ISSN No. 0976-5697

Volume 4, No. 8, May-June 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Agent Movement Optimization in VANET using Load Shedding Algorithm

T. Karthikeyan
Associate Professor
Department of Computer Science
PSG College of Arts and Science
Coimbatore, India

N. Sudha Bhuvaneswari
Associate Professor School of IT & Science
Dr.G.R Damodaran College of Science
Coimbatore, India
sudhanarayan03@yahoo.com

Abstract: In the growing world of Information Technology the Mobile Agent based application has taken a prominent and promising place in distributed system applications over the client server approach. Major concerns of these distributed applications are to provide quality service and improved performance. This is not the case only in network traffic but it also poses problem in vehicular ad hoc networks (VANET) where agent based approach has taken a prominent place in handling effective road and vehicle traffic. Research works are still under progress in VANET to efficiently handle message passing, congestion avoidance and efficient traffic controlling. This paper highlights on the integration of previously defined MATLB, PCM and MSA Agent and proposes an approach for reducing the size of the mobile agent that wanders in the network carrying messages from node to node. This reduction in size of mobile agents will enhance the performance of VANET making the agents more acceptable by the hosts and correspondingly building an effective co-operative vehicular network.

Keywords: *M*ATLB, PCM, MSA, DoS, RFID, VANET

I. INTRODUCTION

The Mobile Agent Architecture is a new paradigm applied in distributed networks and this architecture has found a new place in recent days in vehicular ad-hoc networks (VANET) also. An agent is an entity that is capable of performing various tasks at various places with the feature of mobility and self-control[5]. This agent based approach provides a strong infrastructure to most of the distributed applications making information retrieval and computing efficient and easier. Other potential advantages of using mobile agent approach are fault tolerance, robustness and effective message passing over client server applications [2].

Though mobile agent approach has many advantages there are few limitations of these agents in terms of number of agents, size of agents and routing time of agents [1]. Cost and Time are two major performance factors to be taken in consideration when designing a new architecture. An increase in the size of an agent gradually increases the cost and the time and bandwidth to reach nodes associated with it in the network. This paper refers to already defined agents MATLB used for traffic load balancing with sensors, PCM agent used for identifying emergency vehicles and to control the speed of other vehicles using RFID tag and Hall based sensors, MSA agent to identify selfish nodes in a network that participate in Denial of Service attacks and to isolate them, thereby building a co-operative vehicular network. The approach proposed here integrates all the above discussed agents MATLB, PCM and MSA in a single architecture and applies size reduction techniques to create light weight mobile agents for efficient performance of agents in VANET

II. RELATED WORK

Baek et.al has proposed algorithms to minimize the number of agent movements across the network and to control the cost of transportation associated with the movement of agents[1].

Magdy et.al in his work has proposed an algorithm for dynamic load balancing in which he has proposed agent and its packet contents that can potentially improve the performance of network in an agent based architecture [4].

According to Jin the mobile agent approach he has proposed is to create dynamic mobile agents that can change their itinerary according to the current status of the network with agent cloning facility thereby making agent visits available to all connected nodes in the network [3].

III. INTEGRATED AGENTS ARCHITECTURE

The integrated agents based architecture approach is capable of integrating the functionalities of three agents that forms the backbone for the architecture. The three agents MATLB agent, PCM agent and MSA agent are capable of handling effective network traffic in VANETs. In this integrated architecture the MATLB, PCM and MSA agents should work in co-ordination to achieve the task of handling congestion, giving priorities to emergency vehicles in a busy or congested road and efficient message passing between the infrastructures and vehicles involved in the architecture and identifying selfish behavior of such nodes and helping them to participate in the network to build an effective co-operative vehicular network.

This integrated architecture gives potential solution incase of a VANET with bursty traffic and heavy congestion by providing large number of advantages over the traditional client server system approach.

