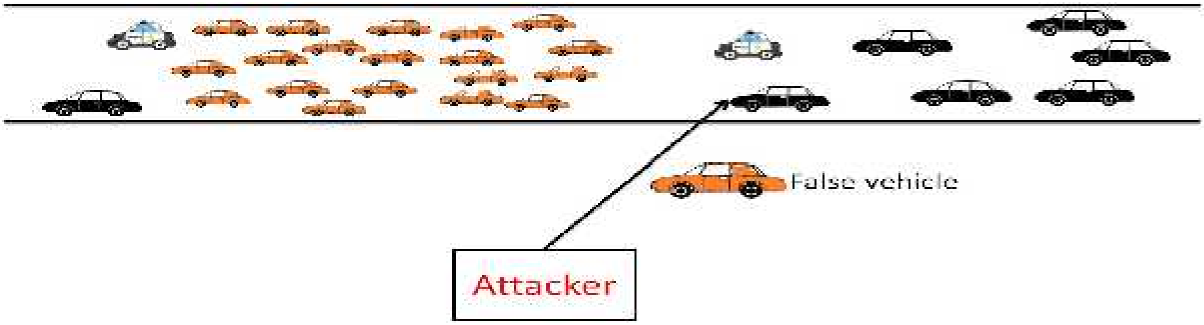


Figure 4.6: Sybil attack

4.1.7 Snooping / Eavesdropper:

Those people who try to collect information about you. While data mining is acceptable over aggregate data, but for identifying information for an individual, that raises serious privacy concerns and is not acceptable. Impersonation is a type of attack done by the snoops .An attacker may take on someone else's identity and gain certain advantages or cause damage to other vehicles. Privacy violation is also done by the snoops and by using a simple mechanism which is to associate the identity of vehicles with the messages they send using asymmetric key cryptography.

4.1.7 Prankster:

Prankster is especially the board teenagers who will attempt things for fun. For example, a prankster targeting collision avoidance might sit by the road and convince one vehicle behind the speed up. A prankster also abuses the security vulnerability of Dos attack to disable the network.

4.2 Security requirements:

Security is the major concern for any kinds of communication. To ensure security some procedures have to follow for the communication. VANET has some security requirements.

4.2.1: Message authentication [53]:

For vehicular communication, every message must have to authenticate and have to ensure the authorization for the message. To ensure authentication every vehicle will assign private & public key. Sender encrypts the message with receiver public key. Signing each message with this, causes an overhead, to reduce this overhead we can use the approach ECC (Elliptic Curve Cryptography), the efficient public key cryptosystem [51].

4.2.2 Message integrity [54][55]:

This is the validity of a transmitted message from other vehicle. This method ensures that the contents of a message did not tamper. The most common and useful approach is to use a one-way hash function and that combines all the bytes in the message with a secret key and produces a message digest that is quit impossible to reverse or understand . Integrity checking is one component of an information security program.

4.2.3 Message Confidentiality [56][57][63]:

Confidentiality prevent the message from unauthorized access. When a vehicle sends a message to another vehicle if the message accessed by other then the message will lose its confidentiality. In vehicular communication message must be confidential otherwise the message may harmful for receiver or sender vehicle. Confidentiality is one of the major requirements in data communication.

4.2.4 Entity Authentication [53][58]:

For vehicular communication every vehicle have to authenticate its identity. Sender or receiver may claim for entity authentication. Entity authentication is very useful to ensure vehicle type.

4.2.5 Access Control [59]:

Access on any object or vehicle must be strictly controlled according to the access control policy. Access control will prevent unauthorized physical or logical access to any vehicle. Access control policies should be reviewed on a regular basis. All vehicle and object should have a unique identifier for their personal use only, and suitable authentication techniques should be chosen [64].

4.2.6 Privacy [60]:

In vehicular communication the privacy of sender and receiver mast have to ensure. Shared information and private information will be according to the security policy.

4.2.7 Non- Repudiation [61][62]:

Non-repudiation will facilitate the ability to identify the attackers even after the attack happens. This prevents cheaters from denying their crimes. Any information related to the car like the trip rout, speed, time, any violation will be stored in the TPD, any official side holding authorization can retrieve this data [51].

4.2.8 Real time guarantees:

To avoid collision the message must be happened on real time or online.

Chapter 5: DESIGN AND SIMULATION

5.1 Simulator and analyzer:

Actually a VANET simulator has two parts. One is network component and it is capable of simulating the behavior of a wireless network. Second is vehicular traffic component and this is capable to provide an accurate mobility model for the nodes of a VANET. Depending on the simulation the simulator can contain others components. To describe them, we refer to the NS2 [47][48][49][50] or NS3, VNSim[66], VanetMobiSim[65], Qualnet[43][44], Sumo Simulator. We have worked on NS2. To analyze traffic we refer NS2-VisualTraceAnalyzer, Xgraph and trace analyzer. In our work we have used NS2-VisualTraceAnalyzer and Xgraph. We have use Comodo edit 7 to write simulation code. For coding we have used otcl and c++ language.

5.2 Simulation environment:

The simulation environment created on windows 7 by using cygwin. To make the simulation model we used NS2 simulator [47][48][49][50]. The NS2 simulator instruction has been use to define the topology structure of the network. For node configuration we used otcl [39][40][41][42][45][46][47] language instructions.

5.3 Traffic model:

The source and destination are spread randomly over the network. We have used TCP with source node and TCPSink with destination node. We have attach FTP data source with TCP. TCP packet size 512B and TcpSink packet size 210B. The maximum data source packet is 2048. Link bandwidth is 10Mbps. According to the number of source-destination pairs can be varied to the packet-sending rate in each pair.

5.4 Mobility model:

The model uses the random waypoint[67] model in a rectangular field. The field configurations are 3000 m × 1600 m field with 100 nodes. Here, each packet starts its journey from a random location to a random destination. Nodes are moving at the speed of 0-84m/s. Sources and destinations are changing with respect of time. Simulations are run for 600 simulated seconds. Identical mobility and traffic scenarios are used across protocols to gather fair results.

5.5 Communication models:

Communication models highlight the information flows between two vehicles and other moving object. VANET applications are affected by wireless networking aspects such as throughput, jitter, congestion window, bandwidth, transmission delay, packet loss or network access scheme. However, accurate network simulation introduces additional complexity and makes several large-scale VANET applications unsuitable for simulation.

5.6 Node diagram:

Mobile node diagram in NS2 simulator for DSDV and DSR.

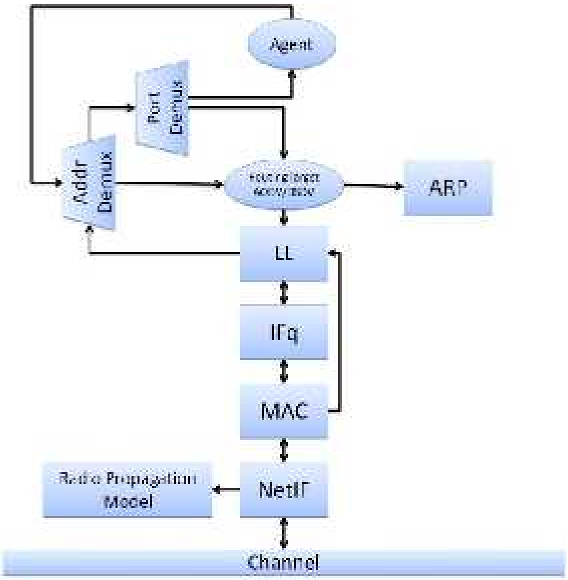


Figure 5.1: DSDV node diagram [69]

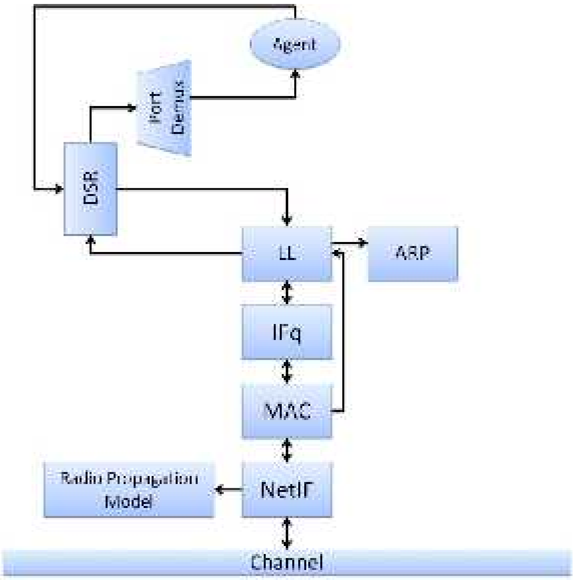


Figure 5.2: DSR node diagram [68]

5.7 Traffic flow diagram:

Traffic flow direction and data source for our design and simulation.

