Vanet Based Traffic Management System Development And Testing Using Aodv Routing Protocol.

PATIL V.P.
Smt. Indira Gandhi college of Engineering, New Mumbai, INDIA

Abstract:
Vehicular Ad-Hoc network (VANET) is a type of Mobile Ad-Hoc (MANET) network where the nodes are constrained to move along the road. Vehicles in VANET are equipped with a radio device to communicate with each other and also with the road side units (base stations). Vehicular networks aims to make the driving experience safer, efficient and enjoyable. Vehicle traffic congestion is reflected as delays while traveling. Traffic congestion has a number of negative effects and is a major problem in today's society. Several techniques have been deployed to deal with this problem. In this paper, it has been proposed an innovative approach to deal with the problem of traffic congestion using the characteristics of vehicular ad-hoc networks (VANET). The system is developed and tested using AODV protocol of ad hoc mobile network to deal with the problem of vehicle traffic congestion in vehicular networks. The performance is measured in terms of no. of packets broadcasted, percentage of packets delivered, and percentage of traffic diverted and overhead to manage the problem of data traffic congestion in computer networks. Using simulations the applicability of the algorithm in the domain of vehicle traffic congestion in a VANET is demonstrated.

Keywords: VANET, Aodv, Traffic management, Java, Testing, proactive protocol, traffic congestion.

I. Introduction

Vehicular ad hoc networks (VANETs) [1, 2] are attracting the interest of a great number of academicians and industrials. One of the most interesting features is the possibility to use a spontaneous and inexpensive wireless ad hoc network between vehicles to exchange useful information such as warning the drivers of an accident or a danger. One key limitation of many existing tools that integrate vehicular traffic and network simulation is the lack of dynamic interactions between the two domains. Thus transportation simulators would use pre-computed and aggregate network level delay and packet loss computations whereas network simulators would use pre-scripted mobility data. The shortcoming of these approaches is the lack of dynamic interaction between an event (e.g. accident) as it unfolds in the transportation simulator and its dissemination to the vehicles using the network as embedded within the vehicles in its vicinity and the feedback to the transportation simulator the change in velocities and positions of the vehicles as they react in real-time to the information conveyed to them by the communication network. The lack of the above type of dynamic interaction between the transportation and the network simulator reduces the level of realism that can be achieved for key ITS applications like active safety and traveler information systems which in reality influence the vehicles' movements significantly.
Incorporating accurate network simulations into transportation simulators is proven to have significant impact on the predicted performance of applications under realistic operating conditions of the VANET network. The penetration ratio required for Dynamic Route Planning to achieve sufficient performance gain in scenarios of high channel bandwidth is determined by the relative roadway capacity on alternative routes. In these cases, tipping point increases as the relative capacity enlarges. With limited channel bandwidth, higher penetration ratio beyond the tipping point greatly impairs the application performance due to channel saturation [3, 4].In this paper Next section reviews about literature of VANET and scope of traffic management using vehicular network communication is given in the section III, in section IV the implementation of the proposed system is done, section V describes about result analysis and testing and at the end section VI concludes bout the findings of the implemented system.

II. Review of Literature

2. 1 About VANET

Vehicular Ad hoc Networks (VANET) is the subclass of Mobile Ad Hoc Networks (MANETs). VANET is one of the influencing areas for the improvement of Intelligent Transportation System (ITS) in order to provide safety and comfort to the road users. VANET assists vehicle drivers to communicate and to coordinate among themselves in order to avoid any critical situation through Vehicle to Vehicle communication e.g. road side accidents, traffic jams, speed control, free passage of emergency vehicles and unseen obstacles etc. Besides safety applications VANET also provide comfort applications to the road users. [4, 5, 6] Each node within VANET act as both, the participant and router of the network as the nodes communicates through other intermediate node that lies within their own transmission range. VANET are self organizing network. It does not rely on any fixed network infrastructure. Although some fixed nodes act as the roadside units to facilitate the vehicular networks for serving geographical data or a gateway to internet etc. Higher node mobility, speed and rapid pattern movement are the main characteristics of VANET. This also causes rapid changes in network topology. VANET is a special type of MANET, in which vehicles act as nodes. Vehicular ad hoc networks present a promising way to build up a decentralized parking guidance system. Designing such an application can be decomposed into major issues: (1) which information on a parking place needs to be known by the vehicles and thus has to be distributed in the vehicular ad hoc network? And finally, (2) how can this information be used to maximize the benefit for the driver? [9, 10, 11].

