International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 5, May - 2013

Reducing Routing Overhead Using Packet Dropper In Dense VANET

Mr. Renish Ratanpara
Post Graduate Student
Computer Engineering Department
Sri Balaji College Of Engineering And Technology, Jaipur

Prof. Mrs. Saroj Hiranwal
Assistant Professor
Computer Engineering Department
Sri Balaji College Of Engineering And Technology, Jaipur

Mr. Chetan Ajudiya
Post Graduate Student
Computer Engineering Department
Noble Engineering College, Junagadh

Abstract

Vehicle Ad-hoc Networks (VANETs) are systems that allow cars to talk to each other. Wireless device sends information to nearby cars, and messages can be routed from car to car so that the information can spread through the city.

A variety of routing protocols are being experimented with both existing wireless protocol systems and new systems that are being developed expressly for this new application. Each day number of cars are increasing due to that number of broadcasting packets are increasing, this case gives new challenge of routing over head in network.

To address the above mentioned issue, we have used functionality of packet dropper nodes. Packet dropper nodes are nodes that drop routing packets intentionally to reduce routing overhead. We have evaluated AODV routing protocol for proposed system and found to reduce routing overhead significantly, from AODV Protocol.

1. Introduction

Vehicular Ad Hoc Network (VANET) is a new challenging network environment that pursues the concept of ubiquitous computing for future. Vehicles equipped with wireless communication technologies and acting like computer nodes will be on the road soon and this will revolutionize the concept of travelling. VANETs bring lots of possibilities for new range of applications which will not only make the travel safer but fun as well. Reaching to a destination or getting help would be much easier. The concept of VANETs is quite simple: by incorporating the wireless communication and data sharing capabilities, the vehicles can be turned into a network providing similar services like the ones with which we are used to in our offices or homes. For the wide spread and ubiquitous use of VANETs, a number of technical challenges exist [1].

In this paper, we have addressed the problem is the using flooding to propagate a broadcast message throughout the network. The "broadcast storm problem" refers to the problem associated with flooding. First flooding results in a large number of duplicate packets being sent in the network. Second, a high amount of contention will take place, because nodes in close proximate of each other will try to rebroadcast the message. Third, collisions are likely to occur because the RTS/CTS are not applicable for broadcast messages.

The remainder of this paper is organized as follows. Section II present related work to address mentioned problem. Section III outlines the basic components for simulation. Section IV discusses proposed approach. Section V summarizes the main findings of our study.

2. Related Work

A literature survey of the existing work regarding the comparison of ad hoc routing protocols and mobility model in context of VANET has been carried out.

A. Routing Protocol In VANET

Because of the dynamic nature of the mobile nodes in the network, finding and maintaining routes is very challenging in VANETs. Routing in VANETs (withpure ad hoc architectures) has been studied recently and many different protocols were proposed [3][4].

1. Ad hoc On demand Distance Vector (AODV)

AODV is a well-known distance vector routing protocol and works as follows. Whenever a node wants to start communication with another node, it looks for an available path to the destination node, in its local routing table. If there is no path available, then it broadcasts a route request (RREQ) message to its neighborhood. Any node that receives this message looks for a path leading to the destination node. If there is no path then, it re-broadcasts the RREQ message and sets up a path leading to RREQ originating node. This helps in establishing the end to end path when the same node receives route reply (RREP) message. Every node follows this process until this RREQ message reaches to a node which has a valid path to the destination node or RREQ message reaches to the destination node itself. Either way the RREQ receiving node will send a RREP to the sender of RREQ message. In this way, the RREP message arrives at the source node, which originally issued RREQ message.

2. Optimized Link State Routing (OLSR)

OLSR is an optimization of the classical link state algorithm adapted for the use in wireless ad hoc networks. In OLSR, three levels of optimization are achieved. First, few nodes are selected as Multipoint Relays (MPRs) to broadcast the messages during the flooding process. This is in contrast to what is done in classical flooding mechanism, where every node broadcasts the messages and generates too much overhead traffic. Second level of optimization is achieved by using only MPRs to generate link state information. This results in minimizing the “number” of control messages flooded in the network. As a final level of optimization, an MPR can chose to report only links between itself and those nodes which have selected it as their MPR. This results in the distribution of partial link state information in the network.

