Efficient Dissemination Based on Passive Approach and Dynamic Clustering for VANET

Abdelali TOUIL[1*], Fattehallah GHADI[1], Asma SBAI[2]
[1*,1,2]Sciences Engineering Laboratory Faculty of science IBN ZOHR University
Morocco
abdelali.touil@edu.uiz.ac.ma
f.ghadi@uiz.ac.ma
a.sbai@uiz.ac.ma

ABSTRACT: *The evolution of the Intelligent Transport System (ITS) has imposed a major impact on road safety by providing a wide variety of applications to decrease the number of accidents and traffic management. As an emerging component of ITS, Vehicular Ad-hoc Network (VANET) is considered as a promising technology for increasing road safety. Basically, VANET provides two types of communications: vehicle-to-vehicle (V2V) and vehicle-infrastructure (V2I) communications. While the vehicles act in a cooperative way to enhance road traffic safety from adopting the V2V communication, also known as Inter-vehicle communication (IVC), the biggest challenge is how to address the overhead and stability issues that are caused by the important number of messages generated by vehicles at critical areas, such as intersections. In this paper, we focus on the implementation of distributed system based on a passive data dissemination approach and dynamic clustering. This consist on each vehicle sending periodical measures about the position, velocity to other vehicles belonging to the same signal range and to the same cluster. Which implied an earlier divi-sion of the network into virtual subgroups to ease management and data dissemination of messages. Afterward, we simulate the solution using microsimulation taking in consideration the behavior of each vehicle.*

Keywords: VANETs , Ad-hoc, V2V, Dynamic Clustering, Collision Avoid-ance

Received: 25 May 2017, Revised 22 June 2017, Accepted 2 July 2017

© 2017 DLINE. All Rights Reserved

1. Introduction

Quite recently, road traffic accidents are classified the ninth cause of death in the world. Besides, they are estimated to be in the seventh place by 2030. As reported by [1], the number of road traffic deaths by injuries has been fixed to 1.25 million per year. Furthermore, over than 50 million people have non-fatal wounds as a consequence of road accidents. While there are additional indirect consequences that are associated to other fields, e.g. environment, economy, etc. In a perspective to reduce this disaster, several researches have been carried out, especially in the field of Intelligent Transportation Systems (ITS) [2]. ITS describe technology applied to the interaction of vehicles with each other and with the infrastructure, to ameliorate driving experience, as well as, to enhance the safety and ability of road traffic.

As a subclass of Intelligent Transportation Systems, Vehicular Ad-hoc Networks (VANETs) are advanced applications which are conceived to provide deep insights into roads traffic and new solution to various issues [3]. In VANETs, vehicles can be connected to each other for many aims such as sending a safety messages to warn vicinity cars in case of a crash or even sending information about the road traffic to authorities, as well as connecting vehicles to other resources, e.g., the Internet. Generally, the performances of VANETs depend on many features, compared to other Ad-hoc networks such as Mobile Ad-hoc Network (high mobility, variation of topology, unbounded network size, unlimited energy and storage resources) [4]. The challenges toward applying VANETs - as efficient collision avoidance system - mostly include network stability and overhead. According to the number of vehicles in urban areas or in congested space, vehicles exchange a large amount of messages, which provoke a high control overhead of network. Due to the proximity of vehicles [5], each vehicle requires disseminating the control messages within the network to alert other vehicles about its position and velocity to avoid crashes. Therefore, minimizing sent messages is an important step toward efficient collision avoidance. For these reasons, in this paper, we will implement a protocol based on clustering approach to facilitate man-agement and data dissemination of messages.

The remainder of the paper is organized into five sections: After the introduction, Section 2 provides application challenges; Section 3 presents the related work; Section 4 describes the dissemination road traffic protocol. Performance evaluation is presented in Section 5; Section 6 concludes the paper.