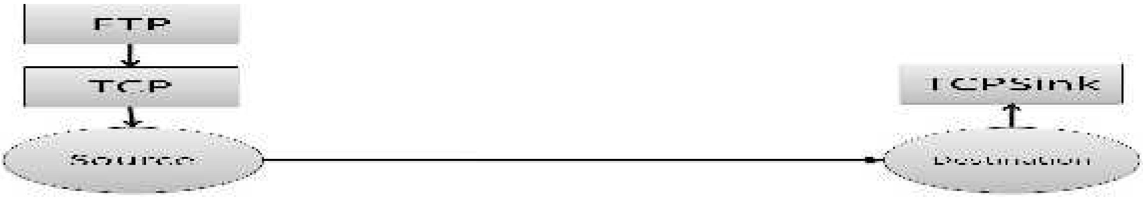


Figure 5.3: Traffic flow diagram

5.8 Design interface & scenario:

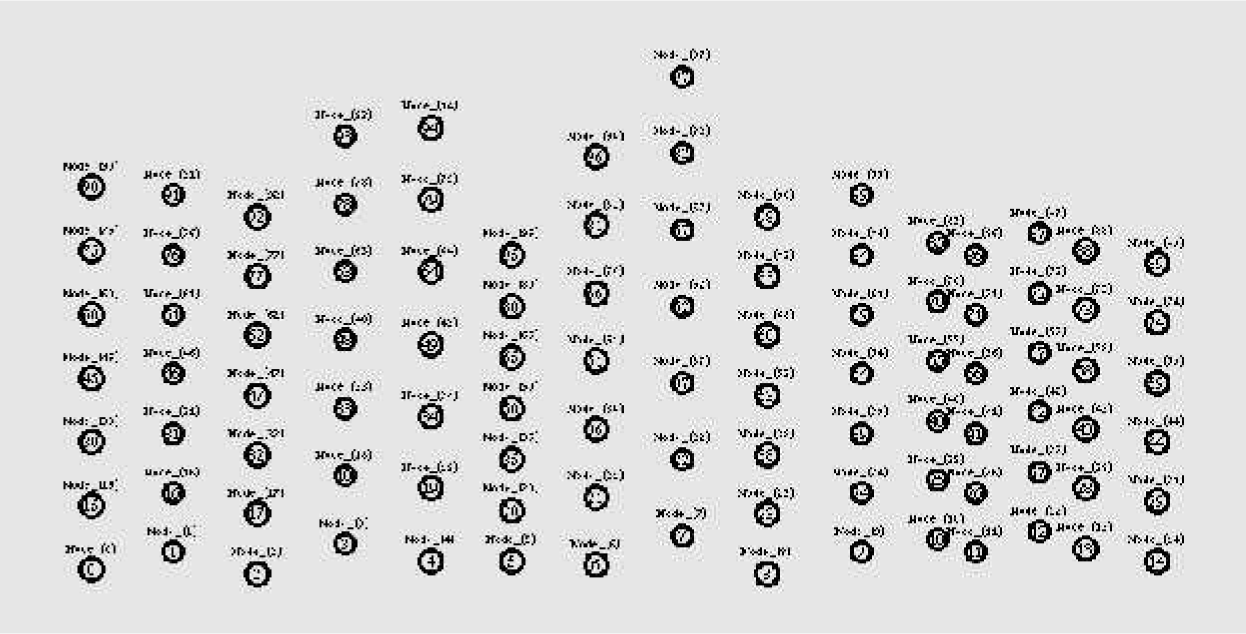


Figure 5.4: Simulation aria scenario

5.9 Simulation parameters Initialization:

```
set val(chan)  Channel/WirelessChannel          ;# channel type
set val(prop)   Propagation/TwoRayGround        ;# radio-propagation model
set val(netif)   Phy/WirelessPhy                ;# network interface type
set val(mac)    Mac/802_11                      ;# MAC type
set val(ifq)     Queue/DropTail/PriQueue        ;# interface queue type
set val(ll)      LL                             ;# link layer type
set val(ant)     Antenna/OmniAntenna            ;# antenna model
set val(ifqlen)  50                             ;# max packet in ifq
set val(nn)      100                            ;# number of mobilenodes
set val(rp)      AODV                           ;# routing protocol
set val(x)       3000                           ;# X dimension of topography
set val(y)       1600                           ;# Y dimension of topography
set val(stop)   600.0                           ;# time of simulation end
```

5.10 Tcl script initialization:

```
set ns [new Simulator]
set namfile [open AODV_final.nam w]
set Time [open time.tr w]
set cwnd1 [open cwnd1.tr w]
set cwnd2 [open cwnd2.tr w]
set cwnd3 [open cwnd3.tr w]
set cwnd4 [open cwnd4.tr w]
set cwnd5 [open cwnd5.tr w]
set b1 [open b1.tr w]
set b2 [open b2.tr w]
set b3 [open b3.tr w]
set b4 [open b4.tr w]
set b5 [open b5.tr w]
$ns namtrace-all $namfile
set wireless_tracefile [open AODV_final.trace w]
set topography [new Topography]
$ns trace-all $wireless_tracefile
$ns namtrace-all-wireless $namfile 3000  1600
$topography load_flatgrid 3000  1600
```

5.11 Node configurations:

The TCL code for node configuration has given below.

```
$ns node-config    -adhocRouting AODV \
                    -llType LL \
                   -macType Mac/802_11\
                   -ifqLen 50 \
                   -ifqType Queue/DropTail/PriQueue \
                   -antType Antenna/OmniAntenna \
                   -propType Propagation/TwoRayGround \
                   -phyType Phy/WirelessPhy \
                    -channel [new Channel/WirelessChannel] \
                    -topoInstance $topography \
                    -agentTrace ON \
                    -routerTrace ON \
                    -macTrace ON \
                     -movementTrace ON
```

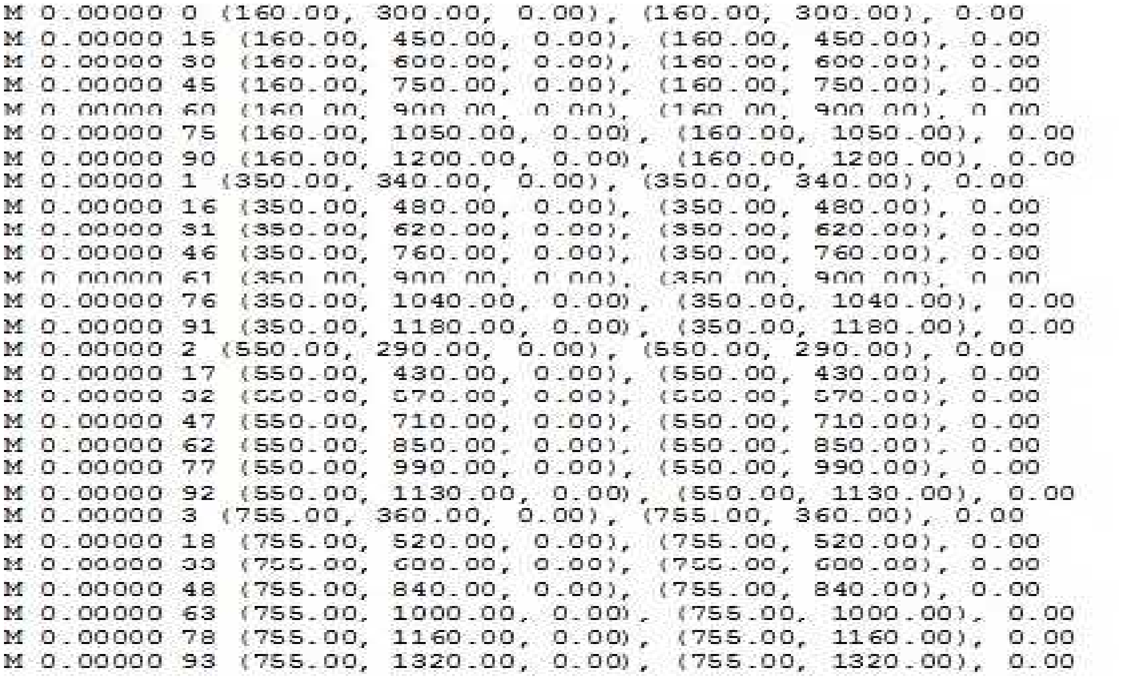
5.12 Bandwidth calculation procedure:

```
proc calcByte {sink file} {
     global ns
     set time 0.2
     set bw0 [$sink set bytes_]
     set now [$ns now]
     puts $file " [expr {$bw0 / $time * 8 / 1000000}] "
     $sink set bytes_ 0
     $ns at [expr $now + $time] "calcByte $sink  $file "
   }
```

5.13 Congestion window calculation procedure :

```
proc calcCwnd {tcpSource file} {
     global ns
     set time 0.2
     set now [$ns now]
     set cwnd [$tcpSource set cwnd_]
     puts $file "$cwnd"
     $ns at [expr $now + $time] " calcCwnd $tcpSource $file"
  }
```

5.14 Trace file Sample:



```
M 0.00000 0 (160.00, 300.00, 0.00), (160.00, 300.00), 0.00
M 0.00000 15 (160.00, 450.00, 0.00), (160.00, 450.00), 0.00
M 0.00000 30 (160.00, 600.00, 0.00), (160.00, 600.00), 0.00
M 0.00000 45 (160.00, 750.00, 0.00), (160.00, 750.00), 0.00
M 0.00000 60 (160.00, 900.00, 0.00), (160.00, 900.00), 0.00
M 0.00000 75 (160.00, 1050.00, 0.00), (160.00, 1050.00), 0.00
M 0.00000 90 (160.00, 1200.00, 0.00), (160.00, 1200.00), 0.00
M 0.00000 1 (350.00, 340.00, 0.00), (350.00, 340.00), 0.00
M 0.00000 16 (350.00, 480.00, 0.00), (350.00, 480.00), 0.00
M 0.00000 31 (350.00, 620.00, 0.00), (350.00, 620.00), 0.00
M 0.00000 46 (350.00, 760.00, 0.00), (350.00, 760.00), 0.00
M 0.00000 61 (350.00, 900.00, 0.00), (350.00, 900.00), 0.00
M 0.00000 76 (350.00, 1040.00, 0.00), (350.00, 1040.00), 0.00
M 0.00000 91 (350.00, 1180.00, 0.00), (350.00, 1180.00), 0.00
M 0.00000 2 (550.00, 290.00, 0.00), (550.00, 290.00), 0.00
M 0.00000 17 (550.00, 430.00, 0.00), (550.00, 430.00), 0.00
M 0.00000 32 (550.00, 570.00, 0.00), (550.00, 570.00), 0.00
M 0.00000 47 (550.00, 710.00, 0.00), (550.00, 710.00), 0.00
M 0.00000 62 (550.00, 850.00, 0.00), (550.00, 850.00), 0.00
M 0.00000 77 (550.00, 990.00, 0.00), (550.00, 990.00), 0.00
M 0.00000 92 (550.00, 1130.00, 0.00), (550.00, 1130.00), 0.00
M 0.00000 3 (755.00, 360.00, 0.00), (755.00, 360.00), 0.00
M 0.00000 18 (755.00, 520.00, 0.00), (755.00, 520.00), 0.00
M 0.00000 33 (755.00, 680.00, 0.00), (755.00, 680.00), 0.00
M 0.00000 48 (755.00, 840.00, 0.00), (755.00, 840.00), 0.00
M 0.00000 63 (755.00, 1000.00, 0.00), (755.00, 1000.00), 0.00
M 0.00000 78 (755.00, 1160.00, 0.00), (755.00, 1160.00), 0.00
M 0.00000 93 (755.00, 1320.00, 0.00), (755.00, 1320.00), 0.00
```

Figure 5.5: Trace file sample

5.15 Delay:

Figure 5.6 shows about the delay of AODV for different node. Delays for all nodes are not same. We have showed the delay of AODV protocol for different node and moving object. Delay is high for those nodes who are moving fast. AODV generate many ACK packets, so the ACK packet in AODV is indeed a great bottleneck [71]. As it was reactive protocol, so the delay little bit more.

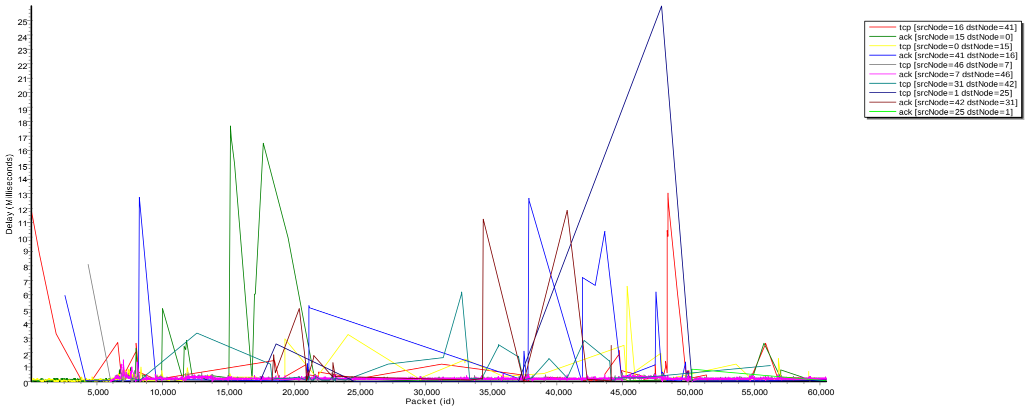


Figure 5.6: Delay per packet

5.16 Jitter:

Jitter is the standard deviation of packet delay between all nodes and object.