2.2 Routing protocols in VANET

A routing protocol governs the way that two communication entities exchange information; it includes the procedure in establishing a route, decision in forwarding, and action in maintaining the route or recovering from routing failure. This section describes recent unicast routing protocols proposed in the literature where a single data packet is transported to the destination node without any duplication due to the overhead concern. Some of these routing protocols have been introduced in MANETs but have been used for comparison purposes or adapted to suit VANETs' unique characteristics. Because of the plethora of MANET routing protocols and surveys written on them, we will only restrict our attention to MANET routing protocols used in the VANET context. VANET routing protocols can be classified as topology-based and geographic (position-based) in VANET.[2-5]

2.2.1 Topology-based Routing Protocols

These routing protocols use links' information that exists in the network to perform packet forwarding. They can further be divided into proactive (table-driven) and reactive (on-demand) routing.

2.2.2 Proactive (table-driven) Routing protocols

Proactive routing carries the distinct feature: the routing information such as the next forwarding hop is maintained in the background regardless of communication requests. Control packets are constantly broadcast and flooded among nodes to maintain the paths or the link states between any pair of nodes even though some of paths are never used. A table is then constructed within a node such that each entry in the table indicates the next hop node toward a certain destination. The advantage of the proactive routing protocols is that there is no route discovery since route to the destination is maintained in the background and is always available upon lookup. Despite its good property of providing low latency for real-time applications, the maintenance of unused paths occupies a significant part of the available bandwidth, especially in highly mobile VANETs.

2.3 Ad hoc On demand Distance Vector routing protocol (AODV)

Unlike traditional wire line networks, ad hoc networks don't rely on existing infrastructure to support communication. Each mobile node acts as an end node when it is the source or destination of a communication and forwards packets for other nodes when it is an intermediate node of the route. This makes multi hop communication possible. Ad hoc networks are easier to deploy than wire line networks and have found many applications, such as in rescue, battlefields, meeting rooms etc., where either a wire line network is unavailable or deploying a wire line network is inconvenient. [6, 7]

AODV: AODV has the merits of DSR and DSDV protocol. DSDV maintains routes to all destinations with periodical route information flooding and uses sequence numbers to avoid loops. AODV inherits the sequence numbers of DSDV and minimizes the amount of route information flooding by creating routes on-demand, and improves the routing scalability and efficiency of DSR, which carries the source route in the data packet. In AODV protocol, to find a route to the destination, the source broadcasts a route request packet (RREQ). Its neighbors relay the RREQ to their neighbors until the RREQ reaches the destination or an intermediate node that has fresh route information. Then the destination or this intermediate node will send a route reply packet (RREP) to the source node along the path from which the first copy of the RREQ is received. AODV uses sequence numbers to determine whether route information is fresh enough and to ensure that the routes are loop free. In AODV protocol, the route is built on demand and is not updated until the route breaks or times out. The route can't adapt to topology changes and breaks frequently in the case of high mobility. AODV uses local repair to restrict the route discovery zone so as to reduce overhead. If the source moves away and causes the route to break, it can re-initiate route discovery to the destination. In case an intermediate link breaks and the broken link is near the destination, the upstream node of this broken link may choose to repair the route. It initiates a route request with a fresh destination sequence and the RREQ will be flooded in a zone with a radius no less than the original hop count between this node and the destination. If the upstream node of the broken link decides not to repair the broken route, it will send a route error packet (RERR) upwards to the source node. The source node will re-initiate route discovery in an even bigger zone than that of the local repair if the route is still needed.