2. Overview of Context-aware

The collision warning system, also known as collision mitigation system, was created to decrease the number of collisions between vehicles using sensors, image recognition or cooperative collision warning. When an accident occurs, sensors control the braking system. The major disadvantage of those sensors is the restriction on detection distance[6], which left only a few milliseconds before a collision. In addition, image recognition may not be efficient in certain weather conditions (heavy rain, snow, thick fog ...) which can cause lack of visibility. The data provided by the cam-eras installed on the car, in this case, becomes untrusted and no longer reliable for collisions detection.

However, collision cooperative warnings (CCW) are based on received messages from wireless communications sent by nearby vehicles. They help drivers to avoid accidents or to diminish the consequences by detecting other vehicles in large scale or in longer distances and informing them of a potential danger of collision. The concept of CCW can be classified into three approaches[7], namely: passive approach, active approach, and hybrid approach.

In a passive approach, each vehicle sends a periodically high precision measurement of its location and velocity to neighbor vehicles, so each neighbor must update information and calculate the inter-vehicle distance to determine the possible collision and warn the driver. In active approach, if a vehicle reacts in an abnormal behavior, such as hard braking or a breakdown, it sends automatically a warning message to the neighbors, which includes the cause of breaking, location and velocity. Each recipient of this message can then make the decision, for example: braking and notify the driver. However, the hybrid approach combines both approaches together to support dif-ferent applications. So, to ensure the correct achievement of these three approaches, it is necessary that each vehicle sends accurate measurements of location and velocity.

3. Related Work

In the passive approach, messages are sent proactively using periodic broadcast. This technique is usually used in safety application, such as collision warning system and road condition. Contrary to delay-tolerant applications, the passive approach requires strict latency constraints.

In [8], authors developed a novel clustering algorithm, that classifies vehicles according to congested traffic flow, the algorithm was adapted to use trajectory abstraction as a metric, which groups vehicles that travel in the same location into the same cluster. The results obtained from the solution provides highly precise and real-time traffic data that reduce traffic congestion in an urban environment, by locating a vehicle at the head of a congested traffic flow, drivers can receive efficient traffic data. Moreover, to compensate the error caused by GPS equipment during the selection of the vehicle in the head of congested traffic, the algorithm employs an abstracted location and trajectory representation.

MOBIC [9], is an algorithm based on clustering method to provide stability of Mobile Ad-hoc Network (MANET). It uses mobility metric for cluster formation which aims to enhance the dissemination of packets. Basically, the mobility is calculated on the ratio between the received powers levels of successive transmissions measured at any node from all its nearby nodes. If relative mobility of two nodes is negative, it designates that they move away. On the other hand, if relative mobility is positive, it means that the nodes are moving closer. To select cluster-head (CH), each node calculates the variance of relative mobility for its neighbors, then, the node with a low variance respect to its neighbors becomes CH and start cluster formation process.

However, our paper is devoted to the passive approach. To meet the needs of large scale and minimizing the number of exchanged messages, we will implement a dynamic clustering approach (Figure 1), which is essentially a method to divide a group of vehicles into virtual sub-groups called clusters. Every cluster will be managed by vehicle called cluster head (CH). Each CH is selected according to specific criteria, which are generally the key to form stable clusters. Clustering improves the stability and quality of service (QoS) of network [11]. So, we envisage ensuring more stability and QoS issues to allow an interactive communication process between vehicles for more efficient cooperative collision warning using passive approach.

4. System Model

The originality of the proposed solution is based on merging dynamic clustering and passive approach, which can be used to provide services related to safety issues to drivers. Thanks to dynamic clustering, vehicles proceed together to form clusters. This method is an effective way to segregate a network into different homogeneous clusters which vehicles shared common features.

From Figure 1, we can distinguish three types of vehicles when dealing with dynamic clustering, namely:

• **Cluster-Head:** Local coordinator of a cluster.

• **Cluster-Member:** Ordinary vehicle.

• **Cluster-Gateway:** Vehicle with inter-cluster connections, forwards information between clusters.

