Mobile Agent in InVANET – Traffic Management

V. Arun[1] and K.L. Shunmuganathan[2]

[1] Sathyabama University, Chennai, India
[2] Department of CSE, RMK College of Engineering, Chennai, India
arunvragavan@gmail.com, kls_nathan@yahoo.com

Abstract. Intelligent vehicular ad hoc networks (InVANETs) provide an effective communication between vehicles with dynamic mobility. This paper enhances the traffic management system using mobile agents called as Intellect Mobile Agent (IMA). IMA is used to identify the traffic in the path from source to destination and provides an effective way to transport through less traffic scenario. IMA designed with an algorithm which calculates the amount of traffic as a parameter and finds a least path to reach the destination. The vehicle sends the IMA with the destination node which traces the traffic path. IMA identifies the path with least traffic using probability calculation and directs the vehicle to reach the destination.

Keywords: Traffic management, mobile agent in InVANETs, Intellect Mobile Agent.

1 Introduction

Traffic on roads is concentrated in signal area due to heavy flow of vehicles. The traffic can be reduced by directing the vehicles to the destination through least traffic area. InVANETs technology provides an efficient way to implement Intellect Mobile Agent (IMA). IMA is an mobile agent which helps vehicle to get traffic less short path. Roadside Equipment (RE) is placed in the signal which calculates the number of vehicles and communicates with IMA. RE senses the number of nodes in the signal area and intimate to the IMA.

1.1 Notations Used

IMA- Intellect Mobile Agent
RE- Road side Equipment
TF – Traffic Flag
ILRE- Identify Linking RE Mobile Agent

2 Category

A group of area is grouped as category which holds the group of RE. Normally, category represents a city with few RE. The RE holds the data about neighbor RE and shortest route to reach any RE in the category.

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 10–20, 2012.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012

3 Road Side Equipment

RE informs the road side warnings to the vehicles in the road. It should be implemented in the signal area where vehicles gather for the pass. RE should be designed with a database that holds the shortest route for various area RE within the category. Category refers to a collection of RE in a place. When the IMA reaches the RE, it will get updated. RE should capable of holding the shortest path to various areas, checks for traffic in the scanning area, raising the flag TF when there is traffic and updates the IMA.

In figure 1, the inner circle represents RE and the outer circle represents the area of calculating the traffic area. RE scans the number of vehicles in the region of traffic. Vehicles are represented as nodes in the network.

