Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 384869, 12 pages
http://dx.doi.org/10.1155/2015/384869

Research Article

Self-Organized Clustering Architecture for Vehicular Ad Hoc Networks

Cándido Caballero-Gil, Pino Caballero-Gil, and Jezabel Molina-Gil

Department of Computer Engineering, University of La Laguna, 38204 Tenerife, Spain

Correspondence should be addressed to Cándido Caballero-Gil; ccabgil@ull.es

Received 5 February 2015; Revised 8 June 2015; Accepted 9 June 2015

Academic Editor: Chuan H. Foh

Vehicular Ad Hoc NETworks (VANETs) are likely to be deployed in the near future. Then, they will become the platform for much of the relayed information in the Internet of Things. This paper proposes a new system based on 1-hop clustering to reduce the number of VANET communications in dense road traffic scenarios while maintaining the security of communications by combining public-key with secret-key cryptography. The proposed distributed clustering architecture creates a dynamic virtual backbone in the network, formed by Cluster-Heads and cluster-gateways, so that these nodes are responsible for the efficient message propagation in the network. The main aim of the described architecture is to balance both stability of backbone connections and cost/efficiency trade-off. Full definitions of all the architecture procedures are provided, including a cluster-head selection algorithm based on a version of the independent set problem and a secret-key agreement scheme that uses the generalized Diffie-Hellman protocol. Simulations show that the proposal improves network performance and security.

1. Introduction

Vehicular Ad Hoc NETworks (VANETs) are spontaneous Peer-to-Peer (P2P) wireless networks formed by moving vehicles. VANETs can be seen as special Mobile Ad Hoc NETworks (MANETs) whose objective is to provide communication among On-Board Units (OBUs) in vehicles and between OBUs and Roadside Units (RSUs). They constitute promising technology for many distributed automotive applications such as augmented reality for driving assistance by offering different services regarding road traffic like dissemination of safety-related messages. VANETs integrate components of multiple ad hoc networking technologies, such as WiFi and WiFi Direct (IEEE 802.11b/g/p), WiMAX (IEEE 802.16), Bluetooth and Bluetooth Low Energy, 2G, 3G, and LTE, to achieve effective wireless communication. These networks are a fundamental part of the Intelligent Transportation System (ITS), which is a priority objective of many governments around the world.

VANETs, as a part of the Internet of Things (IoT), present unique challenges such as high mobility, real-time constraints, scalability, gradual deployment, well-defined scenarios, need of self-management, and privacy. Those challenges result in several problems like interrupting connections, difficult routing, security of communications, changing scenarios, scalability, and need to find a solution to decrease the number and size of packets exchanged among vehicles. In particular, the protection of communication characteristics such as authenticity, privacy, anonymity, cooperation, low delay, stability of communications, and scalability is harder in VANETs than in general MANETs due to their specific features.

As can be seen in Figure 1, vehicular environments for VANETs can be classified into three groups: urban area, rural area, and highways. In the three cases, VANET communications can be used to help prevent accidents, avoid traffic jams, or exchange information about traffic, vehicles, and roads conditions, so they can have many positive effects such as saving time and money and reducing environmental pollution and consumption of fuel reserves. Although the most relevant use of VANETs is for safety-related applications, other uses exist such as notification services, cooperative

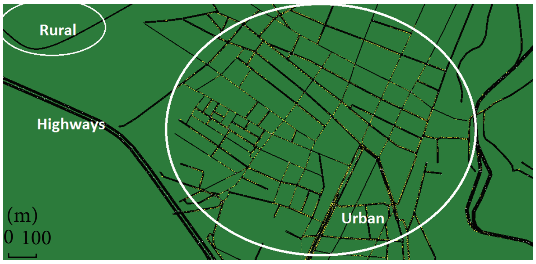


Figure 1: SUMO screenshot of scenarios for VANETs: highways and urban and rural environment.

driver assistance schemes, monitoring systems, and all types of value-added applications, including providing information on the nearest points of interest, interactive communication, and electronic payments. An important feature of these comfort-related and commercial applications is that they should not interfere with safety-related applications, which is reflected in the performing order described in the DSR (Dedicated Short Range) included in the IEEE 802.11p standard WAVE (Wireless Access for Vehicular Environments).

Communication between vehicles is called Vehicle-to-Vehicle (V2V), and communication between OBUs and RSUs is called Vehicle-to-Roadside (V2R). There are several advantages of V2V-based VANETs compared with V2R-based VANETs. First, V2V communications do not depend on the existence of RSUs, which is particularly attractive for developing countries or rural areas where roadside infrastructures can not be available. Second, V2V communications have fewer problems such as fast fading, short connectivity time, and high frequent hand-offs, caused by the high relative-speed difference between fast-moving vehicles and stationary RSUs. Finally, V2V communications fit better with VANET applications that require only exchange of information among vehicles within a geographic area. Equipping vehicles with sensors and OBUs will allow deploying most VANET applications, thanks to V2V communications. Thus, this paper focuses only on V2V communications.

This work proposes using clusters in VANETs to optimize communications in situations with high road traffic density by establishing a virtual backbone of the network. The scheme also allows using secret-key cryptography for more efficient intracluster information exchanges. Besides, clustering provides other benefits, such as network scalability and keeping communication bandwidth.

The proposed architecture implies that one node from each cluster acts as Cluster-Head (CH), so that communication within the cluster is organized by the CH. Furthermore, since vehicles may produce highly redundant information, in order to avoid this drawback known as broadcast storm problem, identical packets produced by different sources inside a cluster may be aggregated.

This paper is organized as follows. Related work on clustering in VANETs is summarized in Section 2. Section 3 gives basic definitions and notation used in the proposed protocols, which are fully described in Section 4. Section 5 explains how communication is conducted within the cluster. Sections 6 and 7 show simulation results. Finally, some conclusions close the paper.

2. Related Work

Many bibliographic references propose the use of clusters in ad hoc networks with different aims such as data dissemination and aggregation, group signatures, overhead minimization, and routing. With respect to ad hoc networks with mobile nodes, many authors have studied the general case of clustering in MANETs [1–5], and the particular case of clustering in VANETs has also been the object of different works [6–8]. Regarding practical applications, a few papers have shown that clustering in VANETs effectively reduces data congestion [9, 10] while supporting the requirements of Quality of Service (QoS) [11].

With respect to cluster formation in VANETs, which is one of the main topics of this work, different algorithms can be found in the bibliography. Many of those papers can be classified depending on the use of static or dynamic clusters. On the one hand, [12] proposed static clusters in urban settings to reduce the effect of signal attenuation due to physical obstacles, and the authors of [13] also proposed static clusters to reduce the number of broadcast packets. On the other hand, for instance, the work [14] proposed dynamic clustering based on grouping vehicles on the fly, depending on the relative-speed difference among neighboring vehicles. Also in [15], vehicles are dynamically clustered according to several metrics. The present work uses dynamic clusters too but based on different metrics.

