Using Groups to Reduce Communication Overhead in VANETs

C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil
Department of Statistics, Operations Research and Computing,
University of La Laguna,
Spain
Email: {ccabgil, pcaballe, jmmolina}@ull.es

***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. The lack of centralized infrastructure, high node mobility and increasing number of vehicles in VANETs result in several problems discussed in this paper, such as interrupting connections, difficult routing, security of communications and scalability. Groups are proposed 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

A VANET is a spontaneous Peer-To-Peer (P2P) network formed by moving vehicles. As any other MANET (Mobile Ad-hoc NETwork), 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. Such autonomic networks present unique challenges such as high mobility, real-time constraints, scalability, gradual deployment and privacy.

Intelligent VANETs hybridly integrate multiple ad-hoc networking technologies such as WiFi IEEE 802.11 b/g, WiMAX IEEE 802.16, Bluetooth, etc. to achieve effective wireless communication. Such networks constitute a fundamental part of the Intelligent Transportation System (ITS). Many research projects on ITS are being financed by the European Commission because road safety is classified as a priority objective. Different possible situations exist where communications between vehicles would help to prevent accidents and to avoid traffic jams, which would save time and money, reduce contamination of the environment and consumption of fuel reserves.

Several general characteristics can be considered in wireless networks: authenticity, privacy, anonymity, cooperation, low delay, stability of communications, scalability, etc. [2], [8]. However, when dealing with VANETs, the protection of those properties is an even greater problem due to specific characteristics of these networks, such as very changing scenarios, from local roads with very few vehicles to cities or highways full of vehicles. In this work we propose the use of groups in VANETs, which will allow to optimize communication in dense traffic situations, and to define group secret keys for the use of symmetric cryptography to ensure information exchanges efficiently.

Section II gives a definition of group. In Section III, the different group stages included in our proposal are explained in full detail: Detection, Election, Creation, Membership and Life of a group. Section IV describes how communications are conducted within the group. Sections V and VI analyze simulation results. Finally, conclusions close the paper.

II. 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. [4] 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 is the main objectives of this work.

Groups will be used only when the conditions of the routes require it. Examples are dense traffic, traffic jams or congested highways, where the density of vehicles in a geographic zone causes that the number of communications is huge. But groups are formed before the number of nodes begins to degrade the network. Without any mechanism to minimize the number of communications, a simple broadcast

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-102-1

will be launched from every vehicle generating a lot of unnecessary redundancy. The number of packets generated depends on the number of nodes in the network and interconnection among them. Therefore, it will be generated n packets for each data communications where n is the number of vehicles with On Board Unit (OBU) in the network (in the scope of interest). This number of connections is not extremely large, and perhaps would not need to taking steps to reduce that number, but some studies like [3] showing that many vehicles duplicate data packets causing collisions in the information that is sent, which degrades communication quality.

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.

III. 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.

A. Group Detection

This is the first stage, where vehicles are in normal conditions without dense traffic. This stage is described in Algorithm 1, where $neighbor(i)$ denotes the i-th neighbor of the node that initiates the stage. 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 1 GroupDetection

```
01:function GroupDetection (...)
02:   numberOfNeighbors = 0;
03:   numberOfLeaders = 0;
04:   while (neighbor(i) exists) do
05:      if (isLeader( neighbor(i) )) then
06:         numberOfLeaders = 0;
07:      end
08:      numberOfNeighbors++;
09:      i++;
10:   end
11:   if (numberOfLeaders == 0) then
12:      GroupCreation();
13:   else
14:      GroupElection();
15:   end
16: end function
```

B. 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. Algorithm 2 shows this stage, where $groupValue$ denotes a quantity used for the choice and $groupLeader(j)$ 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 $groupValue$ 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 2 GroupElection

```
01:function GroupElection (...)
02:   if (numberOfLeaders ≥ 1)then
03:      j = 1;
04:      e = 0;
05:      groupValue[e] = 0;
06:      while (groupLeader[j] exists) do
07:         groupValue[j] = A(j)+B(j)+C(j);
08:         if (groupValue[j] ≥ groupValue[e])then
09:            groupValue[e] = groupValue[j];
```

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-102-1

```
10:         end
11:         j++;
12:       end
13:     else
14:       e = 1;
15:     endif
16:     sendRequest (groupLeader[e]);
17:     receiveGroupKey(groupLeader[e]);
18:     GroupMembership();
19:end function
```

Once the group has been chosen, the vehicle sends a login request encrypted with its public key to the group leader. After authenticating it, the leader sends the group secret key encrypted with such a public key and from then, the vehicle becomes part of the group.