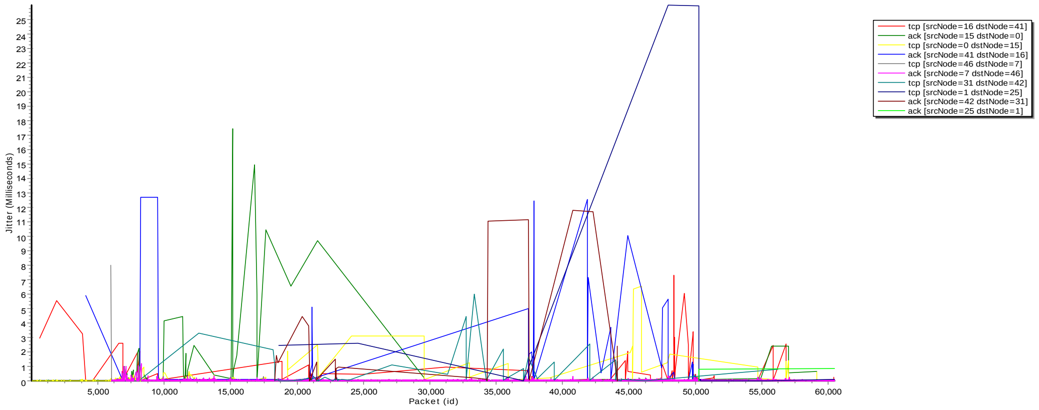


Figure 5.7: Jitter per packet

5.17 Throughput:

Figure 5.8 shows the transferred throughput with respect of time for an node. In the figure at the beginning throughput is high after few moments throughput break down and its fluctuate for several times. Throughput describes the loss rate as seen by the transport layer. It reflects the completeness and accuracy of the routing protocol. From these graphs it is clear that throughput de-crease with increase in mobility. As the packet drop at such a high load traffic is much high[70].

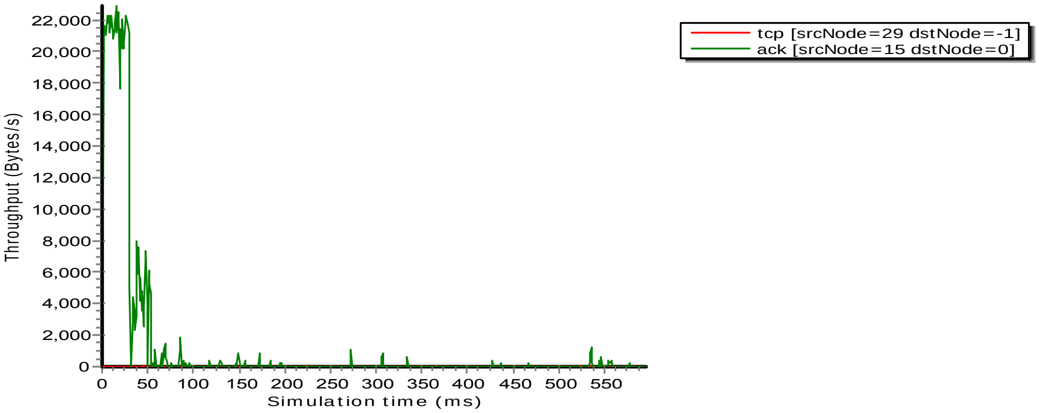


Figure 5.8: Throughput transferred for a node

5.18 Bandwidth in destination node:

The figure 5.9 shows that the bandwidth uses curve for the destination node.

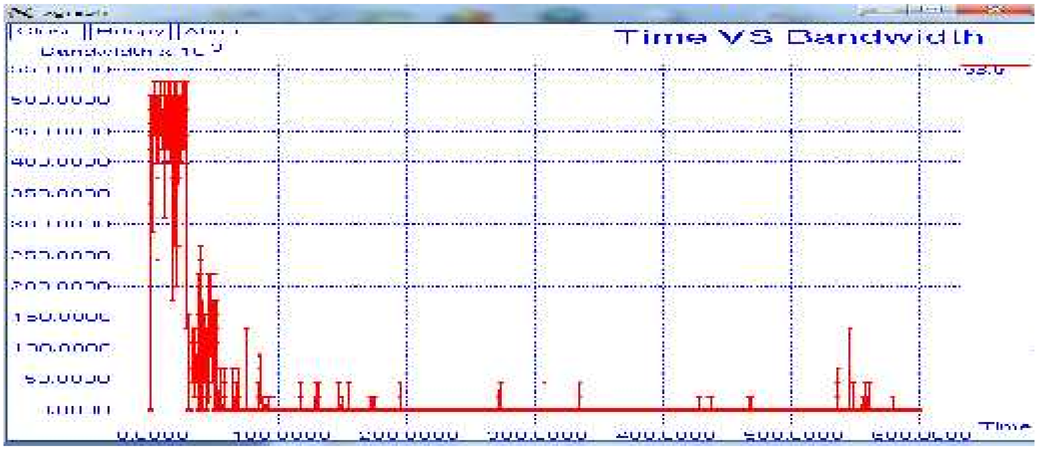


Figure 5.9: Bandwidth transfer for the destination node

5.19 Congestion Window

Congestion window increases according to the number of betting ACK receive.

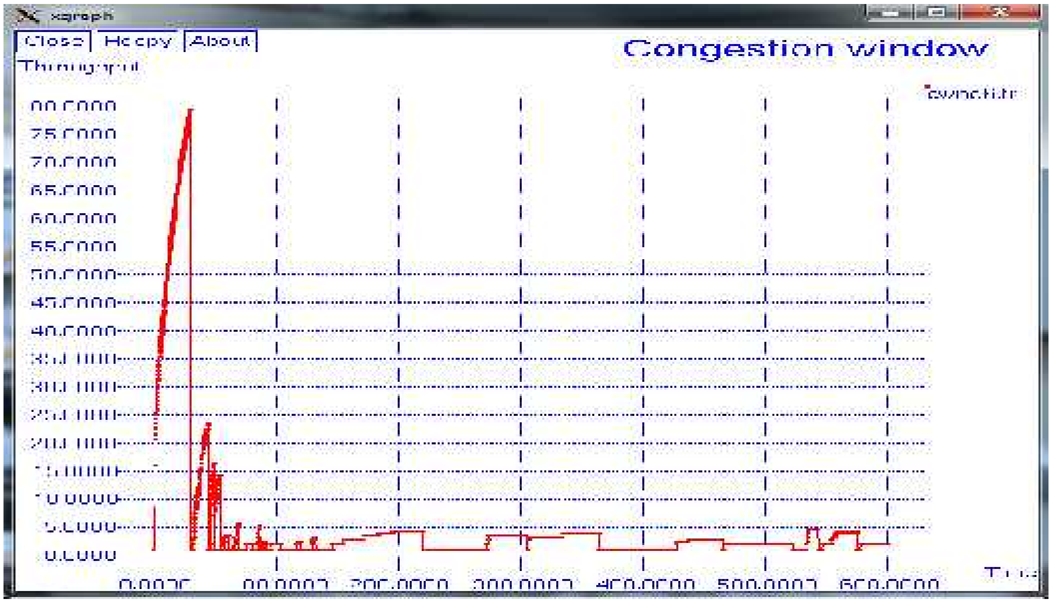


Figure 5.10: Congestion window for a node

Chapter 6: ANALYSIS

For simulation, I have used same configuration for both topology. I have use 100 wireless node five TCP connection and traffic source is FTP. TCP packet size 512B and TcpSink packet size 210B. The maximum data source packet is 2048. Link bandwidth is 10Mbps. I have analyzed Throughput, Jitter, Delay, Congestion window and bandwidth for destination window with graph. Now i am presenting the comparison between proactive (DSDV) (Proactive) and reactive (AODV) (Reactive) protocol in VANET technology.

6.1 Node in long distance:

For Node 0(Source) and Node 15(Destination), here node 0 and node 15 are in same coverage and both is moving object and they start their journey at 10s node 0 moves to node 15 position at the speed of 75m/s and node 15 move 2648m from its position at the speed of 12.97m/s. When time is 12s node 0 reached to node 15 position and node 15 become 25.94 far from node 0. At the time of 29.73s node 15 become out of range from node 0 for first time and connection is drop. In reactive (AODV) connections reconnect by changing the routing but in proactive (DSDV) connection can't reconnect due to table update.

6.1.1 Throughput transferred:

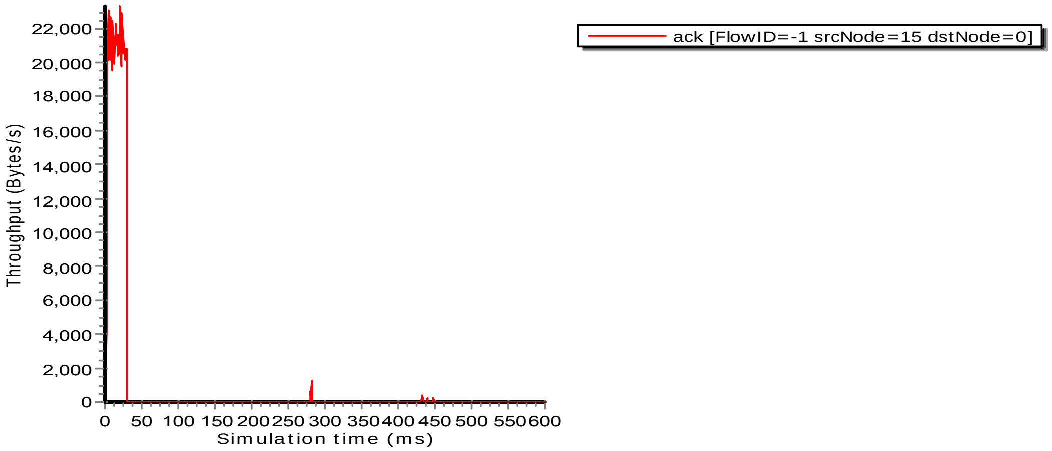


Figure 6.1 (a): Throughput transfer from node 15 to node 0 in DSDV

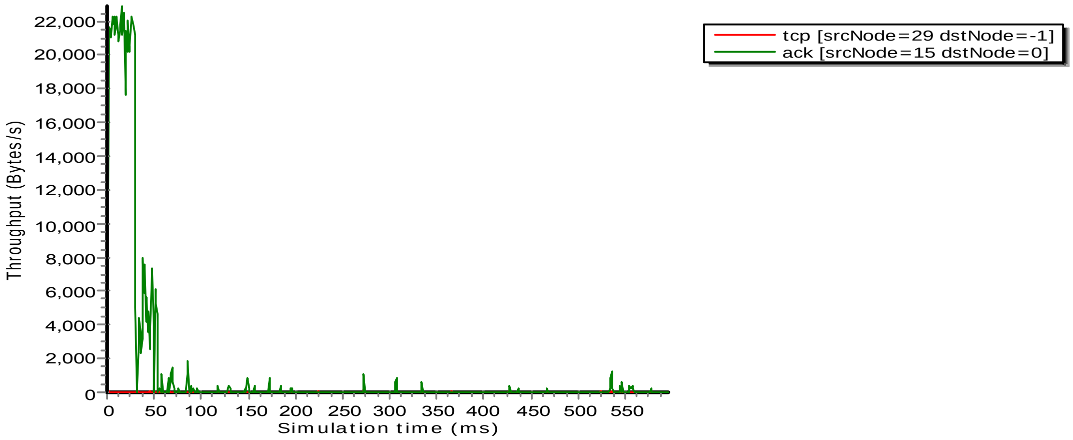


Figure 6.1 (b): Throughput transfer from node 15 to node 0 in AODV

For figure 6.1 (a) & 6.1 (b) 0s to 30s throughput curve was almost same for proactive (DSDV) and reactive (AODV). After 30s in reactive (AODV) throughput fluctuates for many times but in proactive (DSDV) throughput curve almost flat.