Another classification of clustering schemes is based on the use of the mobility characteristic of vehicles. For instance, the work [16] presented a mobility-based clustering scheme for VANET, which deals with routing to produce clusters with high stability. As an example of non-mobility-based schemes, the work [17] proposed an autonomous clustering scheme based on network topology changes without considering the mobility of vehicles as one of the parameters for cluster formation. Here, a different mobility-based scheme is presented.

A frequent issue in the bibliography on clustering is CH selection. In the aforementioned work [13], the CH is simply the first vehicle of the cluster. Two other simple techniques for CH selection described in the works [18, 19] are based on the lowest ID and on the use of beacons, respectively. The first one does not optimize any characteristic of the network whilst the second one uses regular beacon transmissions to advertise node states. This latter approach tries to stabilize the CH state, especially in the case of large clusters because a CH only changes its state if it receives a message from a CH of another larger cluster. However, this simple criterion does not take into account factors such as whether the clusters are moving in opposite directions. The work [20] presented the so-called CASAN algorithm for CH selection, which takes into account the trust level of nodes, so it requires controlling reputation of nodes. In [21], the CH is selected according to mobility information and driver intentions. Another typical criterion for CH selection in ad hoc networks is the location of the CH candidate relative to the other nodes [22], which looks positive in VANETs because vehicles tend to travel in groups. However, such a criterion produces certain overhead because it requires continuous location awareness. Differently, in [9], the CH is proposed to be

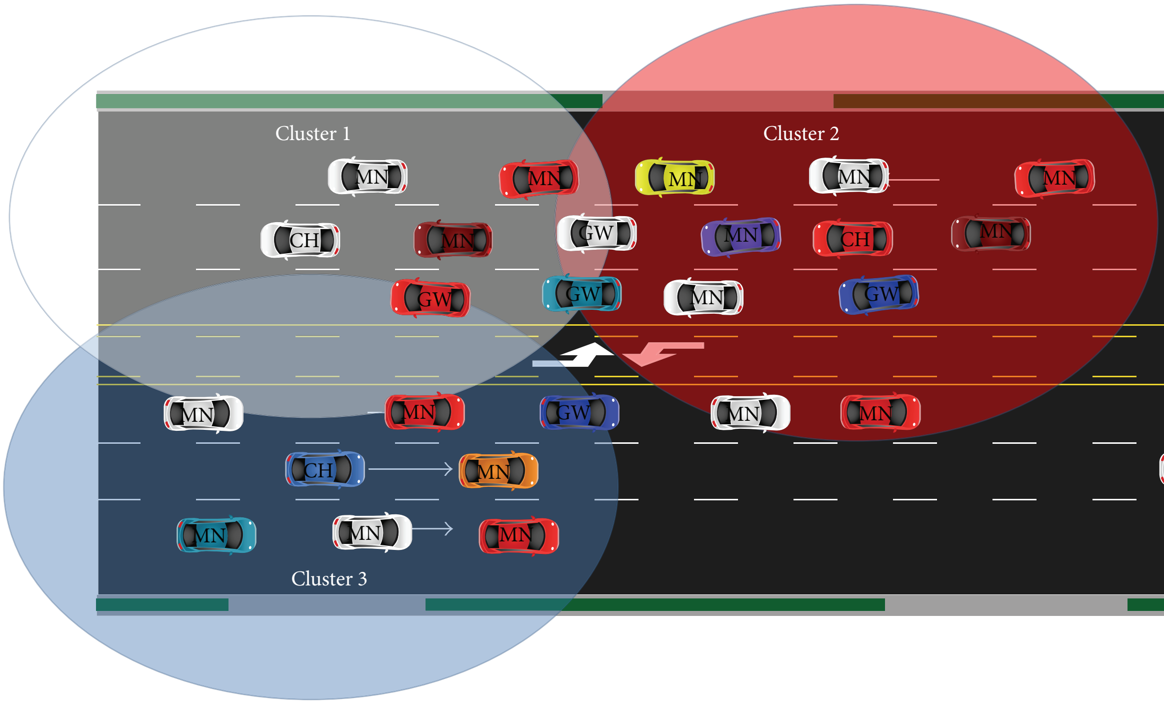


Figure 2: Example of node states in a 1-hop cluster scheme.

the vehicle with velocity closest to the average velocity of all reachable nodes, which makes sense because neighbor vehicles tend to travel with the same velocity so they tend to remain neighbors. However, such a criterion also produces overhead of communications. The work [23] proposed a mixed approach to choose the CH based on a utility function that uses as parameters both the location closest to the average and the velocity closest to the average. This proposal fails to adapt to the traffic dynamics because, for example, the cluster formation interval is fixed. In our proposal, the decisive factors for CH selection can be seen as a combination of several mentioned criteria together with other parameters.

Finally, an example of work where privacy-preserving is among its main goals is the work [24] that described the scheme called Caravan, in which vehicles can receive broadcasts from every other member of the cluster, whose aim is to allow vehicles to prevent tracking of their broadcast communications. Privacy-preserving is also a goal in the present paper.

The aforementioned cluster-based proposals motivate the need for a full study of all the processes that nodes have to complete for cluster management. It is also interesting to show implementations of practical schemes in order to study the performance of the proposals and demonstrate their reliability. These are exactly the two main objectives of the present work, which focuses on minimizing overhead in data dissemination and providing a way to create a shared secret key to use symmetric encryption within the cluster.

3. Basic Definitions and Notation

In this paper, we propose a collection of distributed protocols that allow building a VANET backbone formed by a virtual chain of vehicles to make the fast propagation of broadcast messages possible. The backbone formation and management are performed by exploiting some specific characteristics of VANETs, like the persistence of clustering in common scenarios.

Clusters are here defined as conceptual structures according to which groups of nearby vehicles traveling in the same direction self-organize around their selected representative called Cluster-Head. This special node assumes the role of manager for intracluster communications among the members of its cluster, which must be in a close communication range.

The role of gateways for intercluster communication is delegated to other members, depending on their proximity to other clusters. This is shown in Figure 2 where four basic states of nodes are identified: Cluster-Head (CH), Member Node (MN), GateWay (GW), and Non-Defined (ND).