C. Group Creation

In the Group Creation stage (Algorithm 3), 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 3 GroupCreation

```
01:function GroupCreation (...)
02:   if (numberOfNeighbors ≥ X + Y)then
03:       AcceptedNeighbors = 0;
04:       n = 1;
05:       l = 0;
06:       MulticastNeighbors (NeighborsList[]);
07:       for (n=1; n ≤ numberOfNeighbors; n++)do
08:          ReceiveGroupElection(n);
09:          if (neighbor(n) accept) then
10:             acceptedNeighbors(l) = neighbor(n);
11:             l++;
12:             acceptedNeighbors++;
13:          end
14:       end
15:       if (acceptedNeighbors ≥ X)then
16:          for (n=1; n≤ acceptedNeighbors; n++)do
17:             SendGroupKey(acceptedNeighbors(n),
18:             PuKacceptedNeighbors(n));
19:          end
20:          GroupLife();
21:       else
22:          Y = Y + X - acceptedNeighbors;
23:          Wait(time1);
24:          GroupDetection();
25:       end
26:   else
27:       Wait(time1);
28:       GroupDetection();
29:   end
30:end function
```

In conclusion, this stage requires: a multicast of invitation to join the new group, unicast responses from n users and a multicast to relay a message that enables the members to build the group secret key. This means a total of $2n + 1$ packets in case of positive group creation, and $n + 2$ if the process fails. The Group Creation starts when the appropriate number of neighbors reaches a certain threshold of traffic, but without to be dense traffic. Consequently, management packets generated at this stage not increase communications in dense traffic conditions.

D. 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.

Algorithm 4 shows 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 4 GroupMembership

```
01:function GroupMembership (...)
02:   if (See( groupLeader )) then
03:       Wait(time3);
04:       GroupMembership();
05:   else
06:       Wait(time4);
07:       if (See( groupLeader )) then
08:          Wait(time3);
09:          GroupMembership();
10:       else
11:          finalGroupMembership();
12:          GroupDetection();
13:       end
14:    end
15: end function
```

E. Group Life

Algorithm 5 shows how 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 5 GroupLife

```
01:function GroupLife (...)
02:   for (n=1; n≤ numberOfNeighbors; n++)do
03:      if (Belongs( neighbor(n),group(a) )) then
04:         groupSize++;
05:      end
06:   end
07:   if (groupSize ≥ X)then
08:      Wait(time2);
09:      GroupLife();
10:   else
11:      if (numberOfNeighbors ≤ D)then
12:         newLeader=0;
13:         for(n=1;n≤ numberOfNeighbors; n++)do
14:           //groupSize+withoutGroup
14:           pot = potential(neighbor(n));
15:           if ((pot ≥ X)and(pot ≥ groupSize))then
16:              groupSize=groupSize(n);
17:              newLeader=n;
18:           end
19:         end
20:         if (newLeader == 0) then
21:           Multicast (End-of-Group-Signal);
22:           GroupDetection();
23:         else
24:           Multicast (Leader-Change-Signal);
25:           //New leader init GroupCreation proccess
26:           GroupDetection();
27:         end
28:      end
29:   end
30:end function
```

IV. Message Management inside Groups

By using groups the number of communications can remarkably decrease without missing any useful information.

Algorithm 6 shows the steps that a vehicle beloging to a group must follow in order to process an input signal.