3. Geographic Source Routing

Earlier GSR was used in MANET. Then it was improved to use in VANET scenario by incorporating in to it greedy forwarding of messages toward the destination. If at any hop there are no nodes in the direction of destination then GPSR utilizes a recovery strategy known as perimeter mode. The perimeter mode has two components one is distributed planarization algorithm that makes local conversion of connectivity graph into planar graph by removing redundant edges. Second component is online routing algorithm that operates on planer graphs.

4. Urban Multihop Broadcast protocol

UMB is designed to overcome the interference, packet collision and hidden node problems during message distribution in multi hop broadcast. In UMB the sender node tries to select the furthest node in the broadcast direction for forwarding and acknowledging the packet without any prior topology information. UMB protocol performs with much success at higher packet loads and vehicle traffic densities [7].

B. Mobility Model in VANET

Mobility pattern of nodes in a VANET can significantly influence route discovery, maintenance, reconstruction, consistency and caching mechanism and this can obviously affect data dissemination protocols. [6]

1. Freeway Mobility Model (FMM): Freeway is a generated-map -based model. The simulation area, represented by a generated map, includes many freeways, each side of which is composed of many lanes as shown in the Fig.1.

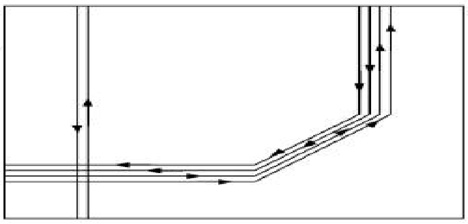

Figure 1- Freeway Mobility Model

No urban routes, thus no intersections are considered in this model. At the beginning of the simulation, the nodes are randomly placed in the lanes, and move using history-based speeds [2].

2. **Manhattan Mobility Model (MMM):** This is also a generated-map-based model to simulate an urban environment. Before starting a simulation, a map containing vertical and horizontal roads is generated as shown in the Fig 2.

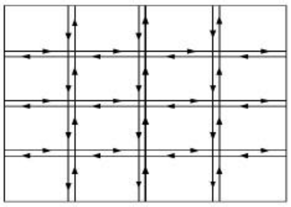

Figure 2- Manhattan Mobility Model

Each of these latter includes two lanes, allowing the motion in the two directions (north/south for the vertical roads and east/west for the horizontal ones). At the beginning of a simulation, vehicles are randomly put on the roads. They then move continuously according to history based speeds (following the same formula like the freeway model). When reaching a crossroads, the vehicle randomly chooses a direction to follow. That is, continuing straightforward, turning left, or turning right [2].

3. **Random-Way point Mobility Model:** The node speed in the Random Waypoint model is uniformly distributed between 9 and 16 m/s covering the speed limits of the different road categories present in the simulation scenarios. The pause time between subsequent trips is uniformly distributed between 0 and 10 seconds to simulate a short stop at the destination. The initial values for a node's position and speed are set according to the steady-state initialization method.

3. Simulator & Tools

The most reliable and authenticated tools used and preferred by most of the researchers for these kinds of simulations are: NS-2 and for or vehicular movements on roads, another particular tool and its extendable variant "SUMO and MOVE" the helping tool for traffic mobility patterns generation for network simulator is used.

1. Network Simulator (NS-2.34)

Ns-2 is an open source discrete event simulator used by the research community for research in networking. It has support for both wired and wireless networks and can simulate several network protocols such as TCP, UDP, multicast routing, etc. More recently, support has been added for simulation of large satellite and ad hoc wireless networks.