Clusters are especially useful under heavy traffic conditions, when density of road traffic in a specific geographic zone is high, such as in traffic jams, because in these cases the number of V2V communications is much higher. Under these circumstances, the highly dynamic topology of VANETs can disturb cluster formation and reorganization, producing an increase in cluster instability. Therefore, clustering algorithms must be designed to maintain cluster structure as stable as possible in order to protect the performance of communication. The goal of the algorithms proposed in this paper is to efficiently manage VANET clusters where each CH is directly connected to all nodes in its cluster. In particular, clusters are defined according to dynamic cells where the CH is chosen by distinct criteria, such as being the node with the largest number of neighbors in its cluster. In particular, the decision rule for the CH selection takes into account different

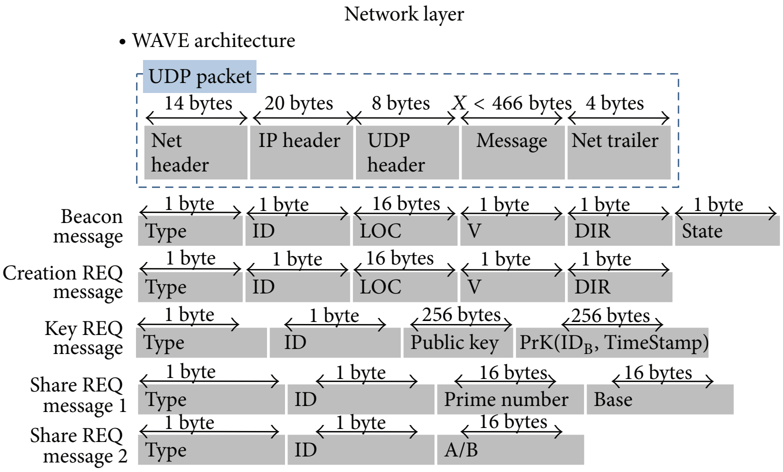


Figure 3: Packet format.

factors, such as the average speed, the central location, and the direction of vehicles. Also the state of neighbors is taken into account because no two CHs are allowed to be neighbors during cluster formation, and each node must have at least one CH in its neighborhood. Thus, the above definition of clusters implies that vehicles circulating in the same direction and at the average speed of the cluster have a low probability of changing cluster within its route. Besides, the proposed cluster scheme is especially useful to manage data aggregation for traffic events [25].

Cluster management must satisfy two important requirements. First, it should minimize resource consumption and message exchange. Second, it must take into account the highly dynamic topology of the network. Our proposal implies a significant reduction in the number of retransmissions through broadcast. In particular, if n denotes the number of nodes in the vicinity of a vehicle, without any cluster the vehicle has to send approximately n packets for each received broadcast. In the best case, if the broadcast's origins are registered in the packet, each neighbor that receives such packets is expected to broadcast again the same information to all neighbor nodes that are not registered in the packet, so the total number of communications among the $n + 1$ nodes in the neighborhood is $n!$. However, when using the proposed cluster-based scheme, only n connections are generated per cluster of $n + 1$ nodes for each data retransmission. The first connection is between the member node that first receives or produces the broadcast information and its CH, and then the next $n-1$ connections are between the CH and the remaining $n - 1$ members of its cluster, including possible gateways responsible for sending the information to neighbor clusters.

In our proposal, nodes are assumed to be periodically broadcasting beacon messages containing the following:

$$\langle ID, loc, v, dir, state \rangle, \tag{1}$$

where ID is a variable pseudonym used by the sender in order to enable the other nodes to link messages sent by it, but with protecting its anonymity, loc denotes the GPS coordinates of the sender's location, v is the speed of the sender, dir is the direction of the sender in degrees that can be parsed to (n, s, e, w, ne, nw, se, sw), and *state* indicates if the sender is CH, MN, GW, or ND.

The basic notation used throughout the algorithms of the proposed architecture is described in Notations at the end of the paper.

Independently, the VANET is running under WAVE architecture (IEEE 802.11p) or under IEEE 802.11 a/g, like in a VANET built on smartphones, and messages are encapsulated in UDP packets in the network layer. In order to try to reduce losses of packets, their size is kept small. In particular, the format of some of the UDP packets involved in the protocols is shown in Figure 3.

The GPS coordinates *loc* of neighbors will allow checking at least partially neighbor information that is sent during the cluster creation phase explained in the following section. The speed v of neighbors is used not only to decide who will be the CH but also to exclude those vehicles whose speeds are outliers with respect to the remaining velocities of their neighbors. The parameter *dir* is used here to identify the nodes that are candidates to form part of a cluster because all nodes in a cluster must travel in the same direction. These data are also useful to determine the destination of messages because, for example, some of them have to be propagated only in one direction, while others, such as warnings about congestion due to an accident, must be propagated in both forward and backward directions. Finally, with respect to the parameter *state*, since in our scheme all nodes have to belong to some cluster, at least formed by itself, the state ND can only be used for the initial state of the node before executing the protocols described in the following section. Furthermore, when a node belongs to more than one cluster, it becomes a GW for the intercluster communication, forming part of the backbone for message propagation in the VANET.

4. Architecture Description

This section contains the description of the procedures that form part of the proposed clustering system architecture,

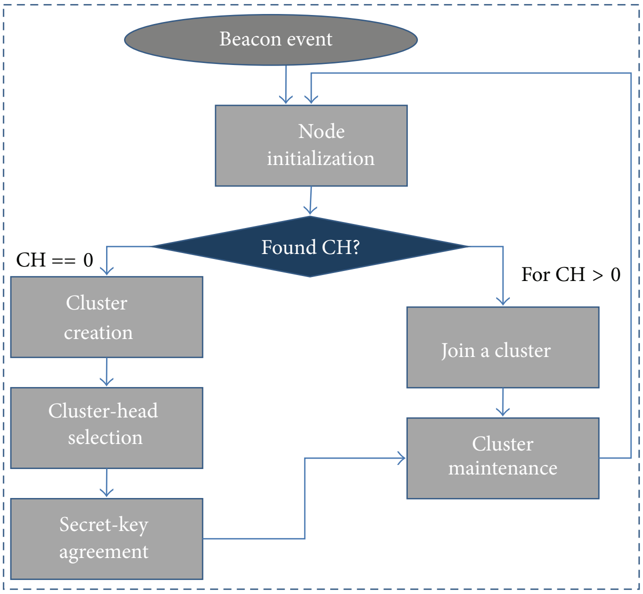


Figure 4: Six-stage lifecycle of nodes.

```
function NodeInitialization (···)
  $i = 1$;
  card(CH(NG[$x$])) = 0;
  foreach (NG[$x$][$i$]) do
    if (NG[$x$][$i$] is CH) then
      card(CH(NG[$x$]))++;
    end if
  end for
  if (card(CH(NG[$x$])) == 0) then
    ClusterCreation( );
  else
    for ($j = 1$; $j \leq$ card(CH(NG[$x$])); $j$++) do
      JoinaCluster( );
    end for
  end if
end function
```

Algorithm 1: Node initialization.

```
function ClusterCreation (···)
  $l = 1$;
  Broadcast(CreationREQ($x$));
  foreach (NG[$x$][$i$]) do
    if (Rec(Answer,$i$,$x$)) then
      CL[$x$][$l$] = NG[$x$][$i$];
      $l$++;
    end if
  end for
  if ($l \geq 1$) then
    CHSelection(CL[$x$]);
  else
    $state$ = CH;
  end if
end function
```

Algorithm 2: Cluster creation.

including all the details of every possible stage in cluster management, depending on the specific situation of vehicles. Thus, in brief, the clustering scheme proposed in this paper is formed by the six stages (see Figure 4) described in the following subsections. When a node is not a member of any cluster, it launches the initialization phase where it implements a cluster discovering process. Afterwards, the node can execute either the join procedure or the cluster creation phase, depending on whether it found a CH nearby or not. During cluster creation, both the CH selection and the cluster secret-key agreement are carried out. After the whole cluster creation phase and also after joining a cluster, the member node proceeds to the cluster maintenance procedure where it periodically checks the validity of the cluster.