The event-based nature of the cooperative collision warning requires reliable intervehicular communication during all times. If

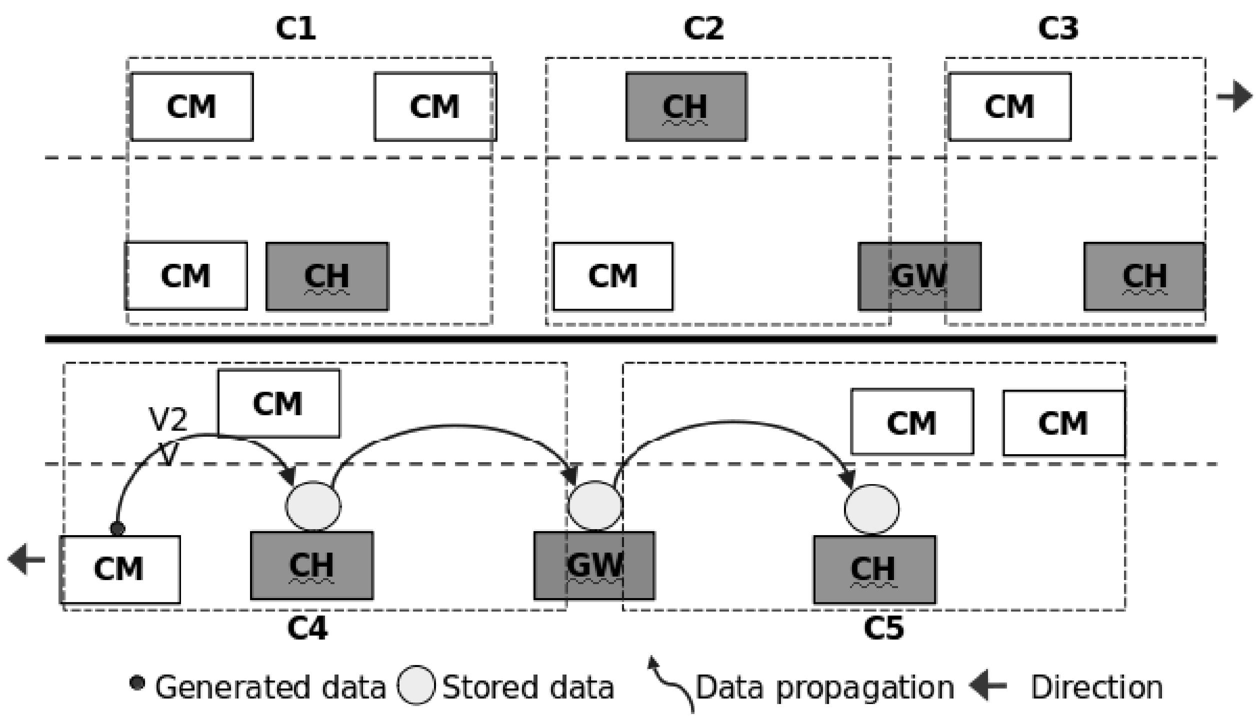

Figure 1. Clustering architecture

the position and velocity of a distant vehicle are not received by the neighbors in real time, the application fails and the traffic returns to the initial situation of crashes. To constitute more stable clusters, each vehicle is a candidate to join a cluster, only if it fulfills the conditions imposed by the Cluster-Head. An example of those rules: a cluster is composed of cluster members moving in the same direction, and their velocity must not exceed the velocity of cluster-head. For these reasons, the vehicles will not process the messages sent by all the vehicles of the network but only the messages sent by the members of its clusters. Also, each cluster member benefits from a time allocation to communicate its position and velocity. Thus, the cluster-head is gathering an important amount of information which decreases the interchanged messages between cluster members. The information from adjacent clusters is received by gateways and is sent only to the vehicles that are close to these clusters.

4.1 Neighbors Table Construction and Update

Basically, each vehicle starts its neighbor discovery process by generating and broadcasting a Beacon message (Figure 2) to its neighbor vehicle that in turn forward this message to their neighbors and so on, until all vehicles are reached. Consequently, each receiver generates a retention period -Time to live (*TTL*) - for each source that is saved in the neighbor table. We note that source is discarded from the neighbor table of the receiver if the prescribed *TTL* elapses.

