KNOWLEDGE MANAGEMENT USING CLUSTERS IN VANETS
Description, Simulation and Analysis

C. Caballero-Gil, P. Caballero-Gil and J. Molina-Gil
Department of Statistics, Operations Research and Computing, University of La Laguna, La Laguna, Tenerife, Spain

Keywords: VANET, Cluster, Knowledge management, Wireless ad-hoc networks.

Abstract: Vehicular Ad Hoc Network (VANET) is a type of mobile network that allows Knowledge Management of the road providing communication among nearby vehicles and between vehicles and nearby fixed roadside equipment. The lack of centralized infrastructure and high node mobility and number of vehicles generate problems such as interrupting connections, difficult routing, security of communications and scalability. Clusters are here proposed as a solution to avoid data collisions by decreasing the number of connections exchanged among vehicles and to reach this goal, nodes must cooperate, and form or join clusters depending on their state. This paper provides a global vision of the life cycle of cooperative nodes who form clusters and a description of how to deal with the information within a cluster. Simulation results show that the proposed scheme reduces the number of communications, avoiding data loss due to collisions. This paper provides a full description of each cluster management process and of how to deal with the information within a cluster.

1 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 P2P 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). Different possible situations exist where Knowledge Management 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 characteristics can be considered in wireless networks: authenticity, privacy, anonymity, cooperation, low delay, stability of communications, scalability, etc. (Caballero-Gil *et al.*, 2009), (Raya *et al.*, 2006) 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 clusters in VANETs, which will allow to optimize communication in dense traffic situations.

2 CLUSTERS

A cluster 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 cluster needs a minimum of vehicles and is managed by a given node called "leader of the cluster". All vehicles forming part of a cluster have a direct wireless connection with the leader of such a cluster and share a secret key.

There are several bibliographic references that propose the use of clusters in VANETs. (Fan *et al.*, 2008) presents a theoretical analysis of a directional stability-based clustering algorithm. (Gunter *et al.*, 2007) describes clusters where the leader is the node in the middle with the lowest identifier. (Rawashdeh and Mahmud, 2008) proposes a different scheme of clusters to maximize the advance of the relayed information and to avoid interferences. None of these

Caballero-Gil C., Caballero-Gil P. and Molina-Gil J..
KNOWLEDGE MANAGEMENT USING CLUSTERS IN VANETS - Description, Simulation and Analysis.
DOI: 10.5220/0003070501700175
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2010), pages 170-175
ISBN: 978-989-8425-30-0

works define in detail the processes that nodes have to complete for cluster management and do not show any implemented scheme to demonstrate the reliability of obtained data, (Caballero-Gil *et al.*, 2007), which is a priority in this work.

Clusters 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 clusters are formed before the number of nodes begins to degrade the network. Without any mechanism to minimize the number of communications, a simple broadcast 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). Some studies like, (Dousse *et al.*, 2003) show that this number of connections causes collisions in the information that is sent, which degrades communication quality.

3 MESSAGE MANAGEMENT INSIDE CLUSTERS

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

Algorithm 1 shows the steps that a vehicle beloging to a cluster must follow in order to process an input signal.

If the node is the final destination, it simply processes the information. Otherwise, it checks whether data were sent by the cluster leader. In particular, the leader can send two types of packets towards any member of the cluster 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

Algorithm 1: Message Management inside Clusters.

```
function MessageManagement (...)
  if(AmIfinalDestination(packet)) then
    TreatData(packet);
  else
    if (AmILeader()) then
      if (IsPublicInformation(packet)) then
        TreatData(packet);
        Multicast(packet, ClusterKey);
      else
      relayer =
      estimatePosition(DestinationNode);
        Unicast (packet, relayer);
      end
    else
    if(IsForwardingSequence(packet))then
      relayer =
      estimatePosition(DestinationNode);
        Unicast (packet, relayer);
      end
    else
      if(IsSentbyLeader(packet))then
      relayer =
      estimatePosition(DestinationNode);
        Unicast (packet, relayer);
      end
    else
      Unicast (packet, ClusterLeader);
    end
  end
end function
```

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

An Internet connection with a RSU (Road Side Unit) can be relaying through another clusters using intermediate nodes who forward the information. For these types of communications, mechanisms for enforcement cooperation, like in (Molina-Gil *et al.*, 2009) 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.

4 CLUSTERS STAGES

We distinguish among several stages in cluster 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 cluster.

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 Cluster Detection stage. After this, they can enter the Creation or the Election stage, depending on the circumstances. After Cluster Creation, the node would be the cluster leader, while after Cluster Election, the node would proceed to Cluster Membership.