2.4. A Robust AODV Protocol

In AODV protocol, usually the RREQ that first arrives at the destination determines the route. The route may contain neighboring nodes with long distances, which leads to low stability. And the node's mobility makes the route more susceptible to breaking. Our purpose is to make AODV adaptive to mobility and more robust. The Robust AODV protocol contains three parts: active route discovery, backup route building, and route maintenance. [6, 7].

A. Active Route Discovery.

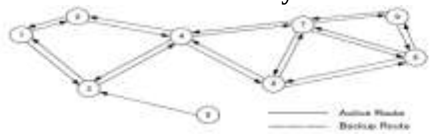

Fig. 1 Active route and backup route

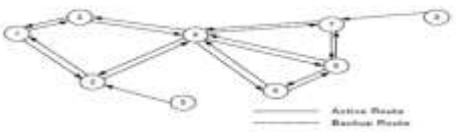

Fig.2 Switch to better route with lower hop count

In our Robust AODV protocol, the route discovery process is almost the same as in the original AODV except that all routes are built as backup routes. Only when data packets pass through will the route flag be changed to ACTIVE. From then on, the active route information is broadcasted to 1-hop neighbors, so that backup routes can be built. Here only the active route is maintained. If no packets pass through within a certain time, the active route will time out and be switched to a backup route then it's not maintained.

Fig. 1 is an example: the active route between node 1 and node 8 is built as 1-3-4-7-8. Node 1 has two routes for node 8, the route whose next hop is node 3 is an active route and the route whose next hop is node 2 is a backup route.

B. Backup Route Building

In Robust AODV protocol, the AODV hello message is substituted by the ad hoc route update message, which is broadcasted only to 1-hop neighbors by the node who is on the active route or who can hear the active route information. For each destination, at most one route is added to the route update message. The route can be either an active route when the node is on the active route or the highest priority backup route when the node can hear the active route information. The number of routes in the route update message is no more than that of the total active routes in the ad hoc network.

The broadcasted route update message contains all necessary route information, especially the destination sequence number and the next hop, which are important for avoiding loops. Multiple route update messages may be received from different nodes. The route with a fresh sequence number and lower hop count has the highest priority and is preferred, which is the same as in the original AODV. For incoming new route information, the sequence number is checked first. If this route has an old sequence number, it is discarded. If it has a fresh sequence number with bigger hop count, it must be stable before it is switched to become the new active route. Next, the hop count is checked. If the received route's hop count is lower than the current hop count +1, it is added to the route table as a backup route.

C. Route Maintenance

Route entries are updated when a route update packet is received. If a better route is available after updating, this route will be switched to become the new active route. Routes are monitored and maintained when they are used to forward data packets. When a link break is detected, if a backup route exists, the route switch is performed and the new route is checked; if no backup route exists, an RERR is sent toward the source node, just like in the original AODV.

1) Update Route Entry: Each time a route update message is received from an old sender, the corresponding route's next hop's next hop is updated. The backup route's lifetime also gets updated. The backup route will be deleted if it times out.

2) Switch to Better Route.: When the current active route is less preferred, the highest priority backup route is switched to become the new active route. And the old active route will be a backup route. In Fig. 1, node 8 moves near to node 4 and the shorter route 8-4-3-1 is switched to become the new active route, as is shown in Fig. 2. In this way, routes in Robust AODV adapt to topology variations. In the original AODV, when node 8 moves towards node 4, the route is not updated; and as node 8 continues moving, the link between node 8 and node 7 will break, and local repair may be conducted, which actually can be avoided.

Iii. Vanet For Traffic Management System

The current traffic management system in India is not complete and efficient for the public. There are billions of vehicles running on the roads. Therefore, it is highly desired to have a reliable and cost effective way to track the traffic or congestion on roads and thus choose the appropriate road that is congestion free. If there is base station in the square who got idea about congestion, then it transmit the signal containing information about the congestion, car receives the signal through trans receiver and thus choose another road to go further. The car can transmit the signal within its range to alert incoming car about accident. With the advance and wide deployment of wireless communication technologies, many major car manufactories and telecommunication industries gear up to equip each car with the On Board Unit (OBU) communication device, which allows different cars to communicate with each other. Roadside infrastructure i.e. Roadside Units (RSUs), in order to improve not only road safety but also better driving experience. Therefore, it becomes possible to track the congestion on roads and then guide drivers to the available congestion free road, through vehicular communications.