4.1. Initialization. This stage is launched by any vehicle that receives a beacon from another vehicle and its state is ND because it does not belong to any cluster yet. It is described in Algorithm 1, which the node executes to discover whether a CH is nearby or not.

Every vehicle in ND state checks periodically whether a CH exists among its neighbors in the range of its speed or not. If there is at least one CH neighbor, the node proceeds to the join procedure. Otherwise, it proceeds to the cluster creation. This stage does not generate any additional traffic of control because all the necessary information to run it is contained in the beacon messages that nodes periodically broadcast.

4.2. Cluster Creation. The cluster creation stage is defined in Algorithm 2, which is launched every time a node in ND state has previously run the initialization phase and found that there is no CH nearby. In order to begin a new cluster creation process, the executor node broadcasts a request towards all neighbors traveling in the same direction and with speed inside the speed range of the neighborhood.

Each node that receives this request has to respond accepting the invitation and indicating the number of its neighbors that are candidates to become members of a new cluster with itself as CH. After this, the nodes that answered to the invitation will launch the CH selection stage, and the shared secret key will be established according to the cluster secret-key agreement protocol. After that, the new cluster can be considered completely established. In conclusion, this stage basically requires a broadcast of invitation to join the new cluster and unicast responses from the n receiving candidate nodes, which means a total of $2n$ packets. Consequently, management packets generated at this stage do not decrease the communication throughput.

4.3. Cluster-Head Selection. In this section, an algorithm to select a node as CH is proposed. The main idea of the CH selection algorithm is to allow a node to evaluate its potential as CH before playing this role and to step down if it is not the best candidate for being CH at that moment. When a node decides to self-nominate as a CH, it broadcasts an invitation message to recruit its neighbors. After getting the invitation from this CH candidate, the neighbors join the new cluster. Each CH periodically checks the ability of its cluster members for being a better CH than itself, and if one of these neighbors is a better candidate for CH, it steps down and proposes such a node to become the new CH. This renovation process is also executed automatically if the CH leaves its cluster.

Multiple criteria are used for CH selection. On the one hand, the CH has the least probability (when compared to others within the same cluster) to move out of the cluster because its speed is close to the average speed of all members of the cluster. This ensures that a node highly mobile with respect to its neighbors will not be elected as CH. At the same time, the efficiency of intracluster communications is maximized with the proposed election of the CH because the CH has the minimum distance from the geographic cluster center.

The problem of arranging nodes into clusters is treated as the problem of finding a maximal weighted independent set of nodes. We introduce a distributed algorithm for the determination of a maximal weighted independent set in the graph that represents the VANET, which only requires that each node has certain knowledge of its neighborhood.

An independent set in a graph $G = (V, E)$ is a set $I \in V$ such that there is no pair of nodes in I linked by an edge in E. The maximal independent set problem consists in finding an independent set that is not properly contained in any other independent set. This problem admits a natural generalization called maximal weighted independent set problem, defined for graphs in which each node is associated with a weight. In this case, the objective is to find a maximal independent set by choosing the nodes of the set that maximize the total weight. Unlike the maximum weighted independent set problem, which is NP-hard [26], a polynomial solution can be found for the maximal independent set problem. In this paper, we deal with this latter problem for the specific class of graphs representing the topology of VANETs.

According to our cluster definition, no two CHs can be neighbors. Furthermore, the network has to be covered with a backbone of CHs, which implies that each node must have at least one CH in its neighborhood. Consequently, the clustering problem can be reduced to the problem of finding a minimal dominating set, which is closely related to the maximal independent set problem formed by the CHs in the network, because any maximal independent set is a minimal dominating set. In particular, in our solution, we associate one weight to each node to indicate how suitable it is for the role of CH according to parameters such as number of neighbors, location, and speed. Therefore, the algorithm for selecting the CHs is equivalent to the problem of finding a maximal weighted independent set in the graph of the network, and the nodes in the independent set can self-nominate to be the CHs. In order to run distributed Algorithm 3, each cluster member only has to know the weights of its neighbors. Initially, only nodes with higher weights with respect to their neighborhood broadcast a message to their candidate neighbors stating that they will be the CH. In a second round, if a node does not receive any of these messages, it broadcasts one of them. Otherwise, it checks whether its role is MN or GW.

```
function CHSelection $(\cdots)$
  CHNom = 1;
  for ($i = 1$; $i \le$ card(CL[$x$]); $i$++) do
    if ($w(i) > w(x)$) then
      CHNom = 0;
    end if
    if ((CHNom == 1)
    ||(card(Rec(CHNom,CL[$x$],$x$)) == 0)) then
     state = CH;
     Broadcast(CHNom);
     SecretKeyAgreement(CL[$x$]);
    else if (card(Rec(CHNom,CL[$x$],$x$)) == 1) then
       state = MN;
     else
       state = GW;
     end if
    end for
end function
```

Algorithm 3: Cluster-head selection.

4.4. Cluster Secret-Key Agreement. Most references about secret communications in VANETs suggest the use of public-key cryptography based on a Public-Key Infrastructure (PKI) with certificates issued by a Certificate Authority (CA). This solution implies that a public/private key pair is assigned to each node and stored in its tamperproof device, and public-key certificates are authenticated either by a centralized or a distributed CA.

Our proposed management scheme allows combining PKIs with the use of secret-key cryptography. It assumes that each message sent in a VANET contains a digital signature that can be used to identify the sender node, but this increases communication and computation overhead. In order to reduce this overhead, the establishment and use of secret keys shared in clusters are proposed because secret-key cryptography is in general more efficient than public-key cryptography. The large size of VANETs prevents vehicles from preloading shared keys, so secret-key establishment must be dynamic. Communication in promiscuous mode with shared secret key and proximity of cluster members make it possible for nodes of the cluster to control that the CH and other nodes in the cluster act properly and send correct messages.