6.1.2 Jitter transferred:

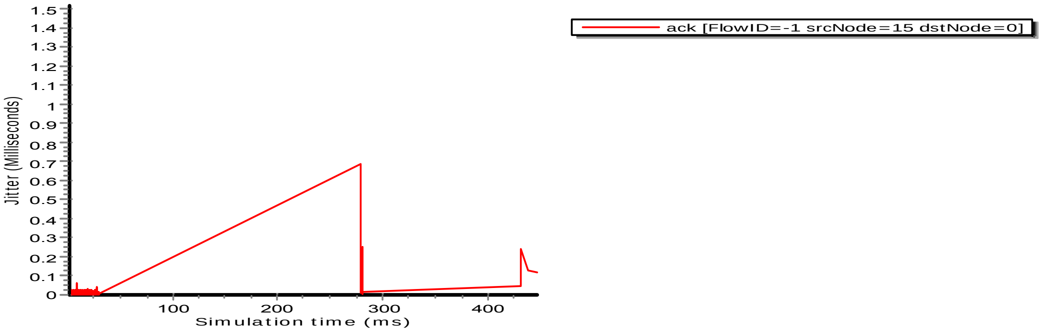


Figure 6.2 (a): Jitter transfer from node 15 to node 0 in DSDV

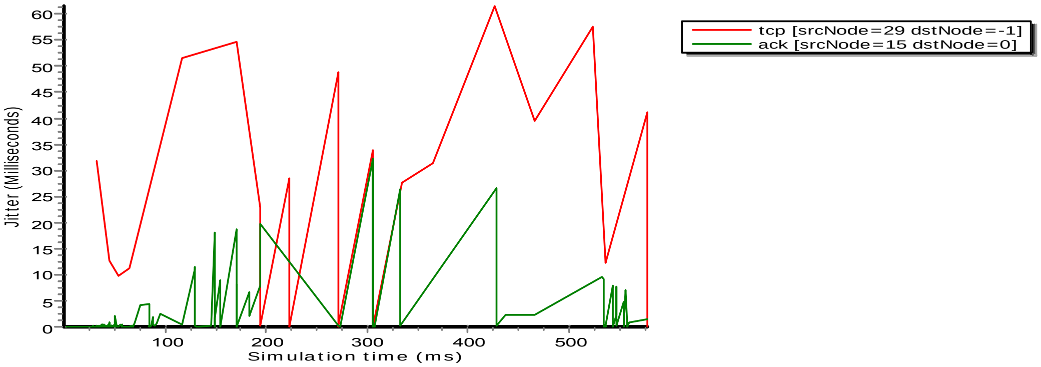


Figure 6.2 (b): Jitter transfer from node 15 to node 0 in AODV

For figure 6.2 (a) & 6.2 (b) jitter of proactive (DSDV) is less than reactive (AODV) and jitter fluctuates for several times in AODV. The maximum jitter is about 0.7 ms in proactive (DSDV) but the maximum jitter is about 35 ms in AODV. Overall the jitter of proactive (DSDV) is less than AODV.

6.1.3 Delay transferred:

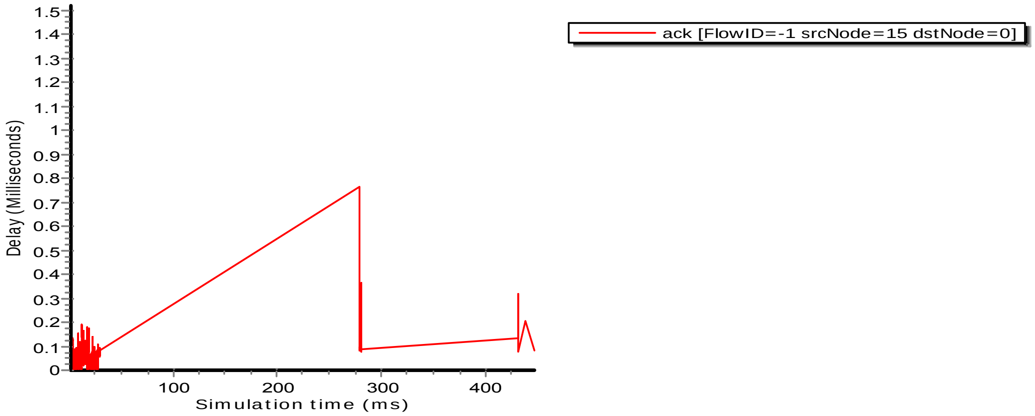


Figure 6.3 (a): Delay transfer from node 15 to node 0 in DSDV

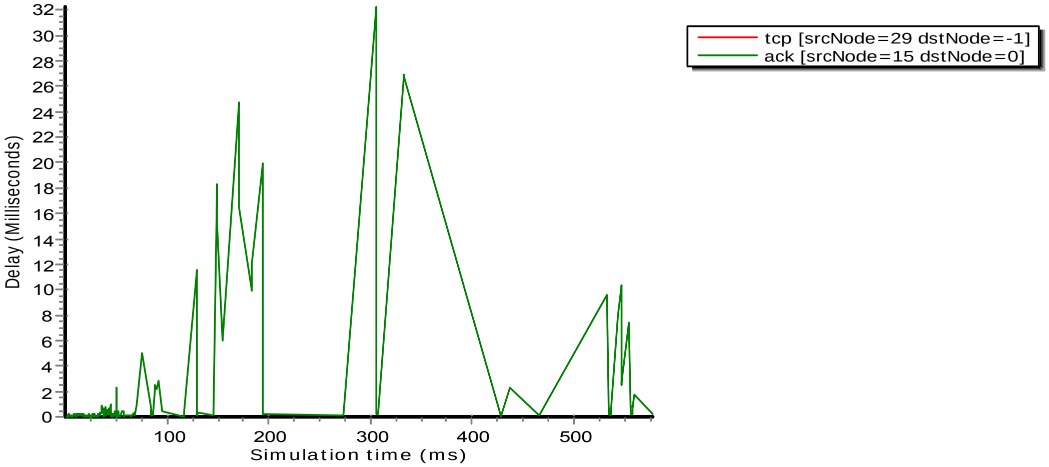


Figure 6.3 (b): Delay transfer from node 15 to node 0 in AODV

For figure 6.3 (a) & 6.3 (b) delay of proactive (DSDV) is less than AODV. In proactive (DSDV) maximum delay is 0.8 ms but in reactive (AODV) maximum delay is about 32 ms. We know the delay of reactive is more proactive because reactive topology use a route discovery packet to find the destination as a result it takes few moment to reach the destination. On the other hand, in proactive source know the destination position before routing happen.

6.1.4 Congestion window:

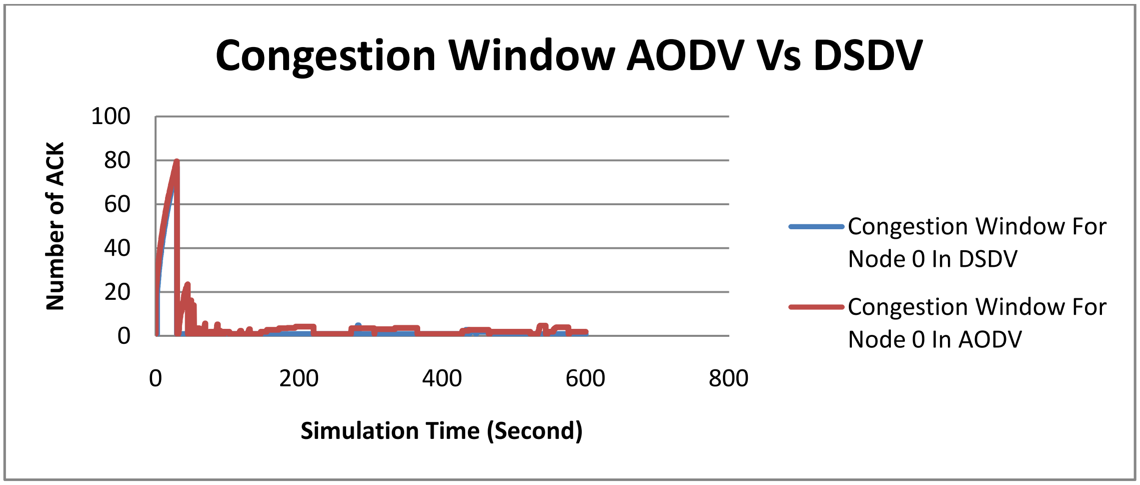


Figure 6.4 : Congestion window reactive (AODV) & proactive (DSDV) for node 0

We can see in the figure at the beginning both curve go in same way. After that the curve of Proactive (DSDV) is almost flat and the curve of reactive (AODV) rises for several times. Overall the number of getting ACK in reactive (AODV) is higher then proactive(DSDV).