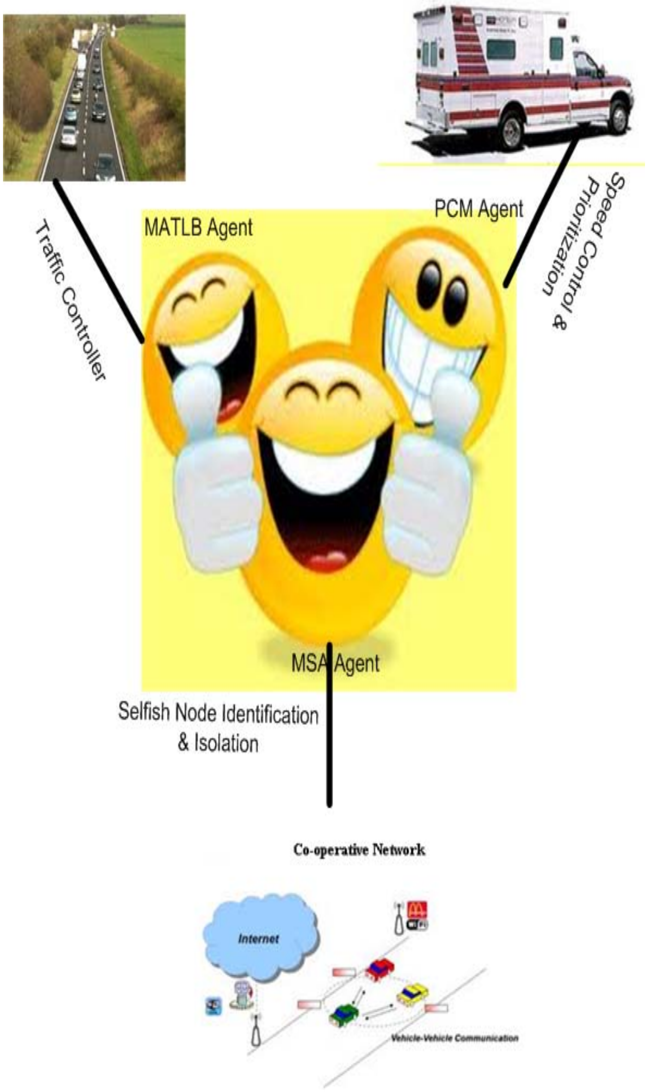

Figure 1. Integrated Agents Architecture

In the above diagram all the three agents MATLB, PCM and MSA should work collaboratively. The basic functioning begins with the MATLB agent that takes the responsibility of identifying a congestion scenario in a road and accordingly controls the traffic either by re-routing the vehicles using automated messaging with the help of sensors attached with each vehicular node[8].

The second stage is handling the vehicles in such a way that they do not create congestion on the first hand and creates a risk free traffic environment where emergency vehicles are given priority using PCM agent along with RFID tags and hall effect based sensors that are capable of controlling the speed of the vehicle and creating an accident free zone[6].

The third stage is the role played by the MSA agent and this agent is capable of communicating with vehicular node and infrastructure node and it can identify misbehaving nodes that are selfish and participate in Denial of Service (DoS) attacks and isolates these selfish nodes are helps in making these nodes to participate in the network activity thereby building a co-operative vehicular network[7].

This integrated architecture takes into consideration the handling of VANET traffic scenario controlling speed, prioritizing vehicle and avoidance of Denial of Service attacks using an agent approach. The next step in this work is towards optimizing the performance of the mobile agent approach that is discussed in detail below.

IV. AGENTS PERFORMANCE OPTIMIZATION

Performance is a major factor of measurement in any architecture. In the discussed architecture there are three agents working in co-ordination with each other and one of the important problems that this architecture faces is the size factor of agents. As agents keep moving in the VANET carrying potential information, there is a possibility that their size may increase. This result in some nodes rejecting or refusing the agent due to its size and this also creates problem in increase of time and network traffic and the approach discussed in this paper helps to reduce the size of these agents there by potentially improving the performance of the entire architecture.

Basically the agent basic unit of representation is packets and an agent packet contains information like the type of agent classified as MATLB, PCM and MSA, a unique identifier for each agent, the lifetime of the agent, assigned nodes represent the list of nodes an agent has to visit, destination node represents the next node the agent has to visit in its itinery, the code represents the task to be carried out by the agent and the home node represents the node that is responsible for creating and invoking the agent[Magdy] and the variable component contains variables, their status and their location of usage.

Type of Agent (MATLB, PCM, MSA)	Agent Id (Unique Identification)	Lifetime (Period an Agent should be active)	Assigned Node (List Item)	Destination Node (Pointer to next node)	Coding (Agent task to be performed)	Home Node (Creator of agent)	Variable Component

Figure 2. Agent Packet Format

A normal agent will have the packet contents as stated above. But the problem with agent based approach is the variation in the packet size as the agent moves from node to node that dramatically increases overheads in terms of cost, time and resource utilization. The approach of Agent Load Shedding mechanism is to increase the performance of mobile agents.