In order to preserve equal roles for all OBUs, we take advantage of the distributed nature of the proposed clusters to define a key agreement process as general recommended approach for key establishment. Other methods can be used, but our proposal implies that the CH broadcasts certain information to all members, which allows them to compute independently the same shared secret key.

The proposed agreement protocol establishes a secret key for all members of a cluster, based on each node's contribution exchanged openly over an insecure wireless medium. The secret key obtained with Algorithm 4 can be used to establish a secure channel between all cluster members.

In particular, in the scheme described in Algorithm 4, nodes forming a new cluster generate a shared secret key

```
(1) function SecretKeyAgreement $(\cdots)$
(2)   Broadcast(ShareREQ($x$));
(3)   for ($i = 1$; $i \leq$ card(CL[$x$]); $i$++) do
(4)     Rec($g^{S_i}(\mathrm{mod}\, p)$,$i$,$x$);
(5)   end for
(6)   Broadcast($\{h(g^{S_i}, g^{S_i S_x})\}$ $\forall i \neq x$);
(7)   $K_x = g^{S_x(1+\sum_{i \neq x} S_i)}$;
(8) end function
```

Algorithm 4: Secret-key agreement.

```
function JoinaCluster $(\cdots)$
  for ($j = 1$; $j \leq$ card(CH(NG[$x$]$x$)); $j$++) do
    KeyREQ($j$);
    Rec($K_j$, $j$, $x$);
    ClusterMaintenance();
  end for
end function
```

Algorithm 5: Join a cluster.

through a scheme based on the difficulty of the discrete logarithm problem, which consists in computing the value of S, given $g^S(mod\ p)$, g, and p. This problem is the basis of the well-known Diffie-Hellman method to exchange a shared secret between two parties. Consequently, this work proposes the use of the generalization of Diffie-Hellman key agreement protocol to more than two users.

The algorithm is based on a bit-commitment scheme so that each node i commits to its contribution to the shared secret key. In this way, the CH, denoted in Algorithm 4 as the executor node x, can neither change this contribution nor read it. The use of a commitment scheme makes the exchange of public information for enabling the generation by each node of the shared secret without putting the shared secret key or the different contributions at risk possible.

The broadcast in step 6 of Algorithm 4 poses no threat to the secret of the cluster key, as it is useless for any node that has not contributed to the secret. It is also important to remark that although Algorithm 4 is launched by the CH, every cluster member i can check if its contribution was correctly included in the message sent by the CH. In such a case, it can compute independently the cluster secret key with the message received from the CH, by removing its share from $g^{S_i S_{\mathrm{CH}}}$ to get $g^{S_{\mathrm{CH}}}$ and then computing the secret key according to the following expression:

$$K_{\mathrm{CH}} = g^{S_{\mathrm{CH}}} \cdot \prod_{i \neq \mathrm{CH}} g^{S_i S_{\mathrm{CH}}} = g^{S_{\mathrm{CH}}(1+\sum_{i \neq \mathrm{CH}} S_i)}. \tag{2}$$

According to the aforementioned algorithm, the cluster key is generated with the contributions of the first members of the cluster. While the cluster exists, those nodes that join it receive the secret key encrypted with the public key of the new node from the CH.

The proposed use of clusters reduces overhead but does not allow defining different security levels among members. Instead, it mainly protects the network from potential outsider attackers. Hence, the delivering of the existing secret keys shared within a cluster is required when a new member joins, but if a member leaves, it does not involve any update of cluster key.

Secret-key encryption is in general more efficient than public-key encryption, so thanks to the shared secret-key establishment process proposed above, any secret-key encryption can be used in VANETs to get confidentiality by secret-key encryption. Besides safety-related applications, other scenarios exist where confidential communications may be necessary. This is the case, for example, of certain commercial applications.

On the other hand, the main security challenge for multicast is to have an effective method for controlling access to the multicast messages. However, if the receiver nodes share a secret key, a basic method to limit access to multicast messages is through encryption. Therefore, after the agreement on the cluster secret has been carried out, multicast in clusters is possible.

4.5. Join Procedure. This stage starts when a vehicle finds among its neighbors at least one node that is CH. Algorithm 5 shows the stage according to which a member node joins all the clusters corresponding to CHs in its neighborhood.

In order to proceed with this stage, the node first has to send a login request encrypted with its public key to every CH neighbor. After authenticating it, the CH sends its corresponding cluster secret key encrypted with such a public key and, in this way, the vehicle becomes part of the cluster and proceeds to the cluster maintenance phase.

4.6. Cluster Maintenance. Mobility in VANETs is usually highly dynamic. Nearby vehicles usually drive close to each other for several kilometers while other vehicles bypass them quickly. This is the main reason why the continuous execution of the cluster maintenance phase is necessary, where the validity of clusters is checked.

Algorithm 6 defines the process that each MN or GW node has to carry out while it belongs to the cluster. The node verifies that it has not lost contact with its CH every T time units. It considers that it has lost contact with its CH if it has not received any message/beacon from its CH after two periods of time T. In that case, it changes its state to ND and begins the initialization stage.

5. Message Management

Thanks to the use of clusters, the number of sent messages decreases remarkably without missing useful information. Algorithm 7 defines the steps that a node has to execute after receiving or generating a message.

If the executor node is the final destination of the message, it simply processes the information. Otherwise, its reaction depends on whether the node is CH or not. In this latter case, it sends it through encrypted unicast to the CH. If

```
function ClusterMaintenance (⋯)
    while (Rec(Message,CH,$x$)) do
        Wait($T$);
    end while
    Wait($T$);
    if (Rec(Message,CH,$x$)) then
        ClusterMaintenance( );
    else
        $state$ = ND;
        NodeInitialization( );
    end if
end function
```

Algorithm 6: Cluster maintenance.

```
function MessageManagement (⋯)
    if (Dest(Message) == $x$) then
        Process(Message);
    else
        if ($state$ == CH) then
            if (card(Dest(Message)) == 1) then
                if (Dest(Message) in CL[$x$]) then
                    Unicast(Message,$x$,Dest(Message));
                else
                    Multicast(Message,$x$,GWCL[CH]);
                end if
            else
                Multicast(Message,CL[CH]);
            end if
        end if
        else
            Unicast(Message,$x$,CH);
        end if
    end if
end function
```

Algorithm 7: Message management.

the executor node is CH, its action depends on whether the message is to be propagated or not. In the first case, it sends it through encrypted multicast to all its cluster members. Otherwise, if the message has a single destination and this destination belongs to its cluster, it unicasts the message to the destination through encrypted intracluster communication. Otherwise, if the single destination does not belong to its cluster, it sends through encrypted unicast the message to all GWs, which forward it to other CHs through encrypted intercluster communication. The existence of GWs is of high importance because they provide some degree of overlap between clusters, which enables intercluster communication not only for spreading of messages but also for other applications such as topology discovery and node location.