3.1 System implementation Requirement

3.1.1 Software specification:

1) Operating System – Windows 7or XP.;2) Front end – Java ;3) Back-end database – Java
4) NetBeans IDE 7.0.

3.1.2 NetBeans Platform

The NetBeans Platform is a reusable framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally-signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again. NetBeans IDE is a free, open-source, cross-platform IDE with built-in-support for Java Programming Language.

NetBeans IDE: NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactoring, version control (supporting CVS, Subversion, Mercurial and Clear case).

Modularity: All the functions of the IDE are provided by modules. Each module provides a well defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

3.2 Operational feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. Fig.4 shows data flow diagram (DFD) for Client to Client Communication and Fig.5 shows DFD for Client to Base Station Communication. Fig 3 shows the flow chart of the system.

3.2 FLOW CHART

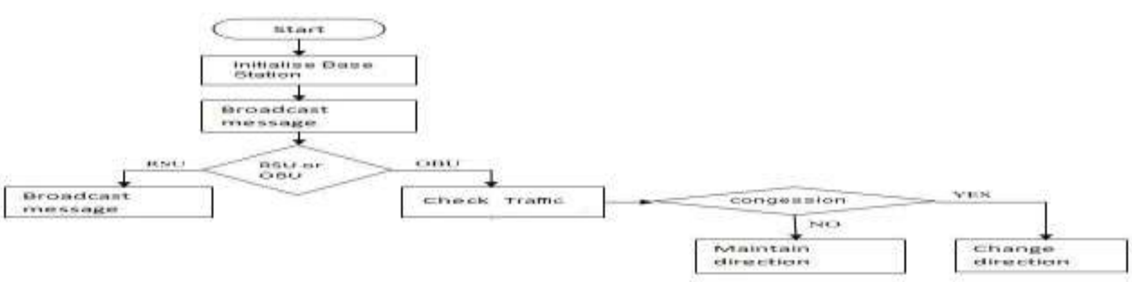

Fig.4: FLOW CHART

IV. Implementation of the Proposed System

4.1 Algorithm development

We have implemented AODV algorithm for the proposed system and its stages are as follows:
a)Active Route Discovery ;b)Backup Route Building ;c) Route Maintenance; i)Update Route Entry Switch to better route; ii) Switch route in the case of link break; iii) Route check; iv) Hop count restriction

4.2 Screen Snapshot & module description

The system is tested for its working under different situations and the working is satisfactorily observed as shown in the screen snapshots shown in fig 6 to 11.

User Interface*:* Fig.6 is first Main page of GUI of Our System shows the simulation scenario without Input panel.Fig.7 is first Main page of GUI of Our System shows the simulation scenario with input panel. Fig.8 is first Main page of GUI of Our System shows the simulation scenario with input parameters. As shown in Fig.9, In this scenario our system shows the accidental scenario and the start the broadcast of messages. In the scenario as shown in fig 10, our system shows Inter Vehicle Communication and broadcast of messages.Fig.11 shows our system shows inter vehicle communication and traffic diversion due to congestion of traffic.

As in Fig.12, The graphs show: Graph.1 Overhead on base station per second; Graph.2 Number of packets broadcasted; Graph.3 Percentage of packets delivered; Graph.4 Percentage of traffic diverted.