4.1 Cluster Detection

This is the first stage, where vehicles are in normal conditions without dense traffic. This stage is described in Algorithm 2, 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 cluster, 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 2: ClusterDetection.

```
function ClusterDetection (...)
   numberOfNeighbors = 0;
   numberOfLeaders = 0;
   while (neighbor(i) exists) do
      if (isLeader( neighbor(i) )) then
         numberOfLeaders = 0;
      end
      numberOfNeighbors++;
      i++;
   end
   if (numberOfLeaders == 0) then
      ClusterCreation();
   else
      ClusterElection();
   end
end function
```

4.2 Cluster Election

This stage starts when the vehicle has found among its neighbors at least one node that is leader of some cluster. If there is only one neighbor who is a cluster leader, the choice is automatic. Otherwise, if there are several leaders, the vehicle has to choose one of them to join it. Algorithm 3 shows this stage, where *clusterValue* denotes a quantity used for the choice and *clusterLeader*(j) represents the j-th neighbor of the node that is leader of a cluster. If there are several leaders among its neighbors, the vehicle chooses one according to the *clusterValue* that depends on the following values for each cluster j:

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

Algorithm 3: ClusterElection.

```
function ClusterElection (...)
   if (numberOfLeaders ≥ 1) then
      j = 1;
      e = 0;
      clusterVal[e] = 0;
      while (clusterLeader[j] exists) do
         clusterVal[j] = A(j)+B(j)+C(j);
         if(clusterVal[j] ≥ clusterVal[e]) then
            clusterVal[e] = clusterVal[j];
         end
         j++;
      end
   else
      e = 1;
   endif
   sendRequest (clusterLeader[e]);
   receiveClusterKey(clusterLeader[e]);
   ClusterMembership();
end function
```

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

4.3 Cluster Creation

In the Cluster Creation stage (Algorithm 4), the vehicle is not close to any leader of a cluster. It should check whether within their neighbors there are at least X nodes that do not belong to any cluster, plus a variable Y that indicates the number of vehicles that can either turn off, separate or not join the new cluster that is being created. If the number of neighbors without cluster is lower than the minimum threshold required for cluster creation, the vehicle waits a period $time1$

and starts again the Cluster Detection stage. Otherwise, if the number of neighbors is greater than the threshold $X+Y$, the vehicle begins a new Cluster Creation process. In order to do it, it multicasts a cluster 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 cluster leader sends to each node the secret key of the cluster encrypted with the public keys of each node. In this moment the new cluster is formed. Otherwise, the number Y of estimated vehicles is increased by adding the number of vehicles that did not accept the invitation.

Algorithm 4: ClusterCreation.

```
function ClusterCreation (...)
  if (numberOfNeighbors ≥ X + Y) then
     AcceptedNeighbors = 0;
     n = 1;
     l = 0;
     MulticastNeighbors (NeighborsList[]);
  for(n=1;n≤ numberOfNeighbors;n++) do
        ReceiveClusterElection(n);
        if (neighbor(n) accept) then
           acceptedNeighbors(l)=neighbor(n);
           l++;
           acceptedNeighbors++;
        end
     end
     if (acceptedNeighbors ≥ X) then
   for(n=1;n≤ acceptedNeighbors;n++) do
     SendClusterKey(acceptedNeighbors(n),
        PuKacceptedNeighbors(n));
        end
        ClusterLife();
     else
        Y = Y + X - acceptedNeighbors;
        Wait(time1);
        ClusterDetection();
     end
  else
     Wait(time1);
     ClusterDetection();
  end
end function
```

In conclusion, this stage requires: a multicast of invitation to join the new cluster, unicast responses from n users and a multicast to relay a message that enables the members to build the cluster secret key. This means a total of $2n+1$ packets in case of positive cluster creation, and $n+2$ if the process fails. The Cluster 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.

4.4 Cluster Membership

Once the cluster is formed, the leader must periodically validate that the cluster continues being useful. Otherwise, it would be necessary to change the leader or to end the cluster.

Algorithm 5 shows the process where a node leaves its cluster. When the node loses any contact with the leader of the cluster for certain time, the node stops to belong to its cluster and begins the Cluster Detection stage if node density exceeds the corresponding threshold.

Algorithm 5: ClusterMembership.

```
function ClusterMembership (...)
  if (See( clusterLeader )) then
     Wait(timeMemberCheck);
     ClusterMembership();
  else
     Wait(timeReCheck);
     if (See( clusterLeader )) then
        Wait(timeMemberCheck);
        ClusterMembership();
     else
        finalClusterMembership();
        ClusterDetection();
     end
  end
end function
```


4.5 Cluster Life

Algorithm 6 shows how the leader of a cluster periodically checks that the cluster is still useful. If cluster 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 cluster in order to avoid introducing cluster 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 cluster ending process. First, the leader asks about the neighborhood density in order to know if neighborhood density (number of neighbors of the same cluster or without any cluster 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 Cluster Creation stage with those nodes without any cluster that are in its transmission range. In the absence of any neighbor exceeding the threshold, the leader sends the cluster ending signal through multicast to all its neighbors.