6. Simulation

Both the feasibility and effectiveness of our approach are shown through a simulation that exemplifies its performance. In Figure 5, the outputs of NS-2 and SUMO show the VANET state when clusters are operating. At the top of that image, we can see an NS-2 simulation with connections between nodes, while, at the bottom, the related SUMO simulation for traffic mobility is shown.

In the simulation, we compared three models: a VANET implemented without clusters, the proposed 1-hop cluster-based scheme, and the Caravan approach [24], where every node is in the range of transmission of all nodes in the cluster. We have selected Caravan because although the objective of Caravan is not the same as in this work, it is interesting to compare the proposed architecture with our proposal. Static clusters or clusters with more than one hop like those presented in [27] have not been taken into account because the differences in the number of communications are obvious. On the one hand, with static clusters every node changes its cluster many times. On the other hand, by using more than one hop, the number of control packets is much higher than that by using only one because the CH has to know all the two-hop distance nodes.

The most relevant choices for the 1-hop clustering simulation were

(i) total number of vehicles: 80,

(ii) number of vehicles with OBUs: 10–80,

(iii) number of lanes for each direction: 3,

(iv) simulation time: 100 seconds,

(v) retransmission period: 10 seconds,

(vi) distance relay nodes: 100 meters,

(vii) number of simulations: 1000,

(viii) departure speed: 15 m/s,

(ix) packet type: UDP,

(x) packet size: 512 bytes.

The simulated VANET is formed with vehicles connected by WiFi 802.11b/g, which is estimated to reach between 50 and 300 meters depending on different factors such as climate conditions, obstacles, interference in the channel, and speed. The simulation does not consider Doppler effect because in dense traffic conditions this factor is minimized due to the low speed of vehicles. The high mobility of VANETs causes a high probability of packet loss. For this reason, we have fixed the packet size to 512 bytes and traffic sources to UDP. Considering that vehicles are under dense traffic conditions, the speed is low, ranging from 18 m/s to 0 m/s.

Four different layers were defined.

(i) The vehicle mobility layer manages the node movement in the simulation pattern, which defines roads, lines, different speed limits for each line, traffic jams, and so forth.

(ii) The node energy layer is used to distinguish between vehicles with and without OBUs. Vehicles without OBUs are on the road but do not contribute to communications.

Table 1: Results of a simple simulation.

Number of nodes	No clusters		1-hop		Caravan	
	Sent packets	Lost packets	Sent packets	Lost packets	Sent packets	Lost packets
10	278	107	167	12	187	20
15	598	402	277	43	271	48
20	825	443	351	61	427	52
30	2343	1804	638	79	749	88
40	2805	2014	932	95	1009	100
50	5077	3981	1101	132	1190	137
60	5732	4415	1314	159	1693	168
80	6675	4529	2120	215	2357	232

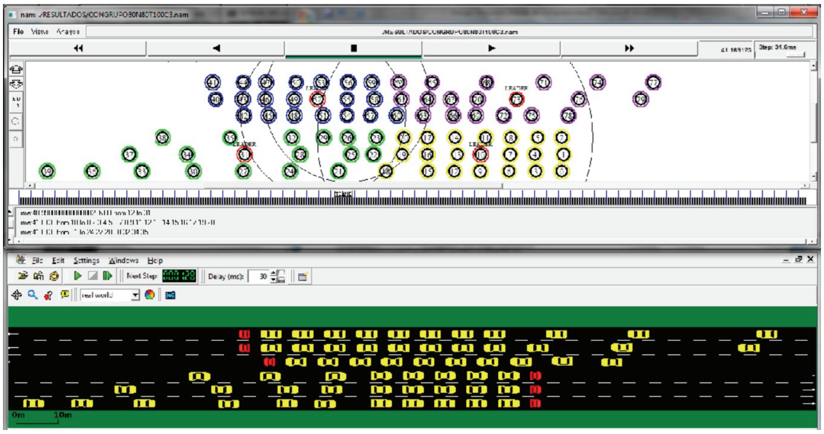


Figure 5: Simulator snapshot for a traffic jam in a highway with 3 lanes on each side.

(iii) The cluster formation layer defines which vehicles belong to each cluster and their roles, that is to say, which one is the CH of each cluster, which ones generate traffic information, and which ones are the GWs and relay information to other clusters.

(iv) The P2P communications layer is responsible for the definition of which nodes are in the transmission range of the retransmitting node at any time.

Simulations give essential statistics such as numbers of generated and lost packets. These basic statistics data are useful to make efficient simulations for large-scale scenarios.

7. Experimental Analysis

The implemented simulations with the cluster-based proposal and with Caravan are first compared with results obtained from the simulation without clusters with the same topology. This helps to illustrate the validity of the cluster-based VANET proposals. Some results of the scheme are shown in Table 1.

Figure 6 shows the obtained topology and results of the simulations. At the top of such a figure, the topology of the network is shown, while, at the bottom, it is the topology of generated packets that is shown. Among the obtained information from the simulations, we have the number of packets and bytes that are generated, sent, broadcast, received, lost, and so forth for each node. Also, other displayed information is the number of generated and lost packets, the number of formed clusters, which nodes are the CHs, which nodes generate packets, which nodes forward them, and so forth. In addition to all this information, another interesting aspect is that the implementation provides a detailed simulation of what is happening 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.

In Figure 7, we can see a comparison between the average numbers of packets that are generated and lost. It is clear that, without the use of clusters in VANETs, the number of generated packets grows up much faster than that with the use of clusters, and so does the number of lost packets. The main reason for such a higher number of generated packets without clusters is the heavier traffic load that VANETs generate in traffic jam conditions when clusters are not used. It is also clear that clusters help to decrease the percentage of lost packets, and so the use of clusters improves VANET performance. From the comparison between our proposal and Caravan, it can be deduced that both numbers of generated and lost packets are lower in our proposal. Furthermore, the retransmission time in Caravan proposal is higher than that in our proposal because the formed clusters are smaller and the number of nodes through which the information passes is greater.

Figure 8 shows the average size of the clusters for different densities both in the approach of this paper and in Caravan

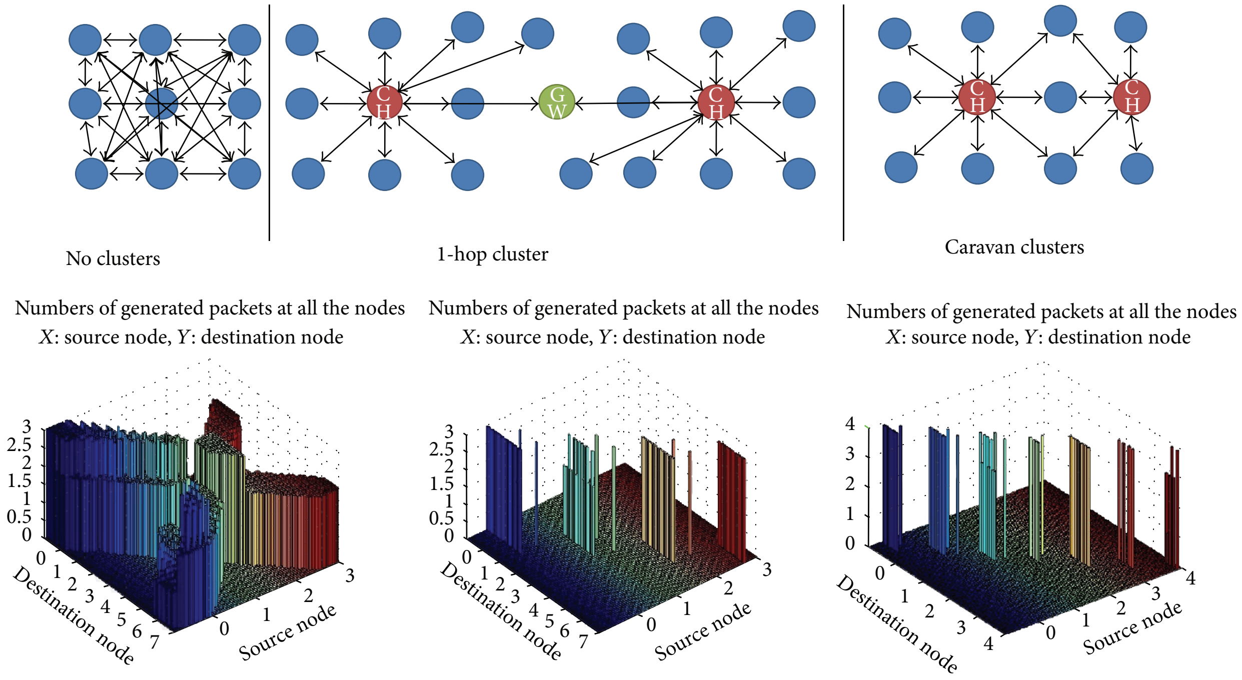


Figure 6: Results obtained from the simulations.

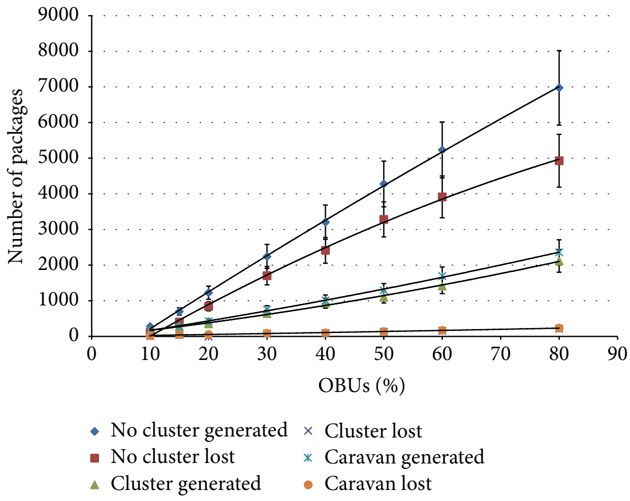


Figure 7: Generated and lost packets: without clusters versus 1-hop clusters versus Caravan.

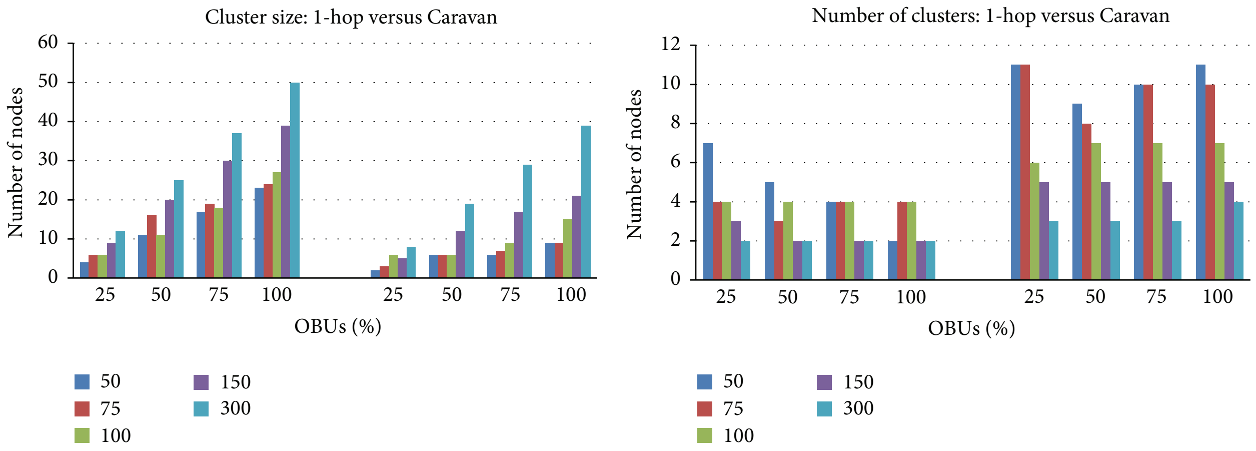
approach. The largest clusters are those with the highest density of vehicles. The proposal was simulated in a realistic traffic jam environment, with a three-lane highway in each direction and 100 vehicles. It can be seen that the number of vehicles inside the cluster is increased in a linear way. The second part of Figure 8 shows the average size of the clusters using Caravan approach.

In Figure 6, we can see that the number of nodes belonging to each cluster is smaller than that in the first approach, and they are about half the nodes of our approach. Figure 8 also shows the number of clusters formed using the same parameters as in Figure 7. In this case, we can see that an increase of the range of transmission reduces the number of clusters. This is an expected result because the size of clusters is larger. In Caravan approach, the same happens, but the number of clusters is higher than that in our approach because all nodes must have connection with all the nodes of the cluster, and the clusters in Caravan approach have a smaller size.

8. Conclusion

The use of 1-hop clusters has been proposed as a solution to decrease the amount of ad hoc transmission in a secure VANET under dense road traffic conditions, when the overhead of sent data leads to a significant drop in communication quality. In particular, a thorough description of the proposed scheme for autonomous cluster management has been provided, including differentiation among possible states of every vehicle, from the initial state when it does not belong to any cluster to the choice of an existing cluster or the creation of a new cluster. This paper also shows how to proceed with inter- and intracluster communications. Besides, both a cluster-head selection algorithm based on a version of the independent set problem and a secret-key agreement scheme based on a generalization of Diffie-Hellman protocol are presented.