Algorithm 6 Message Management inside Groups

```
01:function MessageManagement (...)
02:   if(AmIfinalDestination(packet)) then
03:      TreatData(packet);
04:   else
05:      if (AmILeader()) then
06:         if (IsPublicInformation(packet)) then
07:            TreatData(packet);
08:            Multicast(packet, GroupKey);
09:         else
10:         relayer = estimatePosition(DestinationNode);
11:            Unicast (packet, relayer);
12:         end
13:      else
14:         if(IsForwardingSequence(packet)) then
15:         relayer = estimatePosition(DestinationNode);
16:            Unicast (packet, relayer);
17:         end
18:      else
19:         if(IsSentbyLeader(packet)) then
20:         relayer = estimatePosition(DestinationNode);
21:            Unicast (packet, relayer);
22:         end
23:      else
24:         Unicast (packet, GroupLeader);
25:      end
26:   end
27: end function
```

If the node is the final destination, it simply processes the information. Otherwise, it checks whether data were sent by the group leader. In particular, the leader can send two types of packets towards any member of the group that is not the final destination of the data:

- A connection of a vehicle to Internet services, or any other supplied service where it is necessary a relay of an information sequence,
- A packet of other type of information that must be forwarded towards other parts of the network.

With respect to this second type of packets there are two types of communications that must be differentiated:

- safety-related information
- commercial advertising

In both cases the vehicle belonging to the group that receives or produces the communication, sends it to the group leader who will forward it to all connected members of the group and towards the zones where the message has not been yet spread.

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-102-1

An Internet connection can be passed through to another group through intermediate nodes who forward the information. If one vehicle wants to connect to the Internet, it sends a request towards a vehicle outside its group, which will forward the request towards its leader. The leader will send the request towards other group or the RSU (Road Side Unit), which will answer by giving some details about the transmission such as the number of packets required for the connection. With this information, and knowing both the location and the speed of the vehicles in the group and of the vehicle that wants to connect, the leader calculates how long is the connection between the RSU, the intermediates vehicles and the destination vehicle. Then, it balances the load of connection so that the packets get to the destination as quickly as possible. Once the leader has informed the intermediate vehicles, they connect with the RSU and with the connected vehicles. After this, they relay the Internet connection.

For these types of communications, mechanisms for enforcement cooperation [6] are necessary because without them, intermediate vehicles would not have the necessary incentives to relay others connections, what would disable any type of service that incorporates an indirect connection with the RSU.

V. Simulation

Both the feasibility and effectiveness of our approach are shown through the figures where a simulation exemplifies its performance. In the first part of our demonstration (Figure 1), a NS-2 and SUMO display shows the VANET state in one moment when groups are operating.

The most relevant options selected for the demonstration have been: Total number of vehicles: 80, number of vehicles with OBUs: 80, number of lanes for each direction: 3 and 3, simulation time: 100 seconds, moment when retransmissions begins: 40 seconds, retransmission period: 10 seconds, distance relay nodes: 75 meters, traveled distance before the traffic jam happens: 800 meters.

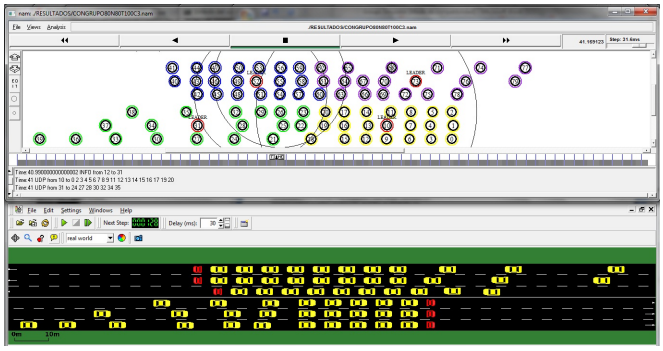

Figure 1. Simulation

The implemented simulations for groups consider four levels of development: vehicle mobility, node energy, group formation and P2P communications in the network.

- The vehicle mobility layer manages the node movement in the movement pattern, which defines roads, lines, different speed limits for each line, traffic jams, etc.
- The node energy layer is used to distinguish between vehicles with and without OBUs. Vehicles without OBUs are present in the road but do not contribute in the communications.
- The group formation layer defines which vehicles belong to each group, who is the leader of each group, who generates traffic information and who relays information to other groups.
- The P2P communications layer is responsible for the definition of which nodes are in the transmission range of the retransmitting node at any time.

Statistics extraction. Simulations give essential statistics such as number of generated, dropped or lost packets or bytes. These basic statistics data are useful to make efficient simulations for large scale scenarios.

Two implementation mechanisms. Simulations provide two mechanisms to implement VANETs: One with groups and the other without them. The implementation without groups does not involve the group formation layer while the implementation with groups allows comparing the behavior and data of both types of simulations.

