Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Towards Efficient Congestion Free Routing on Large Road Network Using Group Detection Algorithm

S. Balaji, V. Vasantharaji, R. Ramachandiran
MCA, Pondicherry University
India

Abstract: A Vehicular Ad hoc Network (VANET) is a type of mobile Peer-To-Peer wireless network that allows providing communication among nearby vehicles and between vehicles and nearby fixed roadside equipment. A VANET is a spontaneous Peer-To-Peer (P2P) network formed by moving vehicles. As any other MANET (Mobile Ad-hoc Networks), a VANET has no central infrastructure, which implies the need of self-management in a distributed environment where nodes have to adapt to unpredictable changes. As a solution to decrease the number and size of packets exchanged among vehicles because by using groups, VANETs can be split in small sub-VANETs that allow to avoid sending the same information through different paths. In this way, the proposal improves the efficiency and safety of communications through a hybrid model that combines symmetric and asymmetric cryptography. To reach this goal, nodes must know how to behave depending on their state, so this paper provides a full description of each group management process and of how to deal with the information within a group.

Keywords-VANET; Groups; P2P; wireless networks;

I. INTRODUCTION

The Automotive industry is currently undergoing a phase of revolution. Today, a vehicle is not just a thermo-mechanical machine with few electronic devices; rather, recent advancement in wireless communication technologies has brought a major transition of vehicles from a simple moving engine to an intelligent system carrier. A wide spectrum of novel safety and entertainment services are being driven by a new class of communications that are broadly classified as vehicle-to-vehicle communication and vehicle-to-infrastructure communication. Components provide a wide range of services such as freeway management, crash prevention and safety, driver assistance, and infotainment of drivers and/or passengers [1]. Recent trends swing toward advertisement, marketing, and business of services and products on wheels [2]. Consequently, these applications appear to be very lucrative and promising in terms of commerce and research.

The significant use of vehicular communications in safety and infotainment applications has resulted in the development of a new class of media access control and network layer protocols. The current domain of vehicular research includes routing, congestion control, collision avoidance, safety message broadcast, vehicular sensing, security, etc.

A routing protocol has to key on some parameters to decide the routing path. When the routing path is the shortest distance path, it may involve a very high number of changes of directions, resulting in higher hop counts. If the connectivity is chosen as the parameter, the most connected road segment would be overcrowded by frequently routing data packets through the same path. As a consequence, the data packets experience longer queuing delays. A third approach suggested in the literature involves broadcasting request messages to fetch the destination position information and connectivity information. However, in a city, flooding is not advisable as multiple nodes would probe for destination position and connectivity information. As a result, every blind search (i.e., flooding) would disrupt all the ongoing communications. However, packet congestion will occur as the path with the highest connectivity may be used by multiple source–destination (src–dst) pairs. Hence, we specify a connectivity threshold, and paths having connectivity parameter beyond this threshold are assigned the same connectivity status. Apparently, the multi constrained optimal path finding problems are known to be NP-hard problems.

Thus, we develop an approximation algorithm to choose a path based on both hop count and connectivity. Apart from the routing algorithm, we introduce a back-bone mechanism in which some specialized nodes perform functions such as tracking the movement of end nodes, detecting void regions on road segments, storing packets on unavailability of forwarding nodes, and selecting the most suitable intersection node as the forwarding node. Since the routing algorithm selects a path using destination position, we employ a unicast request-reply-based destination probing mechanism. To implement this approach, we divide the city into many zones that are outlined by the multilane road structures. Some dense intersections (identified as the meeting point of multiple road segments) on the boundary of the zones are chosen as the boundary intersections.

II. RELATED WORKS

The VANET has witnessed several endeavors toward the development of suitable routing solutions. Originally, many routing protocols were solely designed for mobile ad hoc networks and later enhanced to suit the VANET scenarios

[5], [6], [7]. Later on, few novel protocols were developed for adverse VANET environments [3]–, [4], [5], currently, researchers are working on a more concrete version of routing protocols with a higher performance index. However, noteworthy pioneering works such as greedy perimeter stateless routing (GPSR) [5], greedy perimeter coordinator routing (GPCR) [6], geographic source routing (GSR) [4],vehicle assisted data delivery (VADD), anchor-based street-and traffic-aware routing (A-STAR) [8], connectivity-aware routing (CAR) [2], [3] greedy traffic-aware routing (GyTAR) [4], road-based using vehicular traffic (RBVT), static-node-assisted adaptive data dissemination in vehicular networks (SADV), etc. have laid the foundation for routing in VANETs.