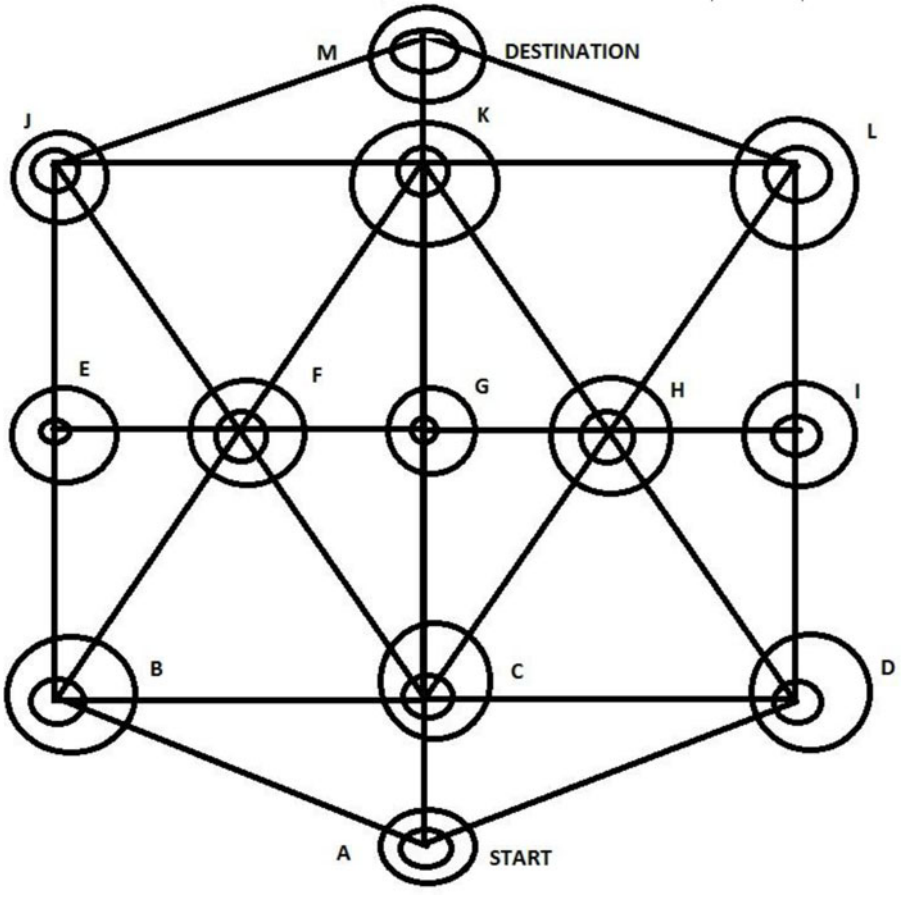

Fig. 1. Category for IMA

3.1 Algorithm

```
Begin
Variable TF = false
Variable Crictical_pt
If Number of nodes>0
Crictical_pt =  Area of coverage / (Average vehicle area* Number of nodes)
Else
Crictical_pt= Area of coverage
If critical_pt < threshold_value
TF=true
End
```

RE raises the TF by calculating the above algorithm. The critical_pt is a value that represents the concentration of traffic in the area. It depends on the area covered by the RE, the average space occupied by the vehicle depending upon the type and number of nodes (vehicles).

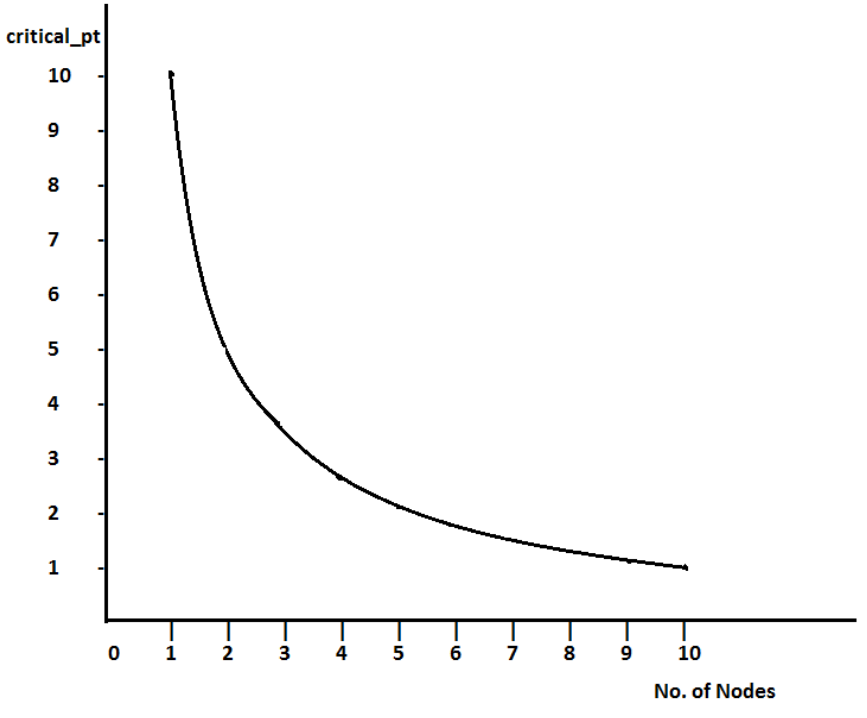

Fig. 2. Graph between critical poin and no. of nodes

Figure 1 describes a category with various RE. If the unit value is considered for average vehicle area and 10 m^3 as area of coverage, the graph between critical_pt and number of vehicles will be as follows.

The figure 2 shows the relation between the critical_pt values that decides to raise the TF.

RE calls the algorithm when an IMA strikes it and remains the state of TF for next collision. The database of RE should contain the shortest path to reach destination and the RE connected to it with distance.