6.1.5 Bandwidth in destination Node:

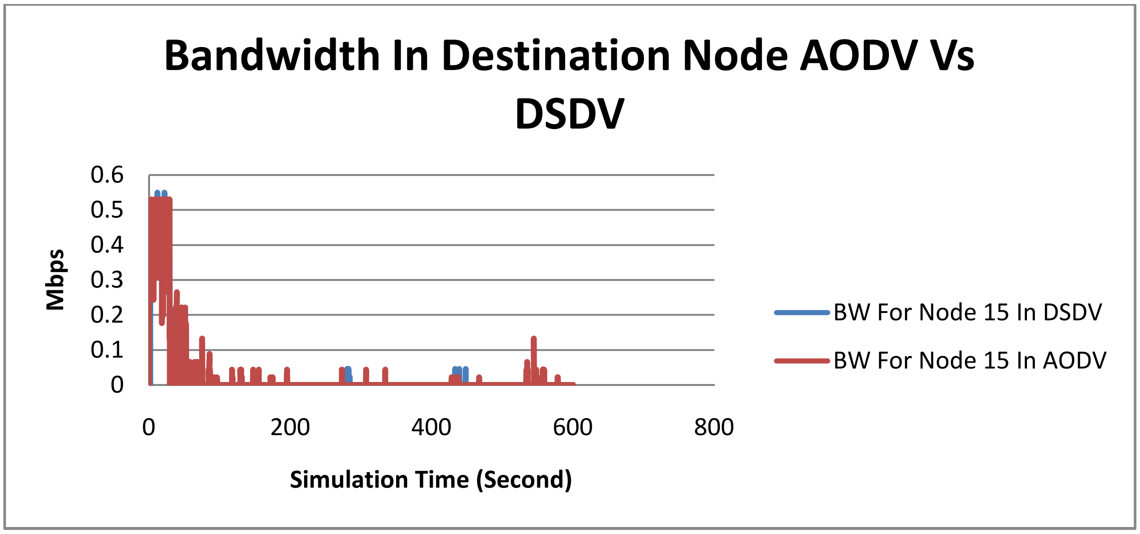


Figure 6.5 : Bandwidth in destination node reactive (AODV) & proactive (DSDV) for node 15

We can see in the figure the uses of bandwidth in proactive (DSDV) is less then reactive (AODV). This node was moving object as a result in proactive (DSDV) node left the neighbour nodes aria before update the routing table. For this reason it makes few data tranjection.

6.2 Node in short distance:

For Node 1(Source) and Node 25(Destination), here node 1 and node 25 are in 9 hop distance. Both are moving object and they start their journey at 10s node 1 moves to opposite direction from node 15 position at the speed of 12.66m/s and its go out of range at 24.36s. At the time of 153s node 28 and 5 comes to its range and connection reconnect.

In proactive (DSDV) at the time of 93.02s route happens according to the 1-28-25 direction but this packet has been lost because in the routing table of node 28 do not locate the right path for destination. At the time of 153s first successful route happens according to the 1-5-16-31-46-61-76-90-25 direction. In reactive (AODV) at the time of 153s first successful route happens according to the 1-28-5-16-32-33-48-63-77-76-90-25 direction.

6.2.1 Throughput transferred:

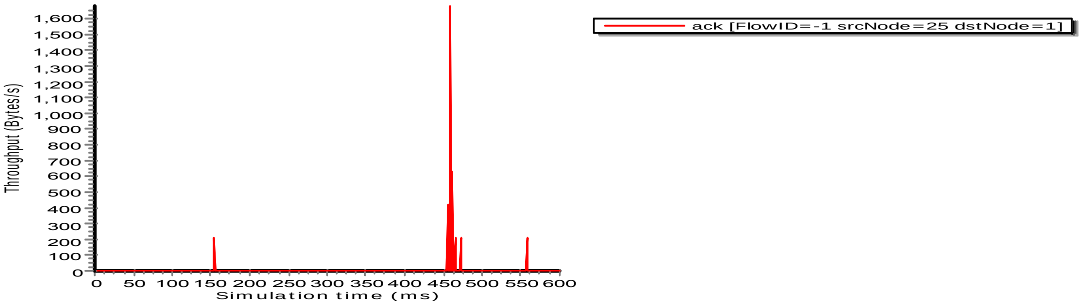


Figure 6.6 (a): Throughput transfer from node 25 to node 1 in DSDV

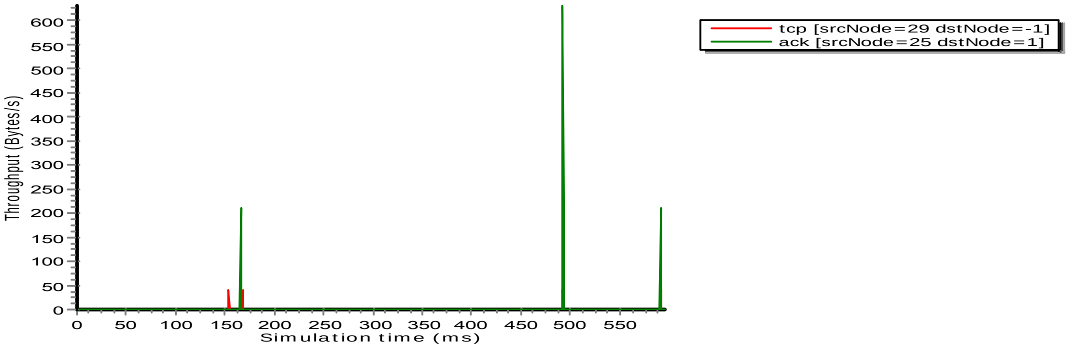


Figure 6.6 (b): Throughput transfer from node 25 to node 1 in AODV

For figure 6.6 (a) & 6.6 (b) throughput starts at form 153s first time the transaction of throughput in proactive (DSDV) is higher than reactive (AODV) but in reactive (AODV) the throughput curve rises after few second. For this TCP connection the throughput high for proactive and it's happen after stop moving the object.

6.2.2 Jitter transferred:

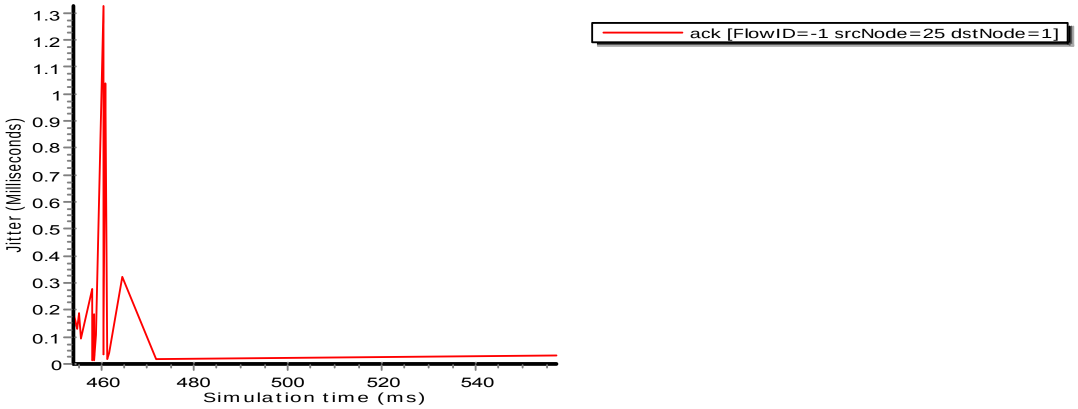


Figure 6.7 (a): Jitter transfer from node 25 to node 1 in DSDV

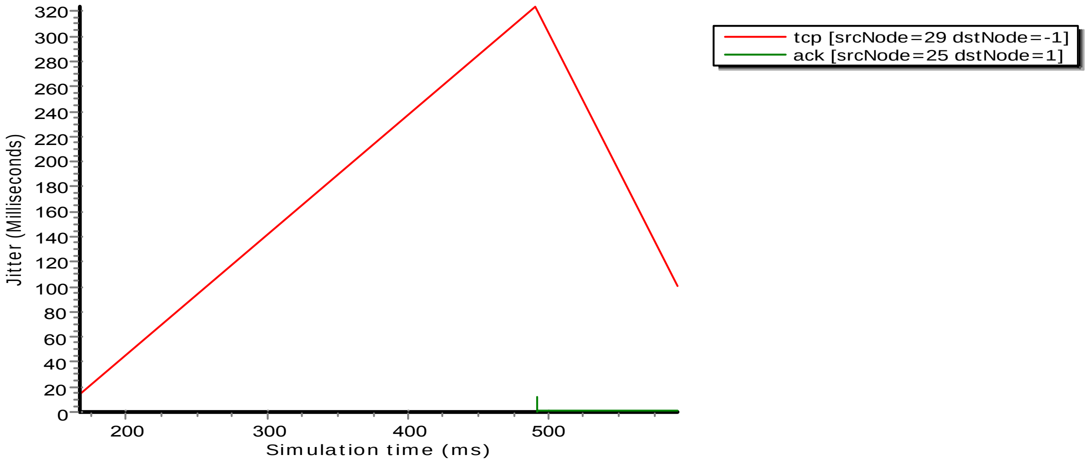


Figure 6.7 (b): Jitter transfer from node 25 to node 1 in AODV

For figure 6.7 (a) & 6.7 (b) jitter curve increase gradually in reactive (AODV). The minimum jitter of reactive (AODV) is higher than proactive (DSDV) jitter. Jitter represents the standard deviation of packet delay between nodes and object. Therefore, the standard deviation of this communication is high in reactive then proactive.

6.2.3 Delay transferred:

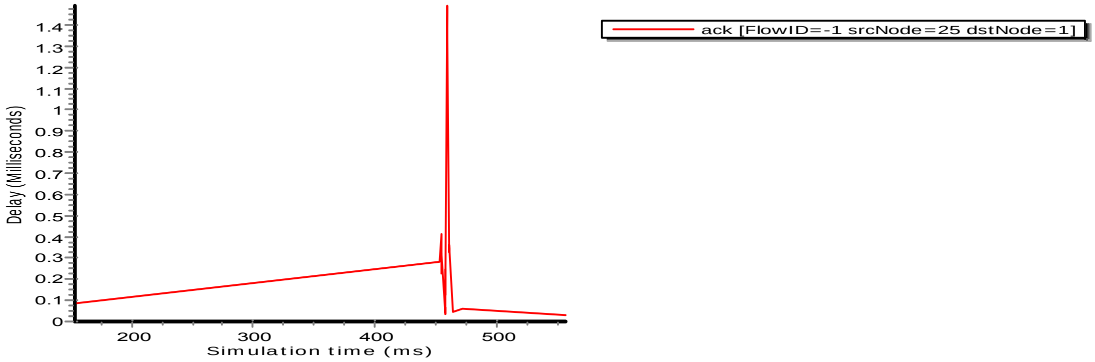


Figure 6.8 (a): Delay transfer from node 25 to node 1 in DSDV

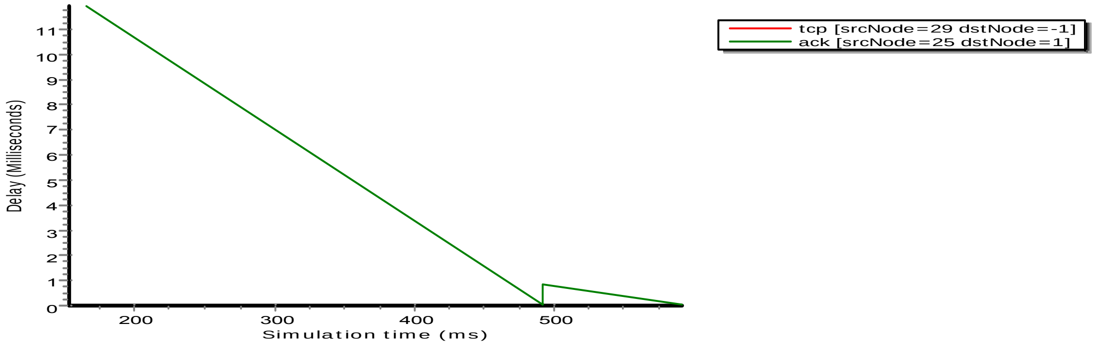


Figure 6.8 (b): Delay transfer from node 25 to node 1 in AODV

For figure 6.8 (a) & 6.8 (b) the delay of reactive (AODV) is higher than proactive (DSDV). Delay curve have been decrease with time in reactive (AODV). In proactive delay is very low.