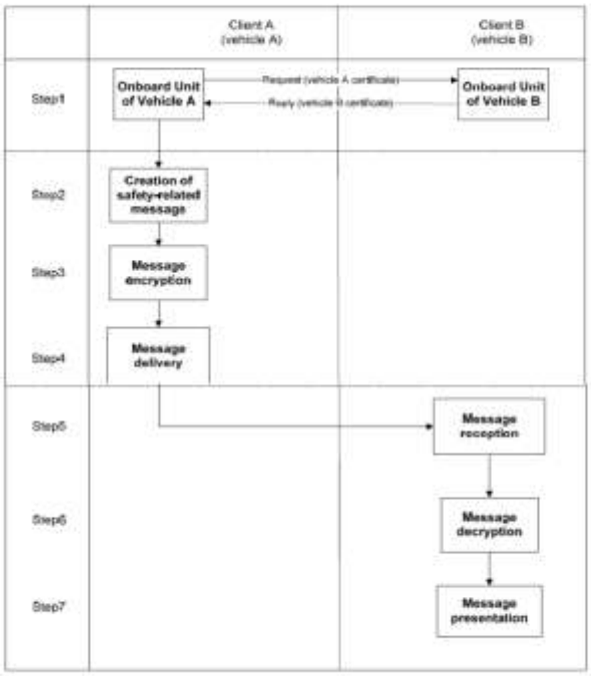

Fig.4: DFD for Client to Client Communication.

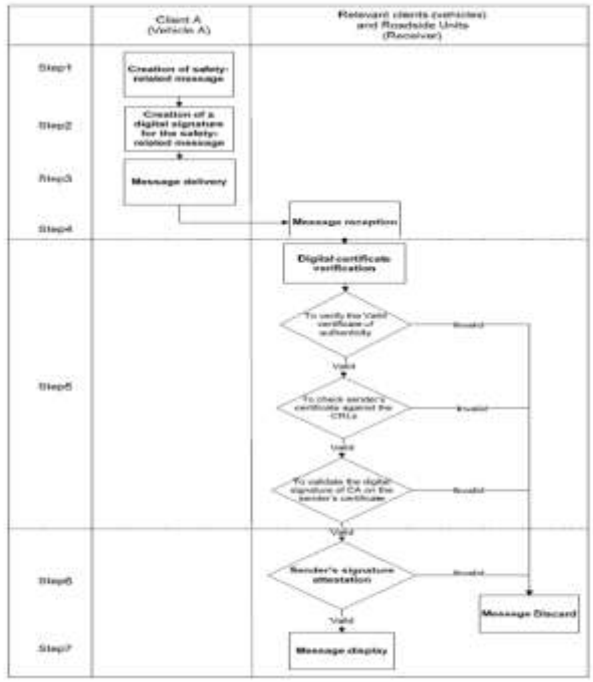

Fig.5: DFD for Client to Base Station Communication

Fig.6: Basic Design

Fig.7: Basic Design With Input Panel

Fig.8 Basic Design With Input Panel

Fig.9: Accident Scenario & Packet Broadcast

Fig.10: Inter Vehicle Communication

Fig.11: Traffic diversion

4.3 Output Graphs:

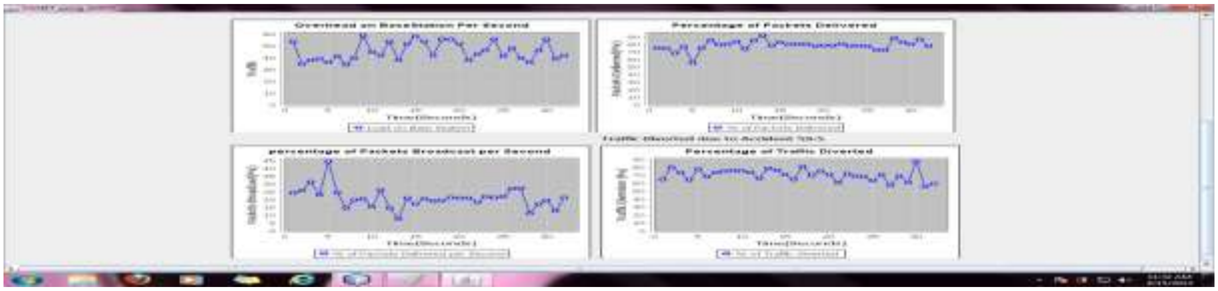

Fig.12: Output Graphs

V. Testing and Result Analysis

Testing is process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers previously undiscovered errors. The main objective of testing, thus, is to design tests that uncover different classes of errors, and to do so with minimum time & effort spent.