A. Intersection Node Probing Problem

In city environments, intersections play crucial roles for data communications. As the intersection region is comparatively small and the probability of change of direction is very high, it will be risky to choose an unstable node as the forwarding node from this region. According to us, the node that crosses the intersection before actually receiving a data packet is termed as the unstable node. This happens when a vehicle speeds up after sending its beacon packet. Vehicle flows are controlled by traffic lights. Let us consider that four road segments meet at an intersection. All the incoming vehicles of two road segments may be blocked by the red signal, whereas vehicles on the other two road segments flow until the green signal is on. When a vehicle crosses the intersection without having another vehicle arrive at the intersection, a disconnection may occur. Such a situation arises only when a fleet of vehicles has crossed the intersection and when another fleet of vehicles has not been arrived at the intersection. Protocols [2] that ensure connectivity on the routing path are not affected by this problem. Although CAR [3] addresses connectivity issues, it could be affected as the average connectivity [2] does not ensure connectivity in individual road segments in a routing path. Protocols such as GPSR, GPCR, and GSR do not ensure connectivity, and hence, the foregoing problem can have a serious impact on their performances.

B. Location Service Requirement Problem

Traditionally, position-based routing protocols are assumed to be aware of the destination position through location services [1].However, fetching the real-time position information of the source or the destination is nearly impossible as that information has to travel a number of hops in a city area, which is generally very large in size. Further, reducing the end-to-end delay is crucial for any routing protocol. Moreover, the location service is found to be superfluous for the nodes that do not take part in any communication. Protocols like GPSR [5] and GPCR [6] take the aid of proactive location services like hierarchical location service (HLS) [1] and grid location service (GLS) [4]. In these location services, the lower beacon interval is the key factor for higher accuracy. Apparently, increased beacon messages create havoc in dense city scenarios [5]. Although the reactive location service [2] used in GSR [4] is an exception, neither the source nor the destination can keep a tab on each other if they change their position in the middle of data communications. As GSR [4] relies on flooding to probe the destination position [2], it also suffers from drawbacks [2], [5] induced by flooding. As far as GyTAR [4] is concerned, it uses CitySense [7] as the tool for location services. Sensors are deployed at the intersections to provide the actual position information of the destination. Aggregating and disseminating the position information throughout the entire network involve both computational and communication overheads. For every minor movement of the destination, there is a need for the computation of a new path to the destination from the intermediate intersection. As a result, the hop count may be increased.

C. Distance- or Connectivity-Based Weighted Graph Problem

When a typical ad hoc network scenario is mapped into a graph, the nodes are considered as vertices, and the communication link between two nodes is considered as an edge. The weight of each edge is represented by the Euclidean distance between the end points. Due to the presence of high rise buildings in city environments, vehicles are unable to communicate with their peers even if the Euclidean distance between them is less than the transmission range. Therefore, in vehicular networks, intersections are mapped to vertices, and the road segments between adjacent intersections are mapped to edges. Generally, these edges are assigned weights based on distance [4] (i.e., path distance) or connectivity [8] (i.e., the edge having the highest connectivity is assigned the smallest weight), or both [24].

III. PROPOSED WORK

A. Groups

A group in a VANET is defined as a set of vehicles that are located in a close geographic area whose formation is determined by the mobility pattern of vehicles. The group needs a minimum of vehicles and is managed by a given node called"leader of the group". All vehicles forming part of a group have a direct wireless connection with the leader of such a group and share a secret key. There are several bibliographic references that propose the use of groups or clusters, which are the same in VANETs. [9] Presents a theoretical analysis of a directional stability based clustering algorithm. [5] describes clusters where the leader is the node in the middle with the lowest identifier. [7] Proposes clusters to maximize the advance of the relayed information and to avoid interferences, but there the head cluster must know the exact positions of nodes in the cluster. None of these works define in detail the processes that nodes have to complete for group management and do not show any implemented scheme to demonstrate the reliability of obtained data [1], which are the main objectives of this work.

On the other hand, where the number of vehicles is low and there is no saturation of communications, the groups are not used. With a group scheme would be generated 3 connections per group for every data. The first one goes from the vehicle which produces the information to the leader, then, the leader launches a multicast to all vehicles of the group. Finally, another connection between the leader and another vehicle (in the best position) continues multicasting the information. Therefore it will be generated (n=numberofgroups)_3 for each data packet. Vehicles will form groups according to dynamic cells where the leader is the vehicle with VANET technology that has initiated the group or that has the greatest number of neighbors when the previous leader falls below an established threshold for group formation. The definition of these groups will be based on the average speed of the route and the direction in which vehicles circulate, so that vehicles that circulate at a speed near that one will not change group during their journey on that route. The group leader will be the one in charge of managing the information and connections.