Address of vehicle	**Location**	**Velocity**
ID of cluster	**Address of CH**	**State**
Weight		

Figure 2. Beacon message format

Initially, the fields: ID of the cluster, the address of *CH* and weight are empty because the vehicle has not yet decided its role. Each receiver collects the incoming messages and uses them to construct or update the neighbor table φ_k.

Every vehicle k will maintain a neighbor table φ_k, which has a neighbor list entry $n_{\tilde{k}}$ for every neighbor $\tilde{k}$. Each neighbor list entry $n_{\tilde{k}}$ contains the following fields $p_{\tilde{k}}$, $v_{\tilde{k}}$, $w_{\tilde{k}}$ and $TTL_{\tilde{k}}$.

$$\varphi_k = \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_i \end{pmatrix}, \text{ where } n_k = \begin{pmatrix} p_k \\ v_k \\ w_k \\ TLL_k \end{pmatrix} \quad (1)$$

where φ_k is the neighbor table, $n_{\tilde{k}}$ is the neighbor vehicle of $\tilde{k}$, $p_{\tilde{k}}$ is the position, $v_{\tilde{k}}$ is the average velocity, $w_{\tilde{k}}$ the weight and k $TTL_{\tilde{k}}$ is the retention period.

4.2 Cluster Head selection Criteria

Once the neighbors tables are created, vehicles participate in the selection process to choose the optimal vehicle within their vicinity as *CH*. Vehicles are clustered according to their position and velocity. To designate the *CHs*, each vehicle computes its weight. The vehicle with the higher weight in its vicinity becomes *CH* . The weight w_k is computed by summing the degree difference D_k and the average velocity v_{ave}. To assure balancing between clusters, the degree-difference provides the optimal number of *CMs* for each cluster. However, the velocity which is a prominent feature of VANET has a negative effect on topology and connections. To form stable clusters, we have added the average velocity to selection metrics.

The *CH* selection process consists of sex steps, executed by each vehicle, as described below:

Step 1: Compute the number of neighbors connected to given vehicle k.

$$\delta_k = |\varphi_k| = \sum_{\tilde{k}\in\varphi_k, \tilde{k}\neq k} \{\text{dist}(k, \tilde{k}) \leq R\} \quad (2)$$

where $dist(k, \tilde{k})$ is the distance between k and $\tilde{k}$, $\tilde{k}$ is neighbor connected to k and R is the signal range of k .