5.1. Testing methods:

Two classes of input are provided to the test process:

i) A software configuration that includes a Software Requirement Specification, Design Specification and Source code.

ii) A test configuration that includes a test plan and procedure, any testing tools to be used and testing cases with their expected results.

5.1.1 White Box Testing:

White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities.

The general outline of the white box testing process is as follows:

i) Exercise all independent paths within the module at least once.

ii) Exercise all logical decisions for both true and false scenarios

iii) Execute all loops at their boundaries and within their operational loops

iv) Execute test cases and communicate results.

5.1.2 Black Box Testing

It is also known as functional testing. A software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs.

Black Box testing attempts to find errors in the following categories:

i) Incorrect or missing functions.

ii) Interface errors.

iii) Errors in data structures or external database access.

iv) Performance errors

v) Initialization and termination errors.

5.2 Testing Vanet Based Traffic Management System.

Vanet Based Traffic Management System is tested using Black Box Testing. In this type the functional requirements of the system are tested. The system is given some input and the outputs of the system are studied to uncover the errors that might be present in the system.We test the our Project at two side i.e. Administrator Side, User Side

5.2.1 Testing VANET Based Traffic Management System at Administrator Side

Test Cases for Default Input:

i) Testing is done to check that when the user logs dosent provide any input.
The System takes default input for every variable parameter.

Test Cases for User Input:

i) The main case for the user input is that all fields are filled properly.
ii) To check whether each field is filled with valid input.

5.3 Running a Test

Once you have created a test or test suite, you can use the Run Test command to initiate execution of the test. Run Test commands are available on source nodes only. After you run a test, you can choose to rerun individual test methods executed during the test and displayed in the JUnit Test Results window.

a) To run tests for an entire project:

1. Select any node or file in the project you want to test in the Projects or Files window.
2. From the main menu, choose Run > Test *project_name* (Alt-F6).
The IDE executes all of the project's tests.

- If you want to run a subset of the project's tests or run the tests in a specific order, you can create test suites that specify
- the tests to run as part of that suite. After creating a test suite you run the suite in the same way you run a single test class.

b) To run a test for a single class:

1. Select the node of the class for which you want to run a test in the Projects or Files window.
2. From the main menu, choose Run > Run File > Test *classname* (Ctrl-F6).

- You can also run a class's test by selecting the test class node itself and choosing Run > Run File > Run
- testclassname (Shift-F6).

c) To run a single test method:

1. Run the test class or suite containing the test method.
2. In the JUnit Test Results window, right-click the test method and choose Run Again.

- To run a single test method the method must be listed in the JUnit Test Results window.

5.4 Working with JUnit Test Output

When you run a test, the IDE shows the test results in two tabs in the JUnit Test Results window:

- A summary of the passed and failed tests and the description of failed tests are displayed in the left pane of
- the window.
- The textual output from the JUnit tests themselves is displayed in the right pane of the window.
- The output from the Ant process that builds and runs the test is displayed in the Output window. Double-click any error to jump to the line in the code where the error occurred.

5.5 Result analysis:

It has been observed that almost all route request broadcasts reach the destination, only a few over long distances with middle car density failed. But the load on the network originating from the naive broadcast is tremendous. As a result it also leads to quickly growing delays and link failure. Several route replies do not come through because broadcasting is still going on. This is a critical problem, especially in city areas with high car density. It seems to be appropriate to replace the common broadcast system by another, more efficient, version. Another phenomenon that can often be observed is that a route breaks before the data packet can be successfully transmitted or even that the route reply does not find its way back. This is most critical for short transmission ranges and high mobility. During analysis of the trace files, we observed that such link breakages mostly come from the following situation: let us take the highway scenario and think of two cars driving in the same direction. They are driving 120 kilometers per hour and the distance between them is several kilometers (dozens of hops). If we are able to establish a route between them where every intermediate node drives in the same direction, the route is more or less stable. But if only one intermediate node drives in the opposite direction we have a serious problem. Such a node covers about 35 meter per second or 70 meters per second relative to the cars driving in the opposite direction. This is about 20 to 50 percent of the simulated transmission ranges. Since we know from the simulations above that a transmission of a single data packet with route establishment over such a distance takes about a second, it is logically that this leads to a link breakage. If one wants to use AODV on highways, it is essential to extend it in a way that avoids such link breakages.