B. Group Stages

We distinguish among several stages in group management, corresponding to different situations of vehicles, depending on the route and on their status in each moment. The stages are: Detection, Election, Creation, Membership and Life of a group. VANETs are wireless networks where there are a large number of highly volatile connections between vehicles. For this reason it is necessary to define in detail the way in which vehicles must act according to their situation. The global network life scheme proposed in this paper is as follows. Initially all nodes start in the Group Detection stage. After this, they can enter the Creation or the Election stage, depending on the circumstances. After Group Creation, the node would be the group leader, while after Group Election; the node would proceed to Group Membership.

i. Group Detection

This is the first stage, where vehicles are in normal conditions without dense traffic. From time to time the vehicle checks the number of neighbors and the number of leaders among them. If there is at least one neighbor who is leader of a group, the node proceeds to the Election stage, and otherwise to the Creation stage. This stage does not generate any traffic of control due to the fact that all the necessary information is contained in the beacons that nodes generate.

Algorithm for Group Detection

```
function Group Detection (...)
number Of Neighbors = 0;
number Of Leaders = 0;
while (neighbor(i) exists) do
if (is Leader( neighbor(i) )) then
number Of Leaders = 0;
end
number Of Neighbors++;
i++;
end
if (number Of Leaders == 0) then
Group Creation();
else
Group Election();
end
end function
```

ii. Group Election

This stage starts when the vehicle has found among its neighbors at least one node that is leader of some group. If there is only one neighbor who is a group leader, the choice is automatic. Otherwise, if there are several leaders, the vehicle has to choose one of them to join it. Where group Value denotes a quantity used for the choice and group Leader (j) represents the j-th neighbor of the node that is leader of a group. If there are several leaders among its neighbors, the vehicle chooses one according to the group Value that depends on the following values for each group j:

- Density A(j) of vehicles.
- Average quality of signal B(j) within the vehicles.
- Time C(j) during which it has been connected to the leader.

Algorithm for Group Election

```
function Group Election (...)
if (number Of Leaders # 1)then
j = 1;
e = 0;
group Value[e] = 0;
while (group Leader[j] exists) do
group Value[j] = A(j)+B(j)+C(j);
```

```
if (group Value[j] # group Value[e])then
group Value[e] = group Value[j];
end
j++;
end
else
e = 1;
end if
send Request (group Leader[e]);
receive Group Key(group Leader[e]);
Group Membership();
end function
```

iii. Group Creation

The vehicle is not close to any leader of a group. It should check whether within their neighbors there are at least X nodes that do not belong to any group, plus a variable Y that indicates the number of vehicles that can either turn off, separate or not join the new group that is being created. If the number of neighbors without group is lower than the minimum threshold required for group creation, the vehicle waits a period time1 and starts again the Group Detection stage. Otherwise, if the number of neighbors is greater than the threshold X + Y , the vehicle begins a new Group Creation process. In order to do it, it multicasts a group creation request towards all neighbors with distance equal to 1. Nodes that receive this request respond accepting or rejecting the invitation. If the number of neighbors that accept the invitation is greater than the minimum threshold X, the new group leader sends to each node the secret key of the group encrypted with the public keys of each node. In this moment the new group is formed. Otherwise, the number Y of estimated vehicles is increased by adding the number of vehicles that did not accept the invitation.

Algorithm for Group Creation

```
function Group Creation (...)
if (number Of Neighbors # X + Y )then
Accepted Neighbors = 0;
n = 1;
l = 0;
Multicast Neighbors (Neighbors List[]);
for (n=1; n # number Of Neighbors; n ++)do
Receive Group Election(n);
if (neighbor(n) accept) then
accepted Neighbors(l) = neighbor(n);
l++;
accepted Neighbors++;
end
end
if (accepted Neighbors # X)then
for (n=1; n# accepted Neighbors; n + +)do
Send Group Key(accepted Neighbors(n),
PuK accepted Neighbors(n));
end
Group Life();
else
Y = Y + X – accepted Neighbors;
Wait(time1);
Group Detection();
end
else
Wait(time1);
Group Detection();
end
end function
```

C. Group Membership

Once the group is formed, the leader must periodically validate that the group continues being useful. Otherwise, it would be necessary to change the leader or to end the group. The process where a node leaves the group which it belongs. When the node loses any contact with the leader of the group for certain time, the node stops to belong to its group and begins the Group Detection stage if node density exceeds the corresponding threshold.