4 Intellect Mobile Agent

IMA is a mobile agent that is generated in the starting point from the RE. IMA initially reads the shortest route to reach destination from the RE in Starting point [1]. IMA migrates to the next RE listed in the shortest path. When it reaches the next RE, it checks for the TF. If the flag is not raised, the process is continued[4]. Else the IMA clone itself and migrate to the neighbor RE. The process is continued and if IMA reaches the destination, it returns back to the starting RE. IMA holds the data about destination path with short distance and traffic free. RE instructs to the vehicle about the path and directs it to reach the destination. IMA is designed effectively using Agent Communication Language (ACL) which is a descriptive KQML-like language for agent communication[2].

4.1 Algorithm

```
Begin
Var
Drop=false
      Clone=false
      Clone_value=0
      Short_path= shortest path from start to destination
      Neighbor_re = next  RE in short_path
      Distance=0
      Previous_re= Current RE
      Sh_distance= distance to reach destination (short-
      path)
repeat
Previous_re= Current RE
Move to Neighbor_re
Distance= distance of neighbor RE
```

```
if RE == destination
return to Starting RE
if TF = true
if cloned==true
Drop the IMA
break
return to Previous_re
Clone IMA
Clone Value= number of neighbor RE -1
Neighbor_re = one of neighbor not cloned for
Else
Neighbor_re= next RE in shortest path
If Distance > 2* sh_distance
Informs to user
Until RE==destination
End
```

4.2 Illustration

Consider a scenario in which a vehicle prepare to start from A to reach M. Vehicle communicate to the RE in the A. RE holds the shortest path to reach the destination M. Since the RE is not a dynamic node, the shortest path will not be alerted regularly. So, it holds the shortest path for various destinations in a category.

Let the database of RE in A holds data as follows:

Table 1. Neighbor node

Neighbor Node
B
C
D

Table 2. Database of RE to serve IMA

Destination	Shortest path	Distance (m)
B	B	1500
C	C	150
D	D	1640
E	B-E	2200
F	C-G-F	1100
G	C-G	900
H	C-G-H	1200
I	D-I	2100
J	B-E-J	2700
K	C-G-K	1900
L	D-I-J	2780
M	C-G-K-M	3120

The Re in the A starts sending the IMA with following initializations for the algorithm.

Drop=false
Clone=false
Clone_value=0
Short_path= C-G-K-M
Neighbor_re = C
Distance=0
Sh_distance= 3120

Let the TF is raised in B,I,F,K which is indicated by a cross in the diagram.

In figure 3 arrow represents the flow of IMA through RE. Stroked arrow shows that the IMA is dropped due to the raise of TF.

A vehicle start at a point A and it communicate with the RE which creates the IMA with initializations to reach destination M . The IMA moves to the next RE as in the Neighbor_re value such as to C. IMA checks for the TF, since TF is not raised because of no traffic at C. So IMA updates the distance parameter and migrates to the next RE at G. Same process is done at G. It migrates to K which is TF raised RE. When IMA identifies that TF flag is raised, it returns to the previous_re value to G. Here IMA is cloned for number of neighbor RE -1 and migrates to all RE nodes. The original node is send to K to make a check for TF flag. If the TF flag is raised, then the IMA is return back. Since it is already cloned, it is dropped. The IMA which migrates to the node H may update the distance parameter. Likewise the IMA reaches the destination node M with calculated path and distance. If the path is higher than twice of the shortest path, it informs the user about it. User wishes to drop IMA.

Consider if the node F and J is not raised with TF flag. The IMA cloned at G also migrate through F-J and reach the destination M. But the updated distance may be higher than the IMA through H-L-M. Both IMA reaches the user and user decides the path.