VI. Analysis Results

The implemented simulations with groups can be compared with results obtained from the simulation without the use of groups with the same topology (see Figure 2). This helps to illustrate the vehicular P2P network evaluation.

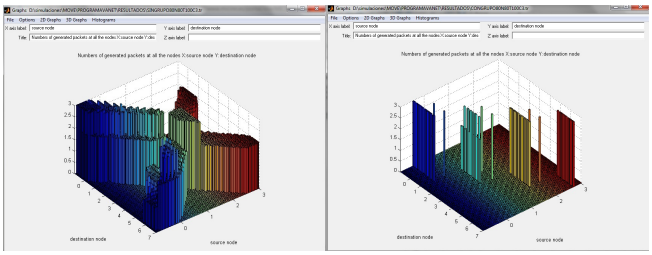

Figure 2. Generated packets without and with Groups

Among the obtained information from the simulations we have the number of packets and bytes generated, sent, broadcast, received, lost, etc. for each node. Also, other general information shown is the number of packets generated or lost in the whole network, the number of formed groups, which nodes are the leaders of the groups, which nodes generate packets and which nodes forward them, etc. In addition to all this information, another interesting aspect is that it provides a detailed simulation of what happens in each moment in the VANET thanks to the use of the NS-2 display. It also shows the traffic model through the SUMO tool while the information is represented using TraceGraph.

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-102-1

Table I
SIMULATION RESULTS

VARIABLES		USING GROUPS		WITHOUT GROUPS		
number of vehicles	vehicles with OBU	generated packets	loss packets	generated packets	loss packets	groups formed
60	10	278	107	167	12	1
31	15	598	402	277	63	2
40	20	825	443	351	0	2
31	31	2343	1804	638	139	2
40	40	2805	2014	932	135	2
50	50	5077	3981	1101	182	2
100	50	3539	2350	1327	68	2
60	60	5732	4415	1314	199	3
80	80	6675	4529	2120	215	4

Table I shown some result of simulations. We have chosen to use the following set of parameters to be varied in order to study the network behavior under different conditions: simulation time: 100 seconds, retransmission period: 15 seconds, distance relay nodes: 75 meters, 3 lines for each direction: 3 and 3, moment when retransmissions begins: 40 seconds, maximum number of hops: 1, routing protocol: DSDV, traveled distance before the traffic jam happens: 800 meters. The remaining variables are indicated in the table. Finally, the values of all parameters which are not explicitly mentioned are set equal to the different simulations.

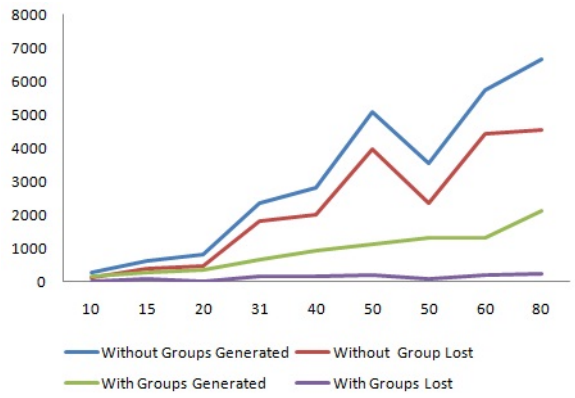

Figure 3. Generated packets

We can observe in Figure 3 the comparison between the average generated and lost packets: it is clear that, without the use of Groups in VANETs, the number of generated packets grow up much faster than with the use of Groups. But also the lost packets grow up much faster. The main reason is likely to be the heaviest traffic load that VANETs generates in traffic jams conditions: indeed, the original protocol makes a massive use of broadcast operations. The use of Groups will help to decrease the percentage of lost packets and to perform the VANETs operation.

VII. 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.

ACKNOWLEDGMENT

Research supported by the Ministerio de Ciencia e Innovación and the European FEDER Fund under Project TIN2008-02236/TSI, and by the Agencia Canaria de Investigación, Innovación y Sociedad de la Información under PI2007/005 Project.

REFERENCES

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

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

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

[4] 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

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

[6] 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.

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

[8] 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.

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-102-1