6.2.4 Congestion window:

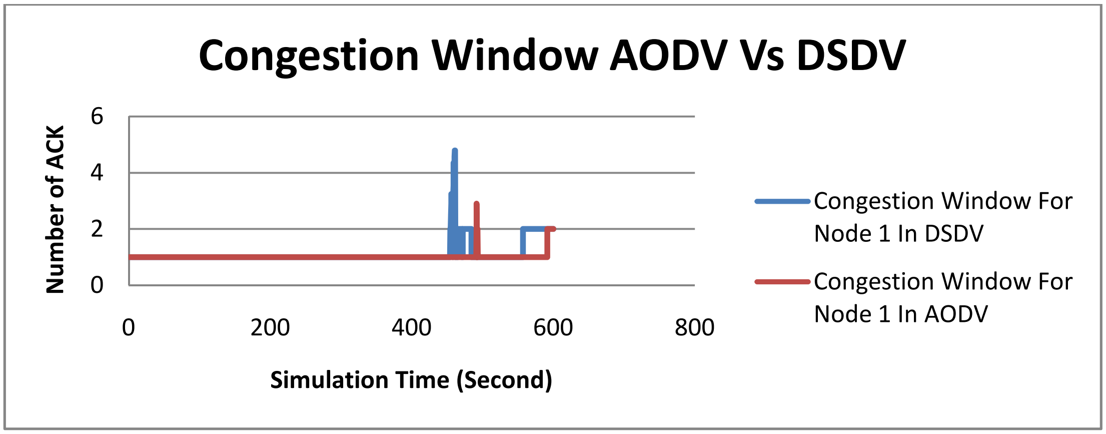


Figure 6.9 : Congestion window reactive (AODV) & proactive (DSDV) for node 1

In the figure number of getting ACK curve increase after a time interval. Before stop the node the communication was stop because the node was moving so fast. After stop moving the node was 5 hop next from source node. As a result proactive (DSDV) build connection before reactive (AODV). Reactive connect the communication after few moment. It also represents the delay of reactive communication.

6.2.5 Bandwidth in destination Node:

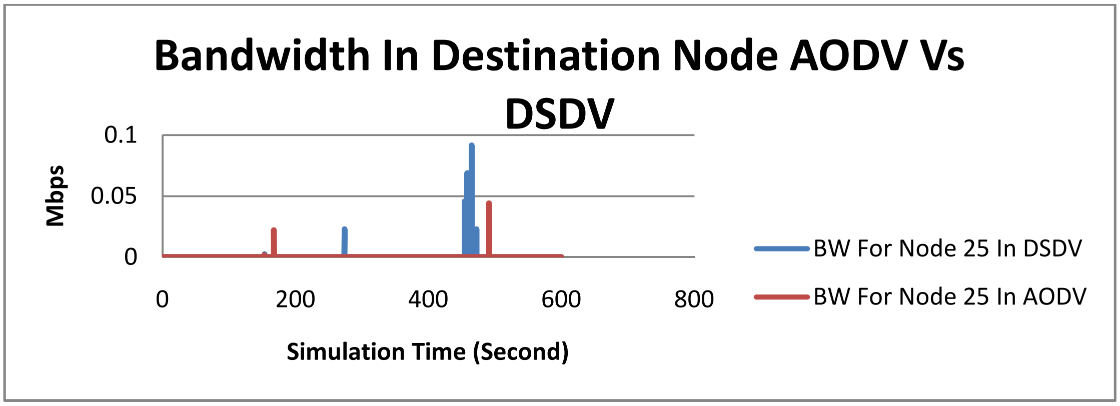


Figure 6.10 : Bandwidth in destination node reactive (AODV) & proactive (DSDV) for node 25

In the figure we can see the curve of reactive (AODV) rises fast then after about 100s proactive curve rises. When the destination node become closer to the source node proactive perform better then reactive. However, at the end of simulation proactive used more bandwidth then reactive. We can say that proactive perform well in short distance for mobile node or object.

6.3 Summary of analysis:

- Reactive is more efficient then proactive for this technology.
- Proactive is good for still object.
- Delay of reactive procedure is more than proactive.
- Reactive make unnecessary route.
- Sometimes reactive don't use the minimum path.
- The data loss of reactive is more than proactive.
- Reactive can't route properly in long distance.
- Proactive takes more time to update its table in long distance.
- In short distance proactive performs better then reactive.

- In short distance the throughput & congestion window of proactive is higher than reactive but in long distance throughput & congestion window of reactive higher then proactive.
- Both topologies don't work after a speed.
- Proactive or reactive is not suitable routing procedure on moving object

CHAPTER 7: OUR PROPOSED PROTOCOL

As VANET is related with moving object, so speed is the major concern for this technology. From the analysis described in previous chapter, we have realized that only proactive of reactive protocol is not enough to track a moving object. So we have decided to propose a hybrid protocol what is the combination of proactive and reactive protocol. The name of our proposed protocol is **speed based routing protocol (SBRP)**

7.1 Characteristics of our proposed protocol:

1. Every node will have both proactive and reactive features.
2. Every node will have routing table and it update its neighbor information according to the neighbor node moving speed.
3. Every node will contain up to 3 next hope's information into its routing table. As a result, every node will create a small network zone.
4. The update time interval and hold on timer will adjust after a time phase depending on node's speed. In fact, it can be periodic or non-periodic.
5. If a node crosses the speed limit it will get high update priority to its neighboring nodes.
6. Source will generate RD packet to find the destination.
7. If the destination is in source network zone then it will follow proactive procedure otherwise it will follow on demand distance vector routing.
8. In data communication, every node sends request to its adjacent node and every node hold the request information up to a defined time.
9. If the connection has dropped before completing the data communication, then the destination node will send a message to its adjacent node to reconnect the connection.

7.2 Algorithm of our proposed protocol:

PS = Previous Speed

MS = Minimum Speed

k = Constant value for stable speed

K = Constant value for time interval

L = Constant value for hold on timer

Proactive Part

```
IF Speed > MS Then
        IF Speed <= $PS - k$ OR Speed >= $PS + k$ Then
                Break Time Interval
                Break Hold On Timer
                SET Time Interval = $\frac{K \times General\ Time\ Interval}{Speed}$
                SET Hold On Timer = $\frac{L \times General\ Hold\ On\ Timer}{Speed}$
        END IF
ELSE
        Break Time Interval
        Break Hold On Timer
        SET Time Interval = General Update Time Interval
        SET Hold On Timer = General Hold On Timer
END IF

IF Speed > Speed Limit Then
        SET Priority = High =1
ELSE
        SET Priority = Low =0
END IF

IF Time Interval Over Then
        For i = Table.Began() To Table.End()
        IF Table[i][Hop Count] > Max Hop Then
            New Table = Table[i]
        END IF
    END FOR
    Broadcast RU(Node ,New Table, Priority)
END IF

Receiving RU (Sender, Table, Priority) From j
    IF Priority High Then
        CALL Update Table ( $j \rightarrow Table$ )
    ELSE
        CALL Update Buffer ( $j \rightarrow Table$ )
    END IF

    IF Hold On Timer Over Then
        Update Table Form Buffer
    END IF
```

Reactive part

```
Generate RD( )
    FOR i = Table.Began() To Table.End()
        IF Table[i][Node] == $N_D$  Then
            Interface = Table[i][int]
            In Zone = True =1
        END IF
    END FOR
    IF In Zone == True Then
        Forward DATA($N_S$ , $N_{D,}$ Interface, DATA)
    ELSE
        Broadcast RD($N_{S,}$ $N_{D,}$ TempPath)
    END IF
```

```
Receiving RD ( $N_S$, $N_{D,}$ TempPath ) From j
    FOR k = Table.Began() To Table.End()
        IF Table[k][Node] == $N_D$ Then
            If TempPath < PATH Then
                PATH = TempPath
                SEND RR ($N_I$, $N_{S,}$ $N_{F,}$ $N_D$, PATH)
            ELSE
                IF RD not same as before Then
                SEND RR ($N_I$, $N_{S,}$ $N_{F,}$ $N_D$, TempPath)
                END IF
            END IF
        Return
        END IF
    END FOR
    IF RD not same as before Then
        ADD $N_I$ To TempPath
        SET Timer = K × $Distance(Nj, Ni\ )$
    END IF
IF Time Out Then
    Broadcast RD ($N_S$, $N_{D,}$ TempPath )
END IF
Receiving RR($N_J$ ,$N_{S,}$$N_{F,}$ $N_D$, PATH ) Form j
    IF $N_I$ == $N_S$ Then
        Then Store The PATH
        Forward DATA($N_S$, $N_F$, $N_D$,PATH)
    ELSE
        Forward RR($N_I$ ,$N_{S,}$ $N_{F,}$ $N_D$, PATH )
    END IF
```

7.3 Route discovery (RD):

When source node wants to send data or information to a destination node then the reactive part of this topology generates a Route discovery (RD) packet to find the destination. RD packet header holds the source address, destination address and sequence number. The source and destination must be unique for accurate communication. As source don't know the position of destination so source needs to broadcast RD packet to its neighbor node and this broadcast will continue for specific times. The rebroadcast of RD will get priority according to the distance between two nodes. If a node receive a RD packet from other node with same source and destination and sequence number what previously received then the node will discard new RD packet.

7.4 Route replay (RR):

When RD find the destination then the node generate a RR packet and it copy the path form RD header and forward according to the path to source node. This route depends on connection path and every node include the current position of RR packet. The RR packet also carry the information of the node where the destination found. When data reached the node where destination found then it follow proactive procedure to get the destination.

7.5 Route updates (RU):

In this topology RU packets are use to update adjacent neighbor node information. Most of the topology uses a beacon message to update adjacent neighbor but we are don't using beacon. Here neighbor will provide its update according to its moving condition. As the max hop is defined for this topology so when any neighbor node sends its update information it will send its max hop-1 node information form table because the max hop adjacent information will be max hop+1 for other node.