VI. Conclusion

The traffic management system using AODV protocol for VANET under java environment is simulated and tested satisfactorily. The testing was done for one of the event of accident at one place and due which it was observed that traffic automatically diverts to another alternative route. This was possible due to multi hop vehicle to vehicle communication and vehicle to road side base station communications. It has been observed that almost all route request broadcasts reach the destination, only a few over long distances with middle car density failed. But the load on the network originating from the naive broadcast is tremendous. As a result it also leads to quickly growing delays and link failure. Several route replies do not come through because broadcasting is still going on. This is a critical problem, especially in city areas with high car density. It seems to be appropriate to replace the common broadcast system by another, more efficient, version. Another phenomenon that can often be observed is that a route breaks before the data packet can be successfully transmitted or even that the route reply does not find its way back. This is most critical for short transmission ranges and high mobility. Since we know from the simulations and testing of this system that a transmission of a single data packet with route establishment over such a distance takes about a second, it is logically that this leads to a link breakage. If one wants to use AODV on highways, it is essential to extend it in a way that avoids such link breakages. In future the system can be implemented using network simulator NS-2 and using any other VANET protocol.

References

[1] Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter, Chaired by Joseph Macker and Scott Corson, http://www.ietf.org/ html.Charters/manet -charter.html

[2] Charles E. Perkins, Pravin Bhagwat, Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, ACM SICOMM'94 Conference on Communications Architectures, Oct, 1994, pp234-244

[3] David B. Johnson, David A. Maltz, Yih-Chun Hu , The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, draft-ietf-manet-dsr-09.txt, Apr.15, 2003

[4] Perkins, C.E.; Royer, E.M., Ad-hoc on-demand distance vector routing, Mobile Computing Systems and Applications, 1999. Proceedings, pp: 90 –100, 1999

[5] C. Perkins, E. Belding-Royer, S. Das**,** Ad hoc On-Demand Distance Vector (AODV) Routing, RFC3561, July, 2003

[6] Pearlman, M.R.; Haas, Z.J., Determining the optimal configuration for the zone routing protocol, Selected Areas in Communications, IEEE Journal on, Vol.17, Issue: 8, 1999, pp: 1395 - 1414

[7] Lee, S.-J.; Gerla, M.;, AODV-BR: backup routing in ad hoc networks, Wireless Communications and Networking Conference, pp:1311-1316, Vol.3, 2000

[8] D. K. Tokuda, M. S. Nakabayashi, M. M. Hamaguchi, and M. H. Tsutsui. Oki project. http://www.ece.osu. edu/hpcnl/okipublic/, 2004.

[9] IEEE Computer society LAN MAN Standards Committee, Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997, IEEE, 1997

[10] Bonnmotion - a mobility scenario generation and analysis tool. http://web.informatik.uni-bonn.de/IV/ Mitarbeiter/dewaal/BonnMotion/.

[11] T. Kosch. Efficient message dissemination in vehicle ad-hoc networks. In Proceedings of the 11th World Congress on Intelligent Transportation Systems, October 2004.

[12] C. Schroth, F. D¨otzer, T. Kosch, B. Ostermaier, and M. Strassberger. Simulationg the traffic effects of vehicle-to-vehicle messaging systems. In Proceedings of the 5th International Conference on ITS Telecommunications, June 2005.

Author

PATIL V.P. is currently working as a faculty member in Electronics and Telecommunication Engineering department in smt. Indira Gandhi college of Engineering .New Mumbai. His is graduate in B.E. and post graduate in M.TECH (ELECTRONICS DESIGN AND TECHNOLOGY).He is having 25 years of experience in teaching in engineering colleges. His area of research is in computer communication networking and microwave engineering.