Algorithm 6: ClusterLife.

```
function ClusterLife (...)
  for(n=1;n≤ numberOfNeighbors;n++)do
     if(Belongs(neighbor(n),cluster(a)))then
        clusterSize++;
     end
  end
  if (clusterSize ≥ X) then
     Wait(time2);
     ClusterLife();
  else
     if (numberOfNeighbors ≤ D) then
        newLeader=0;
  for(n=1;n≤ numberOfNeighbors;n++)do
           //clusterSize+withoutCluster
           pot = potential(neighbor(n));
           if((pot≥ X)and
           (pot≥ clusterSize))then
              clusterSize=clusterSize(n);
              newLeader=n;
           end
        end
        if (newLeader == 0) then
           Multicast (End-of-Cluster-Signal);
           ClusterDetection();
        else
           Multicast (Leader-Change-Signal);
  //New leader init ClusterCreation proccess
           ClusterDetection();
        end
     end
  end
end function
```

5 SIMULATION AND ANALYSIS

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 clusters 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.

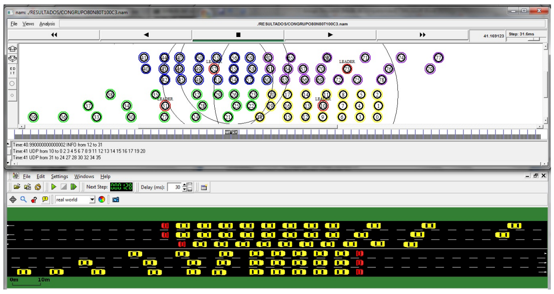


Figure 1: Simulation.

The implemented simulations for clusters consider four levels of development: vehicle mobility, node energy, cluster 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 cluster formation layer defines which vehicles belong to each cluster, who is the leader of each cluster, who generates traffic information and who relays information to other clusters.
- 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 clusters and the other without them, and they can be compared with the same topology (see Figure 2).

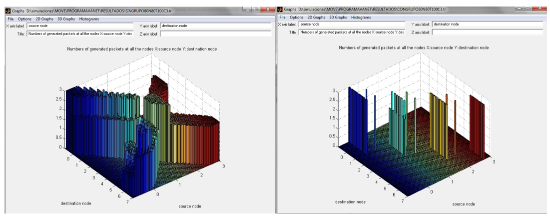
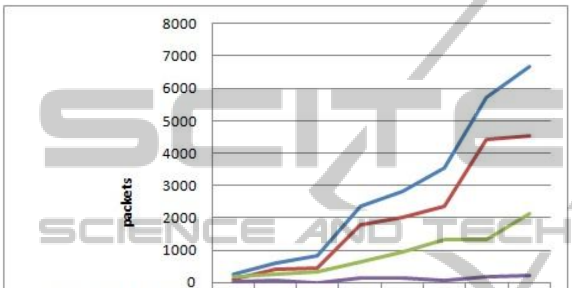


Figure 2: Generated packets without and with Clusters.

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 clusters, which nodes are the leaders of the clusters, 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. Table 3 shown some result of simulations. We have chosen the same set of common parameters previously mentioned.



vehicles with OBU	10	15	20	31	40	50	60	80
Without Groups Generated	278	598	825	2343	2805	3539	5732	6675
Without Group Lost	107	402	443	1804	2014	2350	4415	4529
With Groups Generated	167	277	351	638	932	1327	1314	2120
With Groups Lost	12	63	0	139	135	68	199	215

Figure 3: Generated and lost packets.

We can observe in Figure 3 the comparison between the average generated and lost packets: it is clear that, without the use of clusters in VANETs, the number of generated packets grow up much faster than with the use of clusters. 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. Clusters will help to decrease the percentage of lost packets and to perform the VANETs operation.

6 CONCLUSIONS

In this paper the use of clusters 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 cluster management in VANETs is provided, which includes differentiation among possible vehicle states: from the initial state when it does not belong to any cluster, to the choice of an existent cluster to join it, the creation of a new cluster, and the end of a cluster. This paper also shows how to proceed with cluster 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 clusters and without using them in VANETs.

ACKNOWLEDGEMENTS

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

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

Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. & Hernández-Goya, C., 2009 'Flexible Authentication in Vehicular Ad hoc Networks', *APCC IEEE*.

Dousse, O., Baccelli, F. & Thiran, P., 2003 'Impact of Interferences on Connectivity in Ad Hoc Networks'. *INFOCOM*.

Fan, P., Sistla, P. & Nelson,P. C., 2008 'Theoretical analysis of a directional stability-based clustering algorithm for vanets'. *Vehicular Ad Hoc Networks*.

Gunter, Y., Wiegel, B. & Gromann, H. P., 2007 'Medium Access Concept for VANETs Based on Clustering'. *VTC Fall:2189-2193*.

Molina-Gil, J., Caballero-Gil, P. & Caballero-Gil, C., 2010 'A Vision of Cooperation Tools for VANETs', *WoWMoM*.

Rawashdeh, Z. Y. & Masud Mahmud, S., 2008 'Media Access Technique for Cluster-Based Vehicular Ad Hoc Networks'. *VTC Fall*.

Raya, M., Papadimitratos, P. & Hubaux, J.P., 2006 'Securing Vehicular Communications - Assumptions, Requirements and Principles', *ESCAR*.