2. Simulation of Urban MObility (SUMO)

There are many tools available for microscopic simulation of road traffic like FARSI and VISSIM, but a popular road traffic simulation tool among the research community is Simulation of Urban MObility. SUMO is an open source, portable microscopic road traffic simulator. It allows the user to build a customized road topology, in addition to the import of different readymade map formats of many cities and towns of the world. The later feature helps in generating real world road topology. [8] SUMO also supports feature of microscopic simulation model, like imposing speed limits, defining number of lanes, junctions and traffic lights etc. It is also possible to define vehicles with specific properties like vehicle length, its maximum speed and its acceleration and deceleration properties. SUMO also provides the option to assign user defined as well as random routes to the vehicles. There is also an option available to model public transport system, where every vehicle arrives and leaves according to a timetable.

3. MObility model generator for VEhicular networks (MOVE)

MOVE is a Java-based application built on SUMO (Simulation of Urban Mobility) with a facility of GUI. MOVE comes along with a very good visualization tool and focuses mainly on traffic level. MOVE solves the problem of SUMO complex configuration with just few mouse clicks without worrying about the internal details of the simulator. [9]

MOVE can facilitate simulation by generating mobility traces from the TIGER database or Google earth. In addition to that, it also supports custom graphs defined by user and random generated graphs. But with random generated graphs, it restricts the node movement to grid i.e. the node should only move on the grid. MOVE uses parser to extract topological maps from above mentioned tools. MOVE utilizes the federated approach, in which they both communicate via parser. The traces from the traffic simulators is sent to parser for the translation and then processed by network simulator. The updated file from network simulator is passed to traffic simulator through parser. The problem rose with this approach was the interactions between the two simulators were not held in timely manner.

4. Proposed Work

In AODV, whenever a node wants to start communication with another node, it looks for an available path to the destination node, in its local routing table. If there is no path available, then it broadcasts a route request (RREQ) message to its neighborhood. Any node that receives this message looks for a path leading to the destination node. If there is no path then, it re-broadcasts the RREQ message and sets up a path leading to RREQ originating node. This helps in establishing the end to end path when the same node receives route reply (RREP) message. Every node follows this process until this RREQ message reaches to a node which has a valid path to the destination node or RREQ message reaches to the destination node itself. Either way the RREQ receiving node will send a RREP to the sender of RREQ message. In this way, the RREP message arrives at the source node, which originally issued RREQ message.

In dense Vehicular Ad hoc networks, the routing overhead increases manifolds because of the control packets traffic increasing at a fast pace. A strategy which can help in reducing the routing overhead is to let a certain percentage of nodes in the dense network behave selfishly. Selfish nodes are defined as nodes that maximize their own gain without regard to the welfare of other nodes. They do not aim to harm the network. The selfish behavior which is incorporated in our case is that the selfish nodes will not forward any RREQ or RREP packets. [16].

The reason for incorporating a small number of such nodes in the system is that for eg.,let 20 nodes out of 100 nodes behave selfishly that is they will not forward RREQ packets. But the route formation process will not be hampered. This is because in a dense network with many connections, the packets not forwarded by a selfish node will still be flooded to other nodes by neighbors of the selfish node. So, indirectly our selfish node is not only saving energy but also leading to less control traffic due to lesser amount of flooding. Consequently even the throughput increases up to a certain level. The route formation process will not be obstructed until and unless many nodes start behaving selfishly. So what we intend to do is to incorporate selfish behavior in a certain number of nodes and then check the effect on routing overhead and throughput. We called this as Enhanced AODV

Pseudo code with explanation of En-AODV

1. First we need to modify existing aodv.cc and aodv.h files.

2. In aodv.cc we mention all nodes are initially not selfish.

3. Explicit we will mention which node will work as selfish nodes. This can be done with file management function.

4. If node is in selfish nodes list then, we will drop routing packet otherwise we will continue to forward packets. Sample code is given below:

```
switch(packet_type):
{
case RREQ:
if(selfish==true && node is not destination)
{
drop(packet p);
break;
 }
else
{
        forward_packet(packet p);
        break;
}
case RREP:
if(selfish==true && node is not source)
{
drop(packet p); break;
}
else
{
                forward_packet(packet p);
                break;
}
}
```

5. Results

We will consider topology of 25,50,75,100,125 nodes with varying number of selfish nodes in order to prove our assumption that the routing overhead will decrease with increase in the number of selfish nodes and up to a certain extent our throughput will also increase.