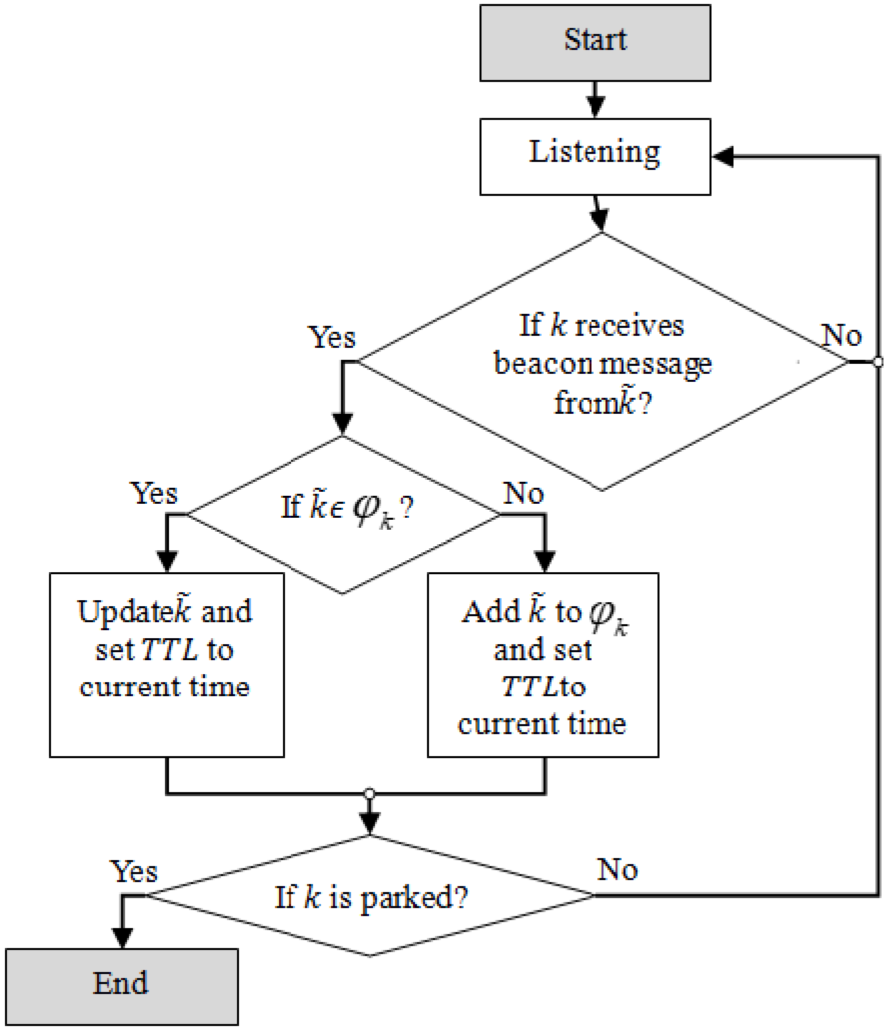

Figure 3. Construction and update of Neighbors Table process

Step 2: Compute the degree-difference D_k:

$$D_k = |\delta_k - c| \tag{3}$$

where c is the number of CMs supported by *CH*.

Step 3: Compute the average velocity v_{ave}, the formula can be written as:

$$v_{ave} = \frac{1}{\delta_k} \cdot \sum_{\tilde{k} \in \varphi_k}^{\delta_k} |v_{\tilde{k}} - v_k| \tag{4}$$

where v_{ave} is the average velocity of the current vehicle k and $v_{\tilde{k}}$ is the velocity of each neighbor.

Step 4: Considering the two weighting factors, the weight of the vehicle k can be calculated by the following formula:

$$w_k = \alpha . D_k + \beta . v_{ave} \tag{5}$$

where α, β are weighting factors, chosen in the simulation.

Step 5: Then, each vehicle broadcast the beacon message including the calculated weight.

Step 6: Each vehicle k compare the calculated weight with the received weights. k become a CH if the calculated weight is highest than the received weights. So, to invite the neighbors to join the cluster of k , k sent CH Message to all its neighbors.

If a vehicle $\tilde{k}$ receives this message and $\tilde{k}$ is CH , it changes its state to CM for Join k , if the weight of k is higher than the weight of $\tilde{k}$, $w_k \succ w_{\tilde{k}}$ and k move in the same direction of $\tilde{k}$. Otherwise, if k is a CM , it compares between the weight of its CH and the weight of k , if the weight of k is higher than the weight of the CH of $\tilde{k}$, $\tilde{k}$ joins k in its cluster.

4.3 New Connection

When a new vehicle $\tilde{k}$ is added to the neighbors table of k , k executes the follow-ing procedure. A vehicle k detects the presence of a new neighbor vehicle $\tilde{k}$. k sends Membership Message to $\tilde{k}$, if they travel in the same direction, the state of $\tilde{k}$ is CH and its weight is higher than the weight of k , $w_{\tilde{k}} \succ w_k$.

However, If the state of k is CH and its weight is higher than the weight of $\tilde{k}$, $w_k \succ w_{\tilde{k}}$. $\tilde{k}$ Checks the direction of k , if they travel in the same direction, k sends Membership Message to $\tilde{k}$.