Using the open source traffic simulator SUMO and the network simulator NS-2, we have done a detailed analysis of the proposal based on software simulations. The obtained data have been used for a broad comparison of communication overhead produced with our cluster-based proposal, with the network without any clusters, and with the Caravan approach. The conclusion is that our proposal improves



(a)

(b)

Figure 8: Size and number of clusters depending on the retransmission distance.

VANET performance, while guaranteeing real-time message delivery.

Notations

x:	Executor node
NG$[x]$:	Set of neighbors of x
$n = \text{card}(\text{NG}[x])$:	Cardinality of NG$[x]$
CH(NG$[x]$):	Subset of NG$[x]$ formed by CHs
NG$[x][i]$:	ith neighbor of the executor node x
isCH(i):	Boolean function showing whether i is CH or not
CreationREQ(x):	Cluster creation request sent by x
Rec(M, i, x):	Message M from node i received by x
CL$[x]$:	Members of the cluster whose CH is x
CHNom:	Self-nomination to become CH
$w(i)$:	Weight value associated with node i, indicating how suitable it is for the CH role according to parameters such as its number of neighbors, location, and speed
ShareREQ(x):	Key share request sent by x
Dest(M):	Destination of message M
p:	Prime number
g:	Generator element of Z_p
S_i:	Integer in $[0, p - 2]$ randomly chosen by i
g^{S_i}:	Public commitment of node i to integer S_i
h:	A cryptographic hash function
K_x:	Secret key of the cluster with CH $= x$
KeyREQ(CH):	Key request sent to CH
Wait(T):	Waiting for a time T before the next step.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research is supported by the Spanish MINECO and the European FEDER Fund under Projects TIN2011-25452, IPT-2012-0585-370000, RTC-2014-1648-8, and TEC2014-54110-R.

References

[1] P. Basu, N. Khan, and T. D. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proceedings of the 21st International Conference on Distributed Computing Systems Workshop*, pp. 413–418, 2001.

[2] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile Ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[3] I. I. Er and W. K. G. Seah, "Mobility-based d-hop clustering algorithm for mobile ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 2359–2364, March 2004.

[4] J. Sucec and I. Marsic, "Hierarchical routing overhead in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 46–56, 2004.

[5] Y. Yi, M. Gerla, and T. J. Kwon, "Efficient flooding in ad hoc networks: a comparative performance study," in *Proceedings of the International Conference on Communications (ICC '03)*, pp. 1059–1063, May 2003.

[6] S. M. AlMheiri and H. S. AlQamzi, "MANETs and VANETs clustering algorithms: a survey," in *Proceedings of the IEEE 8th GCC Conference and Exhibition (GCCCE '15)*, pp. 1–6, Muscat, Oman, Feburary 2015.

[7] L. Bononi and M. Di Felice, "A cross layered MAC and clustering scheme for efficient broadcast in VANETs," in *Proceedings of*

the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS '07), pp. 1–8, October 2007.

[8] C. Shea, B. Hassanabadi, and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–6, December 2009.

[9] W. Chen and S. Cai, "Ad hoc peer-to-peer network architecture for vehicle safety communications," *IEEE Communications Magazine*, vol. 43, no. 4, pp. 100–107, 2005.

[10] M. A. Jan, P. Nanda, X. He, and R. P. Liu, "PASCCC: priority-based application-specific congestion control clustering protocol," *Computer Networks*, vol. 74, pp. 92–102, 2014.

[11] R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multihop mobile wireless networks for quality-of-service support," *Mobile Networks and Applications*, vol. 3, no. 1, pp. 101–119, 1998.

[12] A. Mahajan, N. Potnis, K. Gopalan, and A. Wang, "Modeling VANET deployment in urban settings," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07)*, pp. 151–158, October 2007.

[13] N. Maslekar, M. Boussedjra, J. Mouzna, and L. Houda, "Direction based clustering algorithm for data dissemination in vehicular networks," in *Proceedings of the IEEE Vehicular Networking Conference (VNC '09)*, pp. 1–6, October 2009.

[14] M. S. Kakkasageri and S. S. Manvi, "Multiagent driven dynamic clustering of vehicles in VANETs," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1771–1780, 2012.

[15] A. Benslimane, T. Taleb, and R. Sivaraj, "Dynamic clustering-based adaptive mobile gateway management in integrated VANET-3G heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 559–570, 2011.

[16] B. Hassanabadi, C. Shea, L. Zhang, and S. Valaee, "Clustering in vehicular Ad Hoc networks using affinity propagation," *Ad Hoc Networks*, vol. 13, pp. 535–548, 2014.

[17] K. Okano, T. Ohta, and Y. Kakuda, "A dynamic network gateway selection scheme based on autonomous clustering for heterogeneous mobile ad hoc network environment," in *Proceedings of the IEEE Globecom Workshops*, pp. 513–517, December 2012.

[18] Y. Günter, B. Wiegel, and H. P. Grossmann, "Cluster-based medium access scheme for VANETs," in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC '07)*, pp. 343–348, October 2007.

[19] R. A. Santos, R. M. Edwards, and N. L. Seed, "Inter vehicular data exchange between fast moving road trafic using an ad-hoc cluster-based location routing algorithm and 802.11b direct sequence spread spectrum radio," in *Proceedings of the Post-Graduate Networking Conference*, 2003.

[20] M. E. Elhdhili, L. Ben Azzouz, and F. Kamoun, "CASAN: clustering algorithm for security in ad hoc networks," *Computer Communications*, vol. 31, no. 13, pp. 2972–2980, 2008.

[21] J. Blum, A. Eskandarian, and L. Hoffman, "Mobility management in IVC networks," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 150–155, 2003.

[22] W. Choi, P. Shah, and S. K. Das, "A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks," in *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '04)*, pp. 203–212, August 2004.

[23] P. Fan, J. G. Haran, J. Dillenburg, and P. C. Nelson, "Cluster-based framework in vehicular ad-hoc networks," in *Ad-Hoc, Mobile, and Wireless Networks*, vol. 3738 of *Lecture Notes in Computer Science*, pp. 32–42, Springer, Berlin, Germany, 2005.

[24] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "CARAVAN: providing location privacy for VANET," in *Proceedings of the 3rd Embedded Security in Cars Workshop*, 2005.

[25] J. Molina-Gil, P. Caballero-Gil, and C. Caballero-Gil, "Aggregation and probabilistic verification for data authentication in VANETs," *Information Sciences*, vol. 262, no. 20, pp. 172–189, 2014.

[26] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[27] E. Dror, C. Avin, and Z. Lotker, "Fast randomized algorithm for 2-hops clustering in vehicular ad-hoc networks," *Ad Hoc Networks*, vol. 11, no. 7, pp. 2002–2015, 2013.