7.6 Update time interval:

In proactive procedure every node sends its update information after a moment what is known as update interval. Most of the proactive procedure the update interval is periodic but for our topology it is not. As we are concern about speed and time of moving object so we have defined the update interval according to the node moving speed. The update interval will be inversely proportion to the speed.

Update interval = $\frac{K \times General\ update\ interval}{speed}$

7.7 Hold on timer:

Hold on timer use to make stable the routing update. It stabilizes routing information and helps preventing routing loops during periods when the topology is updating on new information. Generally hold on timer become fixed but here hold on timer also changeable and it depends on node moving speed. At the period of hold on time the route update information will store in buffer. And after expire of hold on timer the information update from buffer to node routing table.

Hold on timer = $\frac{L \times General\ hold\ on\ timer}{speed}$

7.8 Previous node speed:

As in our topology update interval depends on node moving speed so we need to define a speed range for update interval setting. If we change the update interval for every kilometer it may make unnecessary update interval and increase time complexity for a node. So we need define speed range when the update interval will be same. For this reason we need to calculate previous node speed and the stable speed will be previous speed ± k. If the node speed getter or less then speed ± k then the update interval will be change.

7.9 Speed limit:

Speed limit is to give priority for route update and it will depend on the mobility. If the node moving speed cross the speed limit then it will get high priority to update its information to its neighbor node otherwise the update information will update into the buffer.

7.10 Minimum speed:

Minimum speed is the speed when update time interval don't need to change form general time interval.

7.11 Communication types:

Two types of communication & routing will happen here. One is interior of a network zone and another is exterior of a network zone.

7.11.1 Interior communication:

If destination become in range of source node then it follow only proactive procedure to send data.

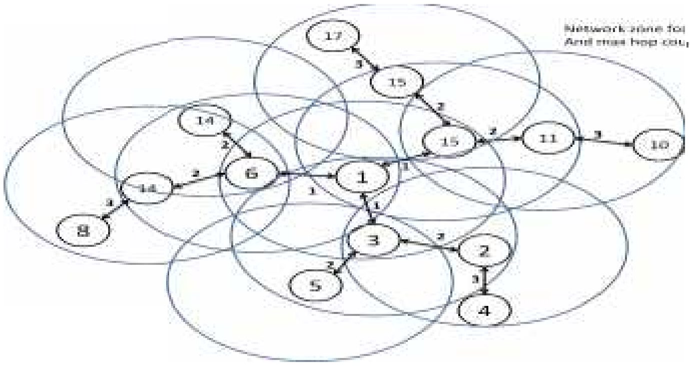
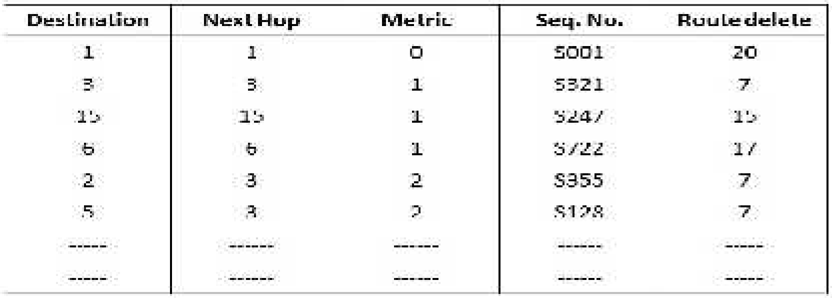


Figure 7.1: Network zone for node 1



Destination	Next Hop	Metric	Seq. No.	Route delete
1	1	0	S001	20
3	3	1	S321	7
15	15	1	S247	15
6	6	1	S722	17
2	3	2	S355	7
5	3	2	S128	7
-----	------	------	------	-----
-----	------	------	------	-----

Figure 7.2: Sample routing table for node 1

7.11.2 Exterior communication:

If the destination don't in the range of source in this circumstance source follow reactive procedure to find the destination node network zone. After getting it follow proactive and reactive procedure to send data.

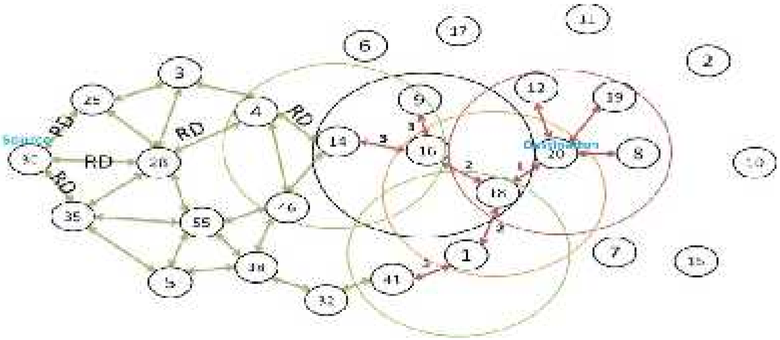


Figure 7.3: Exterior communication.

7.12 Comparison with proactive, reactive and our proposed topology:

Proactive Protocols

- A proactive protocol maintains routing tables for the entire network. As a result, a route is found as soon as it is requested.
- Routing is faseter then reactive
- Destination is known to source so its dont need to discover the node.
- The major advantage of a proactive protocol is its low latency in discovering new routes.
- Proactive generate a high volume of control messages required for updating local routing tables.
- Its not goor for moving object because of update time interval.

Reactive Protocols

- Route happends on demand so it don't create overhead message untill route happend.
- Destination is known to source so its dont need to discover the node.
- The main advantage of a reactive protocol is the low overhead of control messages.
- Latency of reactive protocol is higher then proactive.

A proposed combined protocol

- The proactive and reactive topology work parallel together so it will faster then only proactive or reactive topology.
- Here every node maintain routing table for a smoll network zone as a result it reduce volume of control message.
- If a source outside of destination routing zone the route discovery packet don't need to reach the destination node to build the connection it only needs to reach destination network zone aria.
- If the radius of the node routing zone is k, each node in the zone can be reached within k hops from Source.
- The minimum distance of a peripheral node from S is k (the radius).
- This reduces latency in route discovery and reduces the number of control messages as well.

CHAPTER 8: CONCLUSION

8.1 Summery

The main goal of this research is to study the existing routing protocols for ad-hoc network system and to develop a more efficient routing protocol for VANET.

We have studied different types of routing protocols (e.g. topology based, position based, cluster based, geo-cast based and broadcast based). We have also simulated and compared AODV (Reactive) and DSDV (Proactive) to find out their efficiency and detect their flaws. We have compared the jitter, delay, throughput, and bandwidth of these two protocols. From this analysis we found out that the proactive routing is very much efficient when the nodes in the network are stable. But in VANET the nodes are dynamic. As a result the proactive protocol is not suitable in these circumstances. In other hand the reactive protocol has shown somewhat more promising results, still need of broadcasting & the amount of delay is very much high.

Seeing these results we have developed a new protocol for VANET based on the hybrid (Reactive + Proactive) protocol because hybrid protocol can overcome the issues of the reactive and proactive protocol and get a better result. In or proposed protocol the proactive and reactive part works parallelly to minimize the broadcasting & delay and get higher access.

In our protocol we have considered the speed to be the most vital issue as the routing is done between highly mobile vehicles. As vehicles move at different speed so we can't measure all the vehicles with the same scale. So in the proposed protocol the speed will determine the update pattern for a node. So with the changing speed of a node its hold-on timer & time interval will be altered.

This paper consists of the different routing protocols we have studied so far, the comparison data between AODV & DSDV. It also consists of the protocol we have proposed for VANET and its algorithm.

8.2 Future work

After conducting the research our interest in VANET has increased. VANET is a technology that can bring a solution to our everyday traffic problem. It will enhance the traffic stability and help minimize the losses we traffic accidents bring to our life. We plan to continue our research and contribute in the development of VANET. We plan to

- Study the VANET to its further depths.
- Discover the possibilities of VANET.
- Overcome the limitations we faced with simulating the protocol.
- Simulate our proposed protocol.

8.3 Conclusion

In ad-hoc network community VANET has fascinated many researchers due to its distinctive nature. A huge amount of research has been made in various routing sectors of VANET; still there are many areas that need more research. In our research we have developed a new routing protocol but due to the limitation of parameters we still haven't simulated our protocol. In future we want to continue our research to learn more about the possibilities that lies within VANET.

References

[1] Vehicular Ad-hoc Network, http://www.wikipedia.org/wiki/VANET Last access on 23June 2013

[2] Google image, http://www.google.com.bd/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CDUQFjAB&url=http%3A%2F%2Fwww.cs.sunysb.edu%2F~jgao%2FCSE370-spring06%2Flecture10.pdf&ei=KaRhUfazK4SyrAfw0YDgDw&usg=AFQjCNEqgH87aqJCT1Pmc9HPbqNuLqrPMA&bvm=bv.44770516,d.bmk Last access on 23February 2013

[3] Google imagehttps://www.google.com.bd/imghp?hl=en&tab=ii Last access on 23June 2013

[4] Ad Hoc and Sensor Networks, Copyright © 2003, Dr. Dharma P. Agrawal and Dr. Qing-AnZeng.

[5] Wireless Routing Protocol ShivashisSaha, by Elizabeth M. Royer, Chai-KeongToh, S. Murthy and J. J. Garcia-Luna-Aceves

[6] Scalable Routing Protocols for Mobile Ad Hoc Networks by Xiaoyan Hong, KaixinXu, and Mario Gerla at UCLA

[7] Optimized Link State Routing for mobile ad hoc networks by Philippe Jacquet, INRIA, France

[8] CATEGORIZATION OF ROUTING PROTOCOLS IN VEHICULAR ADHOC NETWORK FOR VEHICLE TO VEHICLE COMMUNICATION by Pinak M. Popat, PG Student (Department of Computer Engineering) Marwadi College; Pooja A. Vaishnav, Bhumi K. Padodara, Ankita M. Parmar, PG Student (Department of Computer Engineering) VVP Engineering College, GTU, Rajkot, Gujarat.

[9] Fisheye State Routing (FSR) by G. Pei, M. Gerla, Tsu-Wei Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *IEEE ICC 2000*, vol. 1, pp. 70 -74.

[10] The Zone Routing Protocol (ZRP) by Dr. R. B. Patel

[11] Zone Routing Protocol (ZRP) by Ikhsan Putra Kurniawan Sun Moon University

[12] Enhanced Perimeter Routing for Geographic Forwarding Protocols in Urban Vehicular Scenarios by Kevin C. Lee, JérômeHärri, Uichin Lee, Mario Gerla

[13] http://www.google.com.bd/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CC4QFjAB&url=http%3A%2F%2Fwww.radford.edu%2F~nsrl%2Fcreu0809%2FPresentations%2FGPSR.ppt&ei=ck-9UeL0C8fqrAf-gYGIAg&usg=AFQjCNHTr_zukYWb70-Z4TCvpyaGol8ReQ&bvm=bv.47883778,d.bmk&cad=rja

[14] WSN presentation ,http://www.google.com.bd/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCoQFjAA&url=http%3A%2F%2Fhscc.cs.nthu.edu.tw%2F~sheujp%2Flecture_note%2F13wsn%2Fppt%2FWSN_Chapter%25204%2520Routing%2520Protocols-I.pptx&ei=RlC9UfCjM8TtrAeBzoD4Ag&usg=AFQjCNESYRNR8CxdxLzS6JoibEYaP6WP7g&bvm=bv.47883778,d.bmk&cad=rja

[15] An Intelligent Routing Protocol for Delay Tolerant Networks using Genetic Algorithm bySaeidAkhavanBitaghsir.