Parameter	Value
Simulation time	1000sec
Number of nodes	25,50,75,100
Protocol	AODV, Enhanced AODV
Pause Time	2 sec
Traffic Type	TCP

We have evaluated our result on following metrics.
End-to-end delay refers to the time taken for a packet to be transmitted across a network from source to destination.
The throughput of the ad hoc network is characterized by the transmission capacity, where the transmission capacity is defined to be the maximum density of spatial transmissions that can be simultaneously supported in an ad hoc network under constraints of maximum retransmissions and reliability.

Nodes often change their location within ad hoc network. Routing protocol need to adapt that situation so they are constantly broadcasting routing packet which generates phenomena called routing overhead on network.

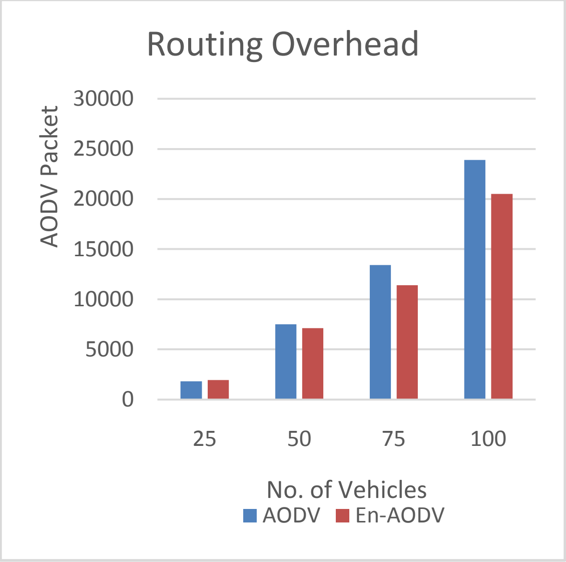

Figure 3 Routing Overhead

It is to be observed from the graph that as we are incorporating selfish behavior with certain number of nodes, we have reduction in routing overhead.

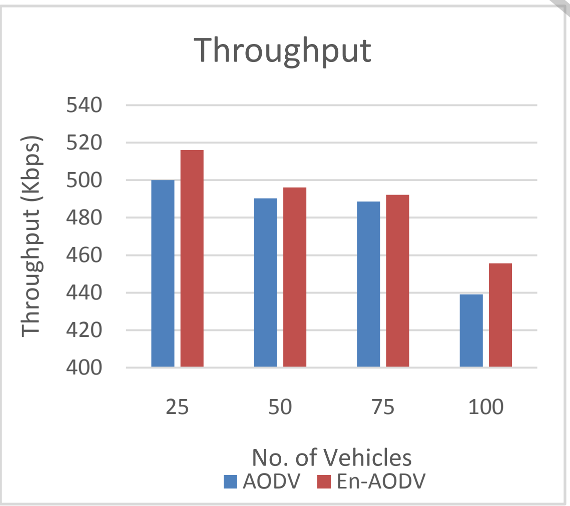

Figure 4 Throughput

From figure 4 we can as number of vehicle are increasing network throughput increase. Even in the case of enhanced AODV because of selfish node communication overhead of network is decreased which intern decrease collision which will help to increase throughput of network.

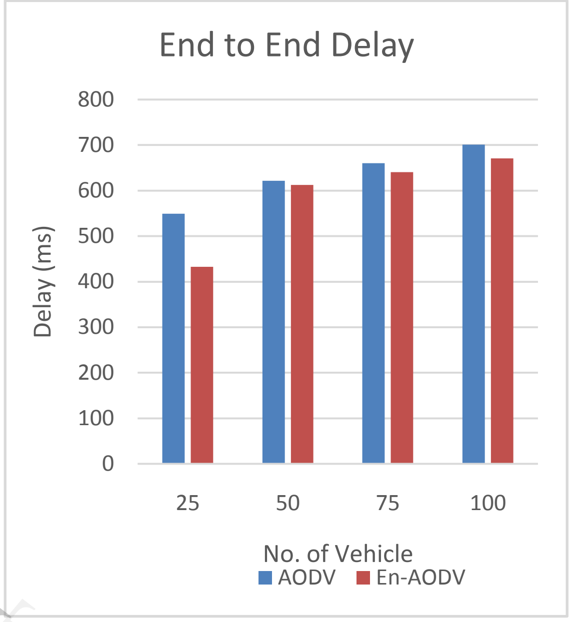

Figure 5 End to End Delay

From figure 5 we can as number of vehicle increase in network total end to end delay increase because packets tend to pass from many number of nodes. Each node will include processing delay, queuing delay and forwarding delay. When we have some node as selfish node, we are able to reduce processing and queuing delay of certain packet which intern reduce overall delay.

So, from all the graphs we can see that as some node behave selfishly there is a general decrease in routing overhead because dropping of control traffic as the selfish nodes don't flood routing packets further and the network stays connected up to a certain percentage increase in selfish nodes.

6. Conclusion

In this paper we have evaluated the performance of AODV as well as Enhanced AODV using NS-2.34 as network simulator and SUMO and MOVE as traffic simulator. We have developed the Enhance AODV protocol for Vehicular Ad-hoc networks which decreased routing overhead and increased the throughput upto a certain level. The basic idea was that when some node in VANET highly mobile environment do not forward routing packets to improve the metrics like routing overhead. This is only possible in dense ad hoc network. If we have sparse network then it is very difficult to create connection between nodes in presence of selfish nodes. In future, more novle ways can be developed to improve routing protocols for VANET.

7. References

[1] Arzoo Dahiya , Dr.R.K.Chauhan, “A Comparative study of MANET and VANET Environment”, Journal of Computing, Volume 2, Issue 7, July 2010, ISSN 2151.

[2] Hadi Arbabi,Michele C. Weigle , “Highway Mobility and Vehicular Ad-hoc networks in ns-3 “, Proceedings of the 2010 Winter Simulation Conference.

[3] Barłomiej Błaszczyszyn, Anis Laouiti, Paul Muhlethaler, Yasser Toor,” Comparison for VANETs: Conventional Routing vs an Advanced Opportunistic Routing Scheme using Active Signaling”, ITS Telecommunications, 2008. ITST 2008. 8th International Conference,IEEE.

[4] Victor Cabrera, Francisco J. Ros, Pedro M. Ruiz, -“Simulation-based Study of Common Issues in VANET Routing Protocols” , IEEE 69th Vehicular Technology Conference, 2009.

[5] Moez Jerbi, Sidi-Mohammed Senouci, Rabah Meraihi and Yacine Ghamri-Doudane,” An Improved Vehicular Ad Hoc Routing Protocol for City Environments”, IEEE International Conference on Communications, 2007.

[6] F. Karnadi, Z. Mo, K.-C. Lan, ”Rapid Generation of Realistic Mobility Models for VANET”, in Proc. of the IEEE Wireless Communication and Networking Conference (WCNC‟ 07), March 2007.

[7] Imran Khan and Amir Qayyum Center of Research in Networks and Telecom (CoReNeT),” Performance Evaluation of AODV and OLSR in Highly Fading Vehicular Ad hoc Network Environments”, IEEE 13th International Multitopic Conference, 2009.

[8] D.Krajzewicz , G.Hertkorn , C.Rossel , P.Wagner , “SUMO(Simulation of Urban Mobility): An Open Source Traffic Simulation”. In: Proceedings of the 4th Middle East Symposium on Simulation and Modelling(MESM2002).

[9] Kun-chan Lan and Chien-Ming Chou,” Realistic Mobility Models for Vehicular Ad hoc Network (VANET) Simulations”, 8th International Conference on ITS Telecommunications, 2008.

[10] Sascha Schnaufer, Holger, Matthias Transier, Wolfgang Effelsberg,” Vehicular Ad-Hoc Networks: Single-Hop Broadcast is not enough”, Proceedings of 3rd International Workshop on Intelligent Transportatio (WIT), pp. 49-54, March 2006, Hamburg, Germany.