A. Agent Load Shedding Mechanism:

The Agent Load Shedding Mechanism is a process of size reduction of mobile agents for optimization purpose. The various assumptions considered for implementing this mechanism are:

a. Agents are mobile.
b. They move from node to node to perform tasks.
c. Agents require mobility management mechanism.
d. Agents are controlled by control units present in distributed locations.
e. Control unit takes the complete responsibility of agent size reduction.

In addition to the above made assumptions the various components required for reducing the size of the mobile agent are: Control Unit and Agent Status Tracker. The Control unit consists of other sub components like Garbage

Collector, Reconstructor, Data Collector, Security Stamper and Itinerary Changer.

- ***a.*** ***Agent Status Tracker:*** This agent tracker maintains all important information about the agent maintaining details about the variable, its location of usage and the status of usage of the variable. If the status is set to ON it is an indication that the agent still needs the variable or element else the usage of that particular element is over.
- ***b.*** ***Control Unit:*** The control unit works in co-ordination with the sub-components garbage collector, reconstructor, data collector, security stamper and itinerary changer to help the agent to reduce its size as it moves over the network.

The control unit is the place where the agent can shed its size securely and this unit controls all the other sub-components. The services to be offered by the control unit are:

a) Encrypting and Decrypting of Agent Status Tracker and duly updating the status of this Agent Status Tracker.
b) Identifying unused elements and marking them as deleted.
c) Rebuilding the agent duly removing deleted elements
d) Updating information of new agent in the Agent Status Tracker.
e) Making arrangements for the agent to continue its duty.

- *a.* ***Garbage Collector:*** This component is responsible for identifying the elements that are to be removed from the agent based on the agent's itinerary and Agent Status Tracker.
- *b.* ***Reconstructor:*** With the specification given by the Garbage Collector, Reconstructor takes the responsibility of removing these elements and to rebuild the agent packet for further transmission.
- *c.* ***Data Collector:*** This component is responsible for storing the data collected by the agent as on time and prepares a summary of the data collected for future use.
- *d.* ***Itinerary Changer:*** This component is responsible for scheduling or re-scheduling the itinerary of the agent, if the routing decision of the agent has to be changed dynamically.
- *e.* ***Security Stamper:*** This component takes the responsibility of securely moving the agent from place to place using encryption and decryption techniques.