[16] Comprehensive Study of Routing Protocols and Power Saving in Cluster Based AODV-ERS Protocol by Dr.S.K.Shriwastava, Director,SBITM,Betul; MamtaSood, TIT COLLEGE, Bhopal, India.

[17] Border-node based MovementAware Routing Protocol by Monika, Sanjay Batish&Amardeep Singh Dept. of Comp. Science, PEC University of Technology, Chandigarh, India

[18] Performance of Modified Edge Based Greedy Routing Algorithm in VANET Using Real City Scenario Ravi Shankar Shukla Associate Professor & Head CSE & IT Department ,Irfan Ali Khan Department of Computer Science & Engineering and NeerajTyagi Department of Computer Science & Engineering, MNNIT, Allahabad, India.

[19] Efficient Packet Forwarding Approach in Vehicular Ad Hoc Networks Using EBGR Algorithm by K.Prasanth, Research Scholar, Department of Information; Dr. K. Duraiswamy Dean Academic, Department of Computer Science and Dr. C. Chandrasekar, Research Scholar, Department of Computer Applications K.S.Rangasamy College of Technology, TiruchengodeTamilnadu, India.

[20] Modeling and Analysis of the Associative Based Routing (ABR) Protocol by Using Coloured Petri Nets, Rahul Bhargava, Deptt. of Computer Science and Engineering; RekhaKaushik (P.hd.) Deptt. of Computer Science and Engineering M.A.N.I.T. Bhopal, India.

[21] Analysis of Enhanced Associativity Based Routing Protocol Said Abu Shaar, Fawaz A. M. Masoud, AymanMurad, Riyad Al-Shalabi and GhassanKanaan Amman University, Jordan The University of Jordan, King Abdullah II School for Information Technology, Computer Information Systems Department, Jordan Arab Academy for finance & Banking Sciences, Jordan Amman Al-Ahliyya University, Jordan

[22] MORA: a Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks 許子衡教授, 董藝興

[23] MORA: a Movement-Based Routing Algorithm for Vehicle Ad Hoc Networks FabrizioGranelli, Senior Member, Giulia Boato, Member, and DzmitryKliazovich, Student Member

[24] Contention Based Routing in Mobile Ad Hoc Networks with Multiple Copies Ms. E. JenefaJebaJothi , Dr. V. Kavitha, Ms. T. Kavitha.

[25] Location Privacy Protection in Contention Based Forwarding for VANETs Qing Yang Alvin Lim XiaojunRuan and Xiao Qin Computer Science and Software Engineering Auburn University, Auburn, AL, USA.

[26] TO-GO: TOpology-assist Geo-Opportunistic Routing in Urban Vehicular Grids Kevin C. Lee, Uichin Lee, Mario Gerla University of California Department of Computer Science Los Angeles, CA.

[27] VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks Zhao, J.; Cao, G. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, 2008.

[28] The Cluster-Based Routing Protocol Tim Daniel Hollerung University of Paderborn, project group ‘Mobile Ad-Hoc Networks Based on Wireless LAN.

[29] Hierarchical Cluster-based Routing in Wireless Sensor Networks SajidHussain and Abdul W. MatinJodrey School of Computer Science, Acadia University Wolfville, Nova Scotia, Canada

[30] A Reactive Location Routing Algorithm with Cluster-Based Flooding for Inter-Vehicle Communication Raúl Aquino S. and Arthur Edwards B. Facultad de Telemática Universidad de Colima.

[31] A Novel Routing Algorithm for Wireless Ad-Hoc Networks R. A. Santos , O. Alvarez, A. Edwards, A. González, M. García, A. Galaviz, M. Cosio, A. MartínezSchool of Telematics, Universidad de Colima, Av. Universidad 333, School of Mechanical, Electrical and Electronic Engineering, Universidad de Colima, Coquimatlan, Colima

[32] Dynamic Time-stable Geocast Routing in Vehicular Ad Hoc Networks by HamidrezaRahbar

[33] Geocast Routing Protocols for VANETs: Survey and Geometry-Driven Scheme Proposal SalimAllal, SaadiBoudjitUniversit´ e Paris 13, Sorbonne Paris Cit´e, Laboratoire de Traitement et Transport de l’Information (L2TI), (EA 3043), 99, Avenue Jean-Baptiste Cl ´ ement, F-93430, Villetaneuse, France.

[34] Study of Various Routing Protocols in VANET Uma Nagaraj, Dr. M. U. Kharat, PoonamDhamal, Dept. of Computer Engg., M.A.E., Pune, India, Dept. of Computer Engg., M.E.T., Nashik , India.

[35] Broadcasting Routing Protocols in VANET Uma Nagaraj, PoonamDhamal Pune University, Alandi, Pune India.

[36] Broadcasting in VANET OzanTonguz, NawapornWisitpongphan, FanBai , PriyanthaMudalige, and VarshaSadekar Carnegie Mellon University, ECE Dept., Pittsburgh, PA 15213-3890, USA General Motors Corporation, ECI Lab, Warren, MI 48092, USA

[37] VTC2007 - Reliable and efficient Information dissemination in Intermittently Connected Vehicular Adhoc Networks by J. L. Chiang.

[38] VANET Routing Protocols: Pros and Cons, by Bijan Paul and Md. Ibrahim Dept. of Computer Science & Engineering, Shahjalal University of Science & Technology, Sylhet, Bangladesh

[39] Syntax of tcl language, http://rigaux.org/language-study/syntax-across-languages-per-language/Tcl.htmlLast access on 13February 2013

[40] Syntax of tcl languagehttp://tmml.sourceforge.net/doc/tcl/incr.html Last access on 13February 2013

[41] Syntax and expression of tcl languagehttp://www.astro.princeton.edu/~rhl/Tcl-Tk_docs/tcl/expr.n.htmlLast access on 13February 2013

[42] Control and loop statement in tcl language, http://www.bin-co.com/tcl/tutorial/for_loop.phpLast access on 30January 2013

[43] QualNet exercises , http://degas.cis.udel.edu/QualNet/Last access on 10November 2012

[44] Download QualNet, http://www.findthatzipfile.com/search-53334998-fZIP/winrar-winzip-download-qualnet-4-0-university-wireless-mme-tar.htmLast access on 10November 2012

[45] Tcl language, http://en.wikipedia.org/wiki/TclLast access on 30May 2013

[46] Tcl basic, http://www.cse.scitech.ac.uk/ccg/software/chemshell/manual/tclbasics.htmlLast access on 3June 2013

[47] Tutorial for Network simulator, http://www.isi.edu/nsnam/ns/tutorial/index.htmlLast access on 4June 2013

[48] NS2 installation in cygwin, http://www.scribd.com/doc/42038608/Cygwin-Ns2-Complete-Installation-GuideLast access on 27December 2012

[49] Introduction to network simulator NS2,http://www.ns2ultimate.com/post/441093095/ns-2-35-works-on-cygwinLast access on 2January 2013

[50] Install ns2 in Ubuntu 10.4, http://tech-nikk.blogspot.com/2011/03/install-ns2-ns-allinone-235-rc7-in.htmlLast access on 30December 2012

[51] Security Issues and Challenges of Vehicular Ad Hoc Networks (VANET) by Ghassan Samara#1, Wafaa A.H. Al-Salihy*2, R. Sures#3 #National Advanced IPv6 Center, UniversitiSains Malaysia Penang, Malaysia

[52] Challenges in Securing Vehicular Networks by Bryan Parno and Adrian Perrig November 2005

[53] Authentication , http://en.wikipedia.org/wiki/Authentication Last access on 15 June 2013

[54] Integrity, http://en.wikipedia.org/wiki/Integrity Last access on 15 June 2013

[55] Integrity, http://msdn.microsoft.com/en-us/library/vstudio/k62k71x0%28v=vs.100%29.aspx Last access on 15 June 2013

[56] Confidentiality, http://en.wikipedia.org/wiki/Confidentiality Last access on 11 June 2013

[57] Confidentiality,http://msdn.microsoft.com/en-us/library/ff650720.aspx Last access on 11 June 2013

[58] What do we mean by Entity Authentication? By Dieter Gollmann, Department of Computer Science, Royal Holloway, University of London Egham, Surrey TW20 OEX, United Kingdom.

[59] Access control, https://en.wikipedia.org/wiki/Access_control Last access on 15 June 2013

[60] Privacy, https://en.wikipedia.org/wiki/Privacy Last access on 14 June 2013

[61] Non-repudiation, http://en.wikipedia.org/wiki/Non-repudiation Last access on 14 June 2013

[62] Non-repudiation,https://www.nics.uma.es/research/nonrepudiation Last access on 14 June 2013

[63] Information security,http://en.wikipedia.org/wiki/Information_security Last access on 13 June 2013

[64] Access control, http://www.oucs.ox.ac.uk/network/security/ISBP/toolkit/index.xml?ID=accesscontrol Last access on 15 June 2013

[65] VanetMobiSim download, http://vanet.eurecom.fr/ Last access on 24November 2012

[66] Vn Simulator, http://cipsm.hpc.pub.ro/vanet/vnsim.html Last access on 7November 2012

[67] Waypoint, http://en.wikipedia.org/wiki/Waypoint Last access on 11January 2013

[68] Design Routing Protocol Performance Comparison in NS2 By Yinfei Pan, Department of Computer Science SUNY Binghamton Vestal Parkway East, Vestal, NY 13850

[69] International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2012

[70] A Performance Analysis of TORA, AODV and DSR Routing Protocols in MANET using NS2 by G.Pragadeeswaran, D.Ezhilarasi, P.Selvakumar

[71] Bottleneck, http://en.wikipedia.org/wiki/BottleneckLast access on 7November 2012

[72] Presentation on TORA, http://www.ietf.org/proceedings/40/slides/manet-tora/sld004.htmLast access on 17June 2013

[73] Presentation on DSDV, http://www.google.com.bd/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CDAQFjAB&url=http%3A%2F%2Fcone.informatik.uni-freiburg.de%2Fteaching%2Fvorlesung%2Fmanet-s07%2Fexercises%2FDSDV.ppt&ei=J4fJUZSZJc2HrAef7oH4Ag&usg=AFQjCNH82Qz61L2JUs8f_5MdG0j1rKn7hw&bvm=bv.48293060,d.bmk&cad=rjaLast access on 21June 2013