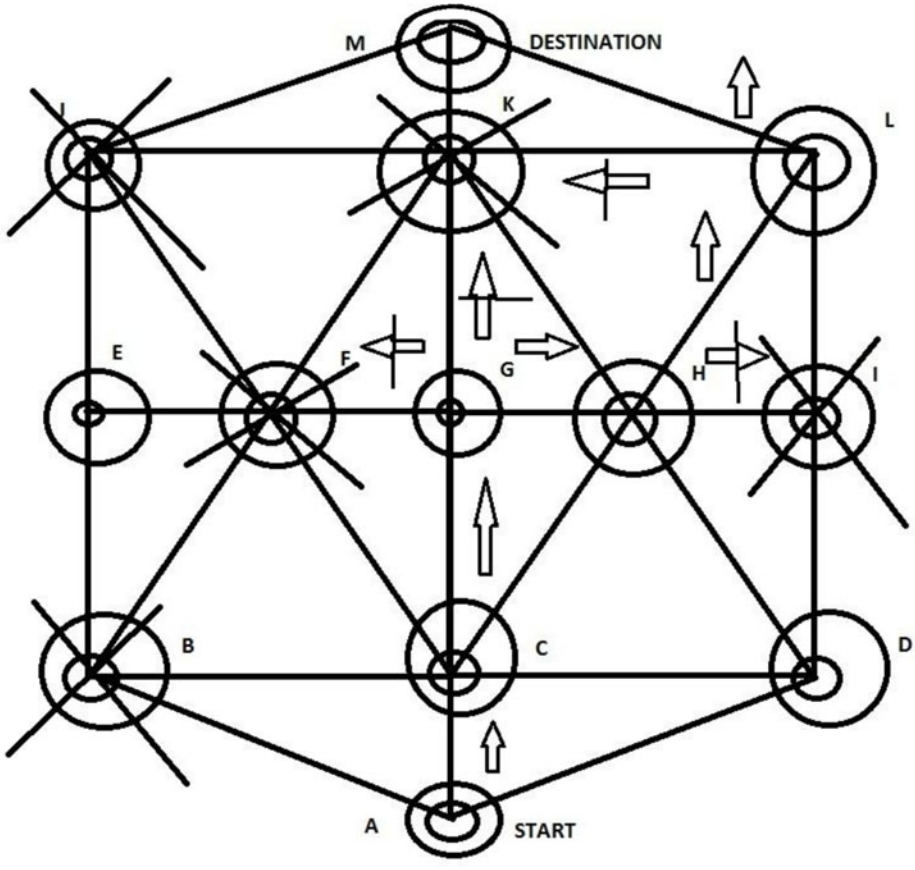

Fig. 3. Category with TF raised RE

The process of identifying the path is updated whenever the vehicle reaches the RE and the RE is initialized as starting RE. This is because of constant change in the traffic flow[3].

4.3 Linker RE

When a user need to travel to a long distance beyond a category, linking RE plays vital role to connect various categories. The RE which connects the two categories is known as linking RE. The database of Linking RE holds the category that connects and shortest path that connects other categories. When a user requests the shortest traffic path to reach the destination which is in another category, there is a need of one mechanism to identify the linking RE to reach the destination. A mobile agent is required for this task and is called as Identify Linking RE Mobile Agent (ILRE). ILRE calculates the linking RE which has the least short path to reach destination and returns the sub destination in each category [5].

The database of each RE contains the location RE and the category region that covers it. The user should choose the region manually when the destination is not in the category.

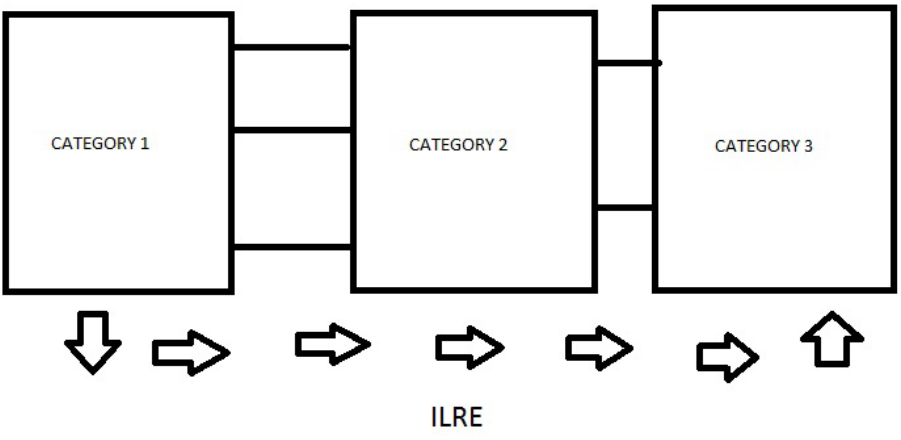

Fig. 4. Flow of ILRE through categories

The figure 4 describes the ILRE through the categories and arrow indicates the flow.