We note that Membership Message is emitted when a vehicle changes its state to become CM if a vehicle joins the new cluster, it must broadcast this message to inform all the neighboring that it has changed its state. Basically, Membership Message contains the type of message sited to *JOIN*, the state of the sender and the new CH , as well as the identifier of the joined cluster.

Address of vehicle	**Location**	**Velocity**
ID of cluster	**Address of CH**	**State**
Weight		**Type**

Figure 4. Clustering message format

If a Cluster-Member vehicle $\tilde{k}$ receives Membership Message. It compares between the address of the sender and the address of its CH, if they are the same, $\tilde{k}$ verifies if there are *CHs* with weights higher than its weight and they move in the same direction, if there is one CH that fulfills those conditions, $\tilde{k}$ sent Membership Message to join this one. Otherwise, $\tilde{k}$ changes its state to the CH and sends a CH Message, where type changed to CH , to all neighbor vehicles.

4.4 Disconnection and Link Failure

When the value of *TTL* assign to a vehicle $\tilde{k}$ is exceeded, a vehicle k detects that a connection between it and $\tilde{k}$ is no longer available. k removes this one in its Neighbor table and Cluster table if is CH . Otherwise, if $\tilde{k}$ is CH and k is a Cluster Member in the same cluster, k checks if there is one a CH with a weight higher than its weight. If exist one CH , k sent Membership Message to this one. Otherwise, it becomes CH and sends a CH message to all neighbors vehicles.

5. Performance Evaluation

In this section, the simulation is conducted to verify the performance of the proposed vehicular cooperative collision warning dissemination solution with the network simulator OMNET++ [12]. To estimate the real-world road environment, we have integrated the Simulation of Urban MObility (SUMO) [13] and Vehicles In Network Simulation (VEINS) to OMNET++.

In the first comparison, two different simulations have been conducted: The first simulation is by way of the passive approach without clustering approach and the second simulation in the study is the passive approach integrated to the clustering approach.

Figure 5 shows that the number of lost packets when using clustering increases slowly with velocity. This explains clearly the significant increase of message success rate since most messages are sent to members of the same cluster, not to all network vehicles.