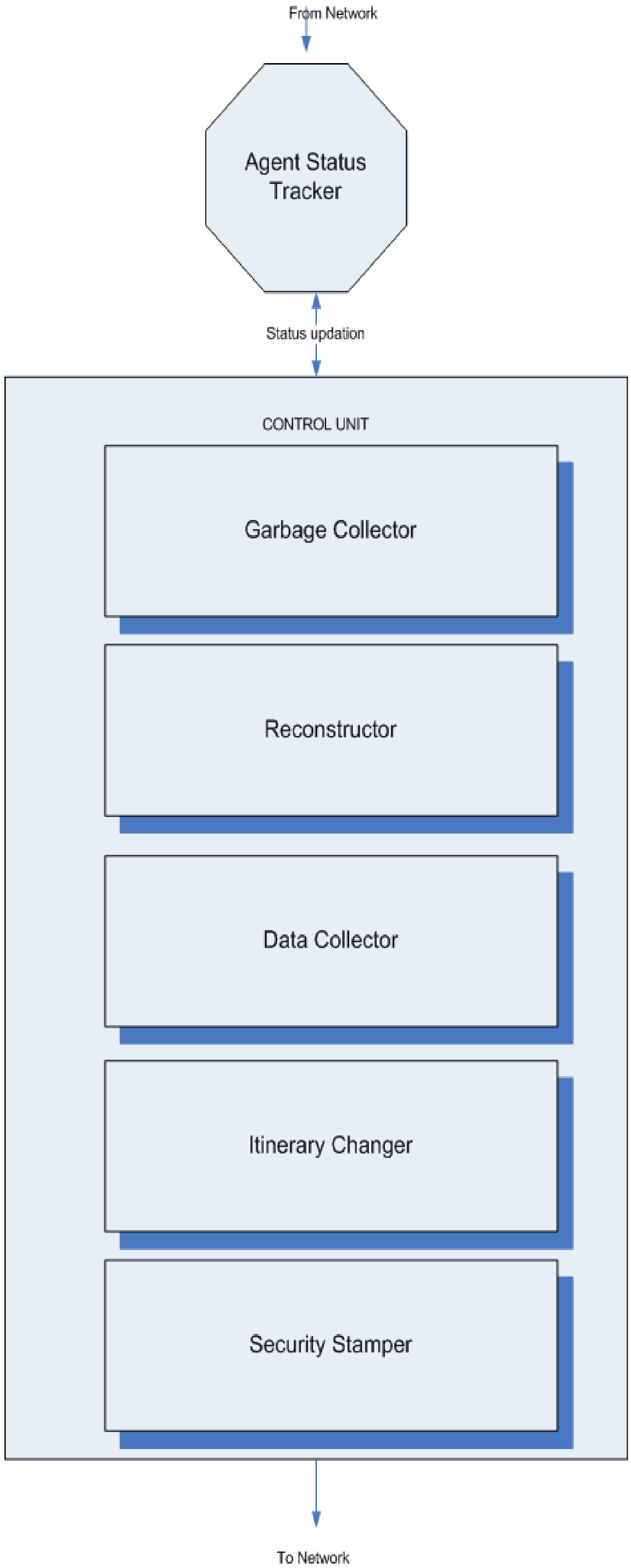

Figure 3. Agent Load Shedding Components

V. PERFORMANCE EVALUATION OF AGENT LOAD SHEDDING MECHANISM

To evaluate the performance of the Agent Load Shedding Mechanism a comparative experimentation was carried out in software developed using .NET framework and compares the agent size reduction before implementing the mechanism and after implementing the mechanism. The results of the experiment are tabulated as given below:

Table: 1 Agent Size Before Implementing Load Shedding Mechanism

Agent Size before Implementing Agent Load Shedding Mechanism	**Agent Size & Location**	
	Nodes	Size in Bytes
	Home Node	8345
	Vehicle Node1	8520
	Infrastructure Node1	8556
	Vehicle Node2	8623
	Vehicle Node3	8627
	Average	8534.2

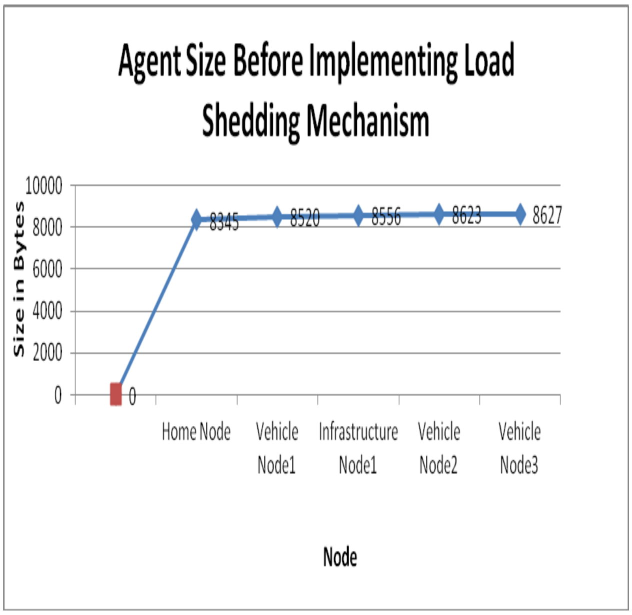

Figure 4. Agent Size before Load Shedding

Table: 2 Agent Size After Implementing load Shedding Mechanism

Agent Size after Implementing Agent Load Shedding Mechanism	**Agent Size & Location**	
	Nodes	**Size in Bytes**
	Home Node	8345
	Vehicle Node1	8520
	Infrastructure Node1	8556
	Control Unit	4730
	Vehicle Node2	4735
	Vehicle Node3	4780
	Average	6987.2

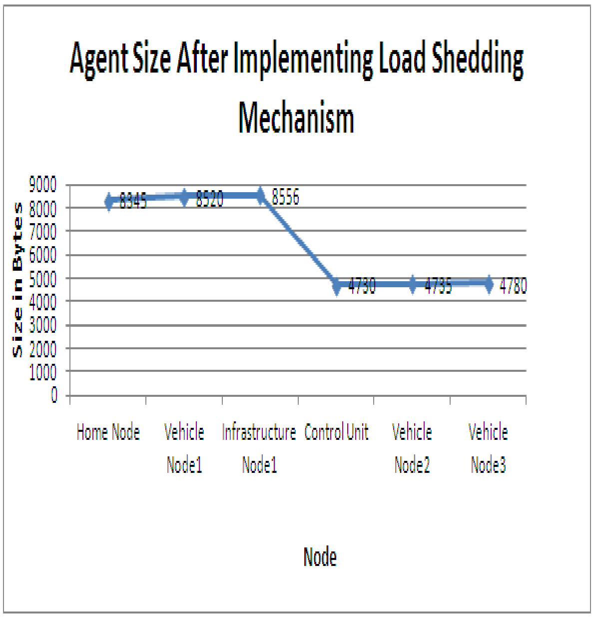

Figure 5. Agent Size after Load Shedding

VI. CONCLUSION

This paper provides an approach to handle the important factor of agent size reduction in an integrated agents environment using the mechanism of load shedding with the help of the control unit. After applying the mechanism of load shedding there is a dramatic reduction in the size of the agent that are shown in Table I and Table II. This approach is therefore helpful to optimize the performance of the agent based approach to manage the traffic in vehicular ad hoc network effectively and efficiently.

VII. REFERENCES

[1] J.W. Baek, J.H. Yeo, G. Kim, H.Y. Yeom, Cost effective mobile agent planning for distributed information retrieval, in: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), pp. 65-72. April 2001.

[2] Danny B. Lange and Mitsuru Oshima, Programming and Deploying Java TM Mobile Agents with Aglets TM, ISBN 0-201-32582-9, Addison-Wesley, 1998.

[3] Jin-Wook Baek, Jae-Heung Yeo, and Heon-Young Yeom, Agent Chaining: An Approach to Dynamic Mobile Agent Planning, School of Computer Science and Engineering, Seoul National University, Seoul, 151-742.

[4] Magdy Saeb, Cherine Fathy, Performance Evaluation of Mobile Agent-based Dynamic Load Balancing Algorithm, Computer Engineering Department, Arab Academy for Science, Technology & Maritime Transport, 2001.

[5] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode. Trafficview: Traffic data dissemination using car-to-car communication. ACM Sigmobile Mobile Computing and Communications Review, Special Issue on Mobile Data Management, 8(3):6–19, July 2004.

[6] T.Karthikeyan, N.Sudha Bhuvaneswari, , S.Sujatha (2012), Traffic Handling Approach with Intelligent Speed Control and Prioritization of Emergency Vehicles using PCM Agent ,International Journal of Computer Technology and Application(IJCTA), Volume 3,Issue 4.

[7] T. Karthikeyan, N.Sudha Bhuvaneswari, An Agent based approach towards building a co-operative vehicular network, International Journal of Advanced Research in Computer Engineering and Technology, Vol.6, 2012.

[8] Karthikeyan.T, S.Sujatha, N.Sudha Bhuvaneswari, Mobile Agent Approach for Traffic Load Balancing using Sensors, International Journal of Computer Applications, Volume 47, No. 6, 2012.

Short Bio Data for the Authors Profile

Thirunavu Karthikeyan received his graduate degree in mathematics form Madras University, Post graduate degree in Applied Mathematics from Bharathidasan University and received doctoral degree in Computer Science form Bharathiar University. Presently he is working as an Associate Professor in Computer Science, PSG College of Arts & Science, Coimbatore. His research interests include image coding, Mobile agents, Pattern Recognition etc. He has published many papers in National and International Conferences and Journals. He has completed many funded projects with excellent comments.

N.Sudha Bhuvaneswari started as a Computer Science graduate. She completed her undergraduate degree at Sri Avinashilingam Deemed University for Women, Coimbatore and completed MCA from Bharathidasan University, Tiruchirappalli and research programme M.Phil in Computer Science from Bharathiar University, Coimbatore, India, and is currently pursuing her doctorate in Computer Science. Her area of interest is Mobile Agent Technology & Networks. She has been participating continuously in research and development activities for the past twelve years. To her credit, she has presented and published technical papers in International Journals, at International Conferences and International Workshops organized by various international bodies like IEEE, WSEAS, and IEEE Explore, ACM. She has published book on Integrating Service Oriented Architecture and Web Services and also contributed chapters in books like Personal Area Network and published articles and working manuals in agent technology. The author is currently employed as Associate Professor at the Dr. G.R Damodaran College of Science, Coimbatore, India. She is an active member of various technical bodies like ECMA, Internet Society of Kolkata and Chennai and acts as a moderator in various international conferences and journals.