4.4 Role of ILRE

- Identifies the shortest path.
- Detects the Linking RE between categories which links the required category.
- Direct the IMA to reach the linked RE as the destination.
- When it reaches the Linking RE, destination of the IMA changes to the next Linking RE or destination RE (if it reaches the same category).

Algorithm

```
Begin
Var
Dest= destination RE
Clone_value= no. of linking RE in category
Status = false
Dest_Cat= The category that has destination RE
Clone LRE to clone_value
Get shortest path to Linking RE
```

```
Move to Linking RE
Read status
If status=true
Inform user
Else
Drop
Clone IMA
Repeat
IMA destination=Linking RE
Move along IMA
Store IMA path
Alter IMA destination
If category= dest_cat
IMA destination=destination RE
Else
IMA destination= linking RE
Until category =dest_cat
Inform path to User
End
```

Illustration

The ILRE clone itself to the number of linking RE in a category. Each ILRE reaches the Linking RE and checks for the way to reach the destination RE. If the Linking RE pays way to reach the category to reach destination, it access the shortest path to reach and reports to the user else the ILRE is dropped [6].

The returned ILRE creates an IMA to reach the linking RE by assigning the destination of the IMA as linking RE. Using IMA algorithm, the shortest path to reach the Linking RE is achieved. ILRE now alters the destination of the IMA and stores the shortest path identified by the IMA. The process continues until the destination is reached. When the Mobile Agents are in destination category, ILRE sets the destination RE as the destination of the IMA. The entire path stored is informed to the user and when each RE is reached, IMA updates the path since traffic is dynamically altered. Figure 5 explains the detail migration path of Mobile Agents.

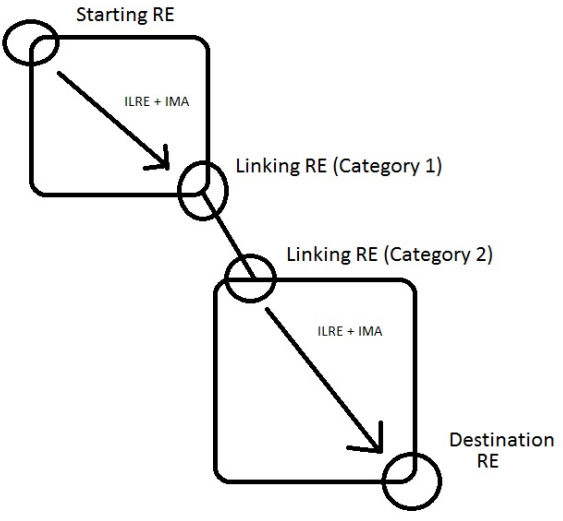

Fig. 5. Migration of ILRE + IMA

5 Conclusion and Future Work

- Flows of vehicles are reduced and direct them in such a way to reduce the traffic [7].
- The user may reach the destination in short time.
- Extends the use by uniting the small cities as category.
- Computational load.

The category plays a vital role to unite various cities together and provide many ways for the mobile agent to travel around. Computational load is more to calculate mobile agent to define next category. Future work involves in minimizing the computational load and enhances the interaction with the user. To enhance the implementation that should have interaction with user in a way to suggest the path with accident parameters to determine the path to destination.

References

[1] Tonguz, O., Wisitpongphan, N., Bai, F., Mudalige, P., Sadekar, V.: Broadcasting in VANET. In: Mobile Networking for Vehicular Environments (2007)
[2] Lange, D.B., Aridor, Y.: Agent Design Patterns Elements of Agent Application Design. In: Proceedings of Autonomous Agents. ACM Press (1998)
[3] Rogers, A., Jennings, N.R., Corkill, D.D.: Agent Technologies for Sensor Networks (2009)

[4] Osborne, M.A., et al.: Towards Real-Time Information Processing of Sensor Network Data Using Computationally Efficient Multi-Output Gaussian Processes. IEEE Press (2008)
[5] Rehák, M., Pechoucek, M., Grill, M., Stiborek, J.: Adaptive Multiagent Sys-tem for Network Traffic Monitoring (2009)
[6] Movaghar, S.R.: Universal vehicular ad-hoc network design: The case for dynamic addressing and relative positioning system
[7] Prakash, A., Tripathi, R.: Vehicular ad hoc networks toward Intelligent Transport Systems