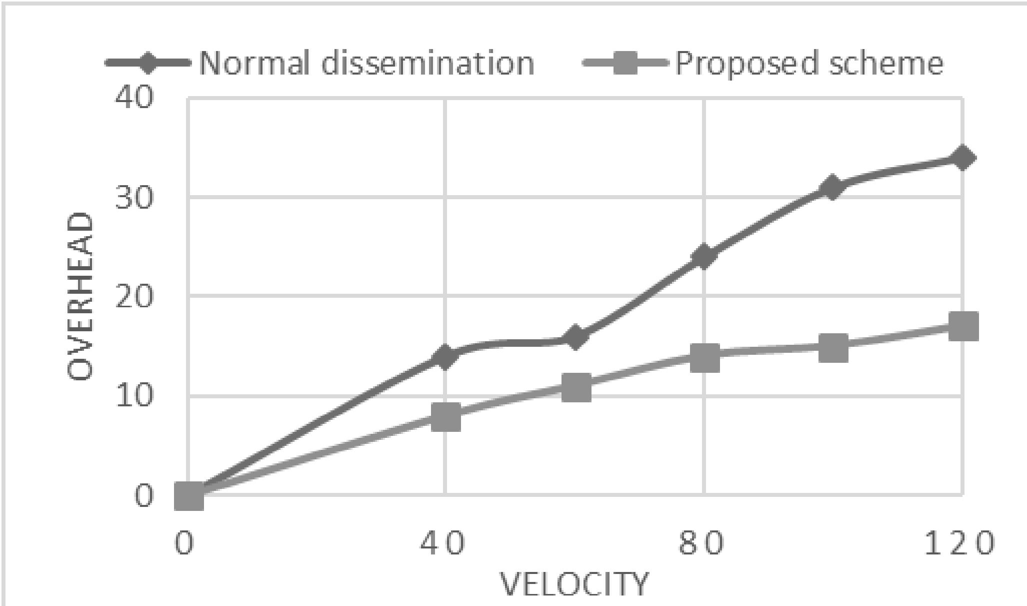

Figure 5. Overhead by velocity

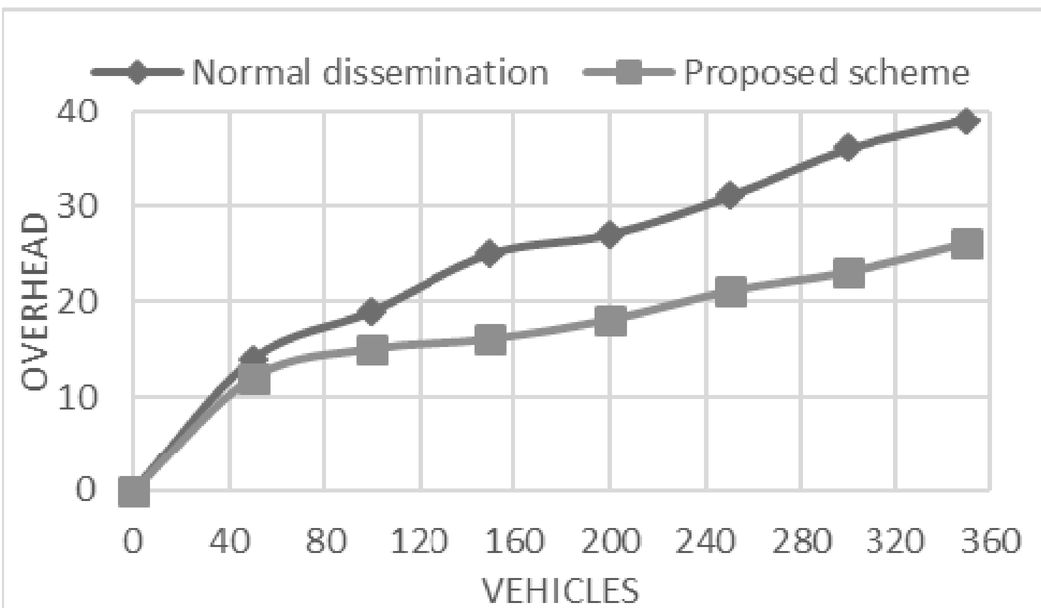

Figure 6. Overhead by number of vehicles

To evaluate the system performance in Figure 6, two important elements were identified: the overhead or the percentage of lost packets and the number of vehicles in the network. So, the proposed scheme was introduced to provide a reasonable comparison between the two elements. As can be seen in the figure the system is considered better in number of lasted packets than the normal dissemination of information.

In terms of clustering visualization, the graphical of the Figure 7 display the percentage of lost packets according to the velocity. We made a comparison with MOBIC algorithm. In this case, the proposed scheme gives more stability than MOBIC when vehicles move at high velocity.