Algorithm for Group Membership

```
function Group Membership (...)
if (See( group Leader )) then
Wait(time3);
Group Membership();
else
Wait(time4);
if (See( group Leader )) then
Wait(time3);
Group Membership();
else
final Group Membership();
Group Detection();
end
end
end function
```

D. Group Life

The leader of a group periodically checks that the group is still useful. If group size falls below a certain threshold, the leader checks whether it has a number of neighbors greater or equal to D (dense traffic threshold) and waits for time2 instead of ending the group in order to avoid introducing group management traffic when the vehicle is in a dense traffic situation. If the leader is not in a dense traffic situation, it begins a leader change or a group ending process. First, the leader asks about the neighborhood density in order to know if neighborhood density (number of neighbors of the same group or without any group near) is bigger than X. It also finds out which of its neighbors has the largest number of neighbors. After this, it sends a multicast signal of leader change to all its neighbors. The new leader will begin a Group Creation stage with those nodes without any group that are in its transmission range. In the absence of any neighbor exceeding the threshold, the leader sends the group ending signal through multicast to all its neighbors.

Algorithm for Group Life

```
function Group Life (...)
for (n=1; n# number Of Neighbors; n + +)do
if (Belongs( neighbor(n),group(a) )) then
group Size++;
end
end
if (group Size # X)then
Wait(time2);
Group Life();
else
if (number Of Neighbors # D)then
new Leader=0;
for(n=1;n# number Of Neighbors; n ++)do
//group Size+without Group
pot = potential(neighbor(n));
if ((pot # X)and(pot # group Size))then
group Size=group Size(n);
new Leader=n;
end
end
if (new Leader == 0) then
Multicast (End-of-Group-Signal);
Group Detection();
else
Multicast (Leader-Change-Signal);
//New leader init Group Creation process
Group Detection();
end
end
end
end function
```

IV. CONCLUSION

In this paper, the use of groups has been proposed as a solution to decrease the number of communications in VANETs under dense traffic conditions when the overhead of transmitted data causes a considerable drop in communication quality. In particular, a complete description of the proposed scheme for autonomic group management in VANETs is provided, which includes differentiation among possible vehicle states: from the initial state when it does not belong to any group, to the choice of an existent group to join it, the creation of a new group, and the end of a group. This paper also shows how to proceed with group communications. A complete analysis has been done through simulations using the open source traffic simulator SUMO and network simulator NS-2. Such simulations allow the analysis of the operations at each stage, and a comparison between communication overhead when using groups and without using them in VANETs.

REFERENCES

[1] J. Bernsern and D. Manivannan, "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification," Pervasive Mob. Compute. , vol. 5, no. 1, pp. 1–18, Feb. 2009.

[2] Q. Yang, A. Lim, S. Li, J. Fang, and P. Agrawal, "ACAR: Adaptive connectivity aware routing for vehicular ad hoc networks in city scenarios," Mob. Netw. Appl., vol. 15, no. 1, pp. 36–60, Feb. 2010.

[3] V. Naumov and T. R. Gross, "Connectivity-aware routing (CAR) in vehicular ad hoc networks," in Proc. IEEE INFOCOMM, 2007, pp. 1919–1927.

[4] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in Proc. IEEE Intell. Veh. Symp., 2003, pp. 156–161.

[5] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in Proc. ACM MOBICOM, 2000, pp. 243–254.

[6] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 9, no. 1, pp. 69–72, Jan. 2005.

[7] H. Menouar, M. Lenardi, and F. Filali, "Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks," in Proc IEEE VTC, 2007, pp. 2101–2105

[8] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, and C.Hern´andez-Goya A simulation study of new security schemes in mobile Ad-hoc NETworks, Lecture notes in computer science, EUROCAST 2007:Vol:4739:73-81

[9] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, and C. Hern´andez-Goya, Flexible Authentication in Vehicular Ad hoc Networks, Proceedings of APCC IEEE Asia Pacific Conference on Communications. 2009.

[10] O. Dousse, F. Baccelli, and P. Thiran, Impact of Interferences on Connectivity in Ad Hoc Networks. INFOCOM 2003

[11] P. Fan, P. Sistla, and P. C. Nelson, Theoretical analysis of a directional stability-based clustering algorithm for vanets. Vehicular Ad Hoc Networks 2008:80-81

[12] Y. Gunter, B. Wiegel, and H. P. Gromann, Medium Access Concept for VANETs Based on Clustering. VTC Fall 2007:2189-2193

[13] J. Molina-Gil , P. Caballero-Gil, and C. Caballero-Gil, A Vision of Cooperation Tools for VANETs, Proceedings of the First International Workshop on Data Security and PrivAcy in wireless Networks, WoWMoM, June 2010.

[14] Z. Y. Rawashdeh and S. M. Mahmud, Media Access Technique for Cluster-Based Vehicular Ad Hoc Networks. VTC Fall 2008

[15] M. Raya, P. Papadimitratos, and J.P. Hubaux, Securing Vehicular Communications - Assumptions, Requirements and Principles, Proceedings of Fourth Workshop on Embedded Security in Cars (ESCAR), 2006.