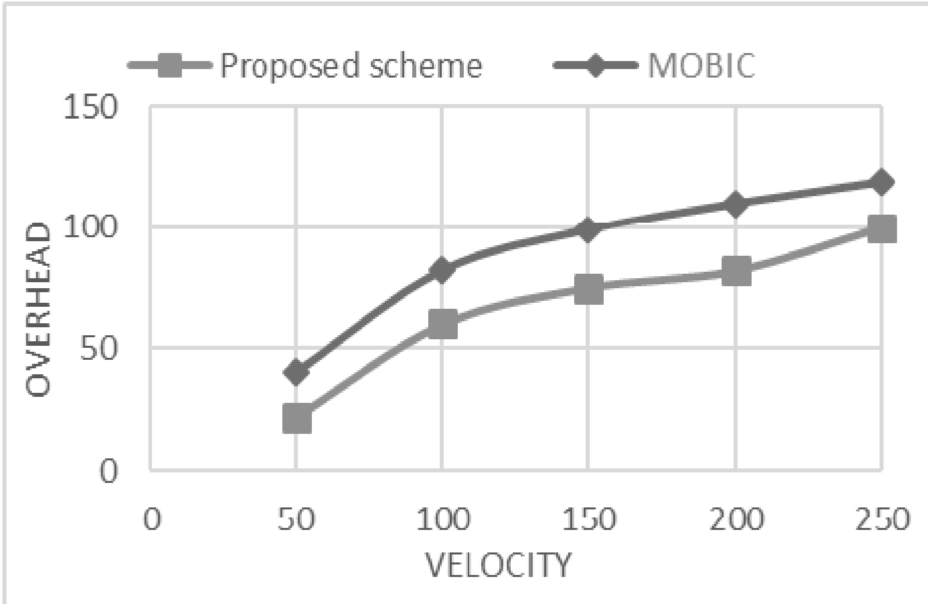

Figure 7. Average number of cluster by vehicles

6. Conclusion

The solution presented in this paper was optimized to better correspond to the road traffic needs. The objective was to increase the stability of a cluster structure, for more dissemination of road traffic information used to avoid a collision. The first simulation results show that the number of lost packets is significantly reduced in comparison. We contend that the fundamental use of connected vehicles and infrastructures will one day make Vehicular Ad-hoc Networks a reality. Over future decades, we foresee significant progress in other technologies, such as distributed control, artificial intelligence, vehicle sensors, and energy management. These advances will enable new challenges, such as accuracy of information, decision support systems in real time, etc.

References

[1] World Health Organization (2015). Global status report on road safety 2015: supporting a decade of action. WHO, Geneva, Switzerland

[2] Williams, B. (2008). Intelligent transport systems standards. Artech House, Boston.

[3] Sam, D., Velanganni, C., Evangelin, T. E. (2016). A vehicle control system using a time synchro-nized Hybrid VANET to reduce road accidents caused by human error. 6, 17–28.

[4] Touil, A., Ghadi, F. (2017). Implementation of Clustering Metrics in Vehicular Ad-Hoc Networks. *In*: Europe and MENA Cooperation Advances in Information and Communication Technologies. p. 441–449. Springer, Cham.

[5] Knipling, R. R., Mironer, M., Hendricks, D.L., Tijerina, L., Everson, J., Allen, J.C., Wil-son, C. (1993). ASSESSMENT OF IVHS COUNTERMEASURES FOR COLLISION AVOIDANCE: REAR-END CRASHES. FINAL REPORT.

[6] Knoeppel, C., Schanz, A., Michaelis, B. (2000). Robust vehicle detection at large distance using low-resolution cameras. *In*: Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511). p. 267–272.

[7] Chaqfeh, M., Lakas, A., Jawhar, I. (2014). A survey on data dissemination in vehicular ad hoc networks. *Veh. Commun*. 1, 214–225.

[8] Kanemaru, Y., Matsuura, S., Kakiuchi, M., Noguchi, S., Inomata, A., Fujikawa, K. (2013). Vehi-cle clustering algorithm for sharing information on traffic congestion. *In*: ITS Telecom-munications (ITST), 2013 13th International Conference on. p. 38–43. IEEE.

[9] Basu, P., Khan, N., Little, T.D.C. (2001). A mobility based metric for clustering in mobile ad hoc networks. *In*: Proceedings 21st International Conference on Distributed Computing Systems Workshops. *p*. 413–418.

[10] Yang, X., Liu, L., Vaidya, N.H., Zhao, F. (2004). A vehicle-to-vehicle communication protocol for cooperative collision warning. *In*: Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. *The First Annual International Conference on*. p. 114–123. IEEE.

[11] Touil, A., Ghadi, F. (2016). Distributed Clustering Scheme to Relieving Broadcast Storms in Vehicular Ad-Hoc Network. *In*: Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016). p. 551–557. Springer, Cham.

[12] Varga, A. (1999). Using the OMNeT++ discrete event simulation system in education. *IEEE Trans. Educ*. 42, 11–p.

[13] Liu, J. (2010). Association for Computing Machinery: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium.