

---

---

# A Weight Based Clustering Algorithm for Internet of Vehicles

Rim Gasmi<sup>a, \*</sup> and Makhlof Aliouat<sup>a, \*\*</sup>

<sup>a</sup>Faculty of Sciences, Ferhat Abbas University Setif 1, LRS D Setif, 19000 Algeria

\*e-mail: Gasmi34000@gmail.com

\*\*e-mail: Aliouat.m@yahoo.fr

Received December 12, 2019; revised March 11, 2020; accepted May 18, 2020

**Abstract**—Owing to the rapid growth in networking field in the recent few years, Internet of vehicles (IoV) has become one of the vast-growing networks, according to the high number of interacted connected nodes. The emergence of the new concept of Internet of Things (IoT) has given vehicles the ability to connect to everything anywhere and anytime. Even so, the increasing number of connected nodes such as vehicles, road sides, and smart phones causes several problems like network congestion that obstructs the quality of service of network. In case of an emergency situation, time is a critical factor to broadcasted messages on network, where the process has to be done as fast as possible to prevent disastrous consequences. Moreover, the high dynamism of vehicles drives routing process to be a very challenging task. Clustering algorithms are the commonly employed techniques to solve these problems. The key purpose of this paper is to propose an efficient mechanism to make IoV network more manageable and stable. In this paper, we propose a new weight-based clustering algorithm using safety, density and speed metrics. The proposed solution was verified and compared with the recent proposed works in this field (MADCCA and CAVDO) with the use of NS3, SUMO and MOVE simulation tools. Simulation results confirm the superiority of our algorithm by showing that our schema achieves better nodes connectivity and clusters stability than the other protocols.

**Keywords:** IoV, clustering, Routing Protocol, quality of service

**DOI:** 10.3103/S0146411620060036

## 1. INTRODUCTION

Over the last few years, IoV has appeared as a new class of efficient information dissemination technology among communities of users primarily because of their large uses in different fields such as Intelligent Transport Systems (ITS) [1]. A vehicle in IoV behaves as a smart machine that provides various services to the end users with or without the help of the existing infrastructures [2]. IoV environment is characterized by the high number of nodes, the high dynamicity and the random distribution of vehicles on the road [3]. Hence, routing messages to their final destination becomes a very challenging task. In the case of an emergency situation with people's lives at risk, time of alert sharing is a critical factor, where the messages have to be broadcasted as quickly as possible to prevent injuries and damages from happening. This can be guaranteed only if one ensures a reliable connectivity and fast distribution of messages. For this, clustering techniques have been widely used in various existing proposals to guarantee network connectivity, network security, low network overhead [4–6] and so on. By definition, clustering algorithms are defined as mechanisms of gathering vehicles and other intelligent machines based on some selected parameters like position, density, speed, direction of nodes, among others [7]. The majority of the proposed clustering algorithms focused on minimal network overhead or minimal number of cluster heads. We believe that all these metrics have less importance compared to the primary goal of providing stable connectivity between nodes, even at the cost of higher network overhead. In this paper we propose a new weight-based clustering algorithm for vehicles environment based on three parameters namely: node safety, node density and node mobility in order to guarantee reliable network connectivity and minimize network overhead. The rest of paper is organized as follows: The precedent proposed works are presented in section two. Section three presents our proposed work. The network architecture is presented in section four. Simulation results and performance evaluation are presented in section five and in the last section we conclude our proposal.

## 2. RELATED WORK

Several algorithms have been proposed to address clustering issue based on different selected metrics. The authors in [8] proposed a new passive multi-hop for clustering algorithm (PMC) in VANETs, where

they used a priority function based on three factors such as: vehicle node degree, the expected transmission count and link lifetime. The link lifetime parameter is calculated based on speed, direction and distance in order to group vehicles and form clusters. Authors assumed that the vehicle node with the highest degree, smallest expected transmission count and longest link life time is selected to be a cluster head with multi-hop strategy. The parameters used in their proposed work show to be important and prominent in improving the quality of service. All the parameters used in this algorithm indicate the quality of link in term of stability but no factor between them indicates the quality of link in term of security of transmission. Authors in [9] suggest to use Bus Node (BN) routes as a reference to collect the moving tracks of nodes such as buses and vehicles and then select suitable CHs. These tracks are used to calculate the contact time between a node and a BN based on time of entering to BN zone radius (meet time) and time of departure of BN. The contact time represents spatial and time dependence within the region of use in increasing cluster stability. Based on this reasoning, the authors proposed a new idea by using BN routes and assuming that cluster stability depends on the calculated contact time. Overall, the cluster head election process is based only on meet time and departure time and no other parameters are used. However, these two parameters are not enough to guarantee a stable clustering schema especially in vehicles network. Authors in [10] proposed a new clustering algorithm named MADCCA that considers two factors to select Cluster Heads, which are: density and mobility. The Cluster Heads (CHs) are selected by calculating the standard deviation of average relative velocity and density matrices in their neighborhood. As such, vehicles that have more homogeneous environments will be elected as Cluster Heads and the rest of vehicles in their communication range will be Cluster Members (CMs). The proposed work in [11] CAVDO is based on the dragonflies algorithm taking direction and velocity of participating nodes in the whole network. Through these basic parameters, the basic network is constructed. Initially all dragonflies have not food and enemy. Then, each dragonfly calculates radius neighboring, food and enemy weights for the entire population. After nodes grouping, each time when CHs are selected, dragonflies will be aligned based on speed, velocity, and direction according to CH. To evaluate the fitness of dragonflies, the authors used a multi-objective function. Overall, the proposals cited above are implemented by some of incomplete and insufficient parameters to select the best CHs in a network. In our work we choose to employ three different metrics in order to elect cluster heads with high QoS, namely: node safety, node density and node mobility.

### 3. PROPOSED WORK

Internet of Vehicles is represented by a directed or undirected graph, depending on the nature of the links,  $G(t) = (V(t), E(t))$  where  $V(t)$  represents the set of nodes at time  $t$  and  $E(t)$  represents the set of links between nodes at time  $t$ .

**Definition 1.** Set  $n = |V(t)|$ ,  $s = |E(t)|$ ,  $e_{ij} \in E(t)$ ,  $e_{ij} = (v_i, v_j)$  represents a link between two nodes  $v_i$  and  $v_j$  at time  $t$ ,  $(v_i, v_j) \in V(t)$ ,  $i, j = 1, 2, \dots, n$ .

**Definition 2.** For any node  $v$ , set  $\mathcal{N}(v)$  is the neighbor nodes set of  $v$ ;  $E(v)$  is the list of links connecting node  $v$  with its neighbor nodes.

**Definition 3.** Non-overlapping clusters are clusters where each node belongs to a single cluster. Set  $C_i$  is the cluster  $i$  and  $C_j$  is the cluster  $j$  with  $(C_i, C_j) \subset V$ ;  $C_i$  and  $C_j$  are non-overlapping clusters if  $C_i \cap C_j = \Phi$ , ( $i \neq j$ ).

**Definition 4.** Set  $d_v$  the node degree that represents the number of neighbor nodes in a  $k$ -neighborhood.

In order to ensure a robust connectivity in IoV environment, we use three metrics that have relationship with nodes connectivity in IoV, each parameter is defined below:

**Node Safety (S).** Safety parameter is an essential factor to guarantee a certain level of safeness. The safety value calculation ( $S_v$ ) of a node  $v_i$  is done in the goal of detecting the level of safety of a node  $v_i$  with its neighboring node  $v_j$  depending on the performance evaluation of the assigned task. In our work we used the method described in [12], which is used to calculate safety metric in mobile networks. Each node in the network has to calculate its own safety value which is based on two computed variables: direct safety value (1-hop neighbors) and indirect safety value ( $k$ -hop neighbors).

**The direct safety (DS)** of any node  $v_i$  is the geometric mean of all safety metrics of various actions that occurred on that specific node. Direct safety ( $DS_{v_i}(v_j)$ ) of a node  $v_i$  on node  $v_j$  is calculated as:

$$DS_{v_i}(v_j) = \left[ \prod_k (p_k) \right]^{\frac{1}{k}}, \quad (1)$$

$p_k$  is a set of  $k$  safety parameters as mentioned in [12].

The indirect safety (*IS*) of node  $v_i$  on node  $v_j$  is based on the *DS* of neighbor node  $v_j$ , and it is calculated as:

$$IS_{v_i}(v_j) = \left[ \prod_m (DS_m(v_j)) \right]^{\frac{1}{m}}, \tag{2}$$

$DS_m(v_j)$  is the calculated *DS* of neighbor node  $v_j$  and  $m$  represents the IDs of neighbor nodes.

The total safety value of each node  $v_i$  is calculated based on direct and indirect safety values. Total Safety  $TS_{v_i}(v_j)$  of node  $v_i$  on node  $v_j$  is given by:

$$TS_{v_i} = \omega_1 DS_{v_i}(v_j) + \omega_2 IS_{v_i}(v_j), \tag{3}$$

where  $\omega_1, \omega_2$  are the weight factors for *DS* and *IS*, respectively, with the constraint that  $\omega_1 + \omega_2 = 1$ .

In our algorithm, we propose to calculate the general Safety value  $S_{v_i}$  of node  $v_i$  that represents the sum of total Safety values on node  $v_i$ . The general Safety value ( $S_{v_i}$ ) of node  $v_i$  is calculated as:

$$S_{v_i} = \sum_{j=1}^n TS_{v_j}(v_i), \tag{4}$$

where  $n$  is the number of  $k$ -hop neighbor nodes of  $v_i$ .

**Node Density (*D*).** The density ( $D_{v_i}$ ) of node  $v_i$  is defined as the ratio between the number of links and the number of nodes in a  $k$ -neighborhood. To calculate the density parameter (*D*) we used Eq. (5):

$$D_{v_i} = \frac{|e = (u, w) \in E, u \in \{v, \mathcal{N}K(v_i)\} \ w \in \mathcal{N}K(v_i)|}{dv_i}, \tag{5}$$

where  $\mathcal{N}K(v_i)$  denotes the set of  $v_i$  neighbors and  $dv_i$  is the number of  $v_i$  neighbors.

**Node Mobility (*M*).** Vehicles moving on the road have different speeds. Furthermore, the speed of a car changes with time. We assume that the node that has the most homogenous speed with its neighbor nodes speeds should take the highest priority to be elected as a CH. By doing so, we can guarantee a long cluster stability. To calculate the priority of each node we use the method described in [13].

$$K_{v_i}^n = (1 - \alpha) K_{v_i}^{n-1} + \alpha F_{v_i}^n, \tag{6}$$

where  $F_{v_i}^n$  denotes the new value of speed and  $K_{v_i}^{n-1}$  the previous exponential-weighted average speed of node  $v_i$ ;  $\alpha$  is a factor  $0 \leq \alpha \leq 1$ , we set its value to 0.5.

In order to determine the convergence of any node's ( $v$ ) speed to the average speed of its neighbors. Each node  $v$  calculates its effective average relative speed based on the average speeds of its neighbors. For a node  $v_i$  we denote  $u_{v_i}$  the average speed of its neighbors. Then,  $u_{v_i}$  is calculated as

$$V_{v_i} = \frac{\sum_{j=1}^N K_{v_j}}{N}, \tag{7}$$

where  $S_{v_j}$  is the speed of each neighbor of node  $v_i$ ;  $N$  is the number of the neighbors of node  $v_i$ . Then, the speed difference between node  $v_i$  and the average speed of its neighbors can be calculated as:

$$M_{v_i} = \left| \frac{k_{v_i} - V_{s_{v_i}}}{k_{\max_{v_i}} - V_{s_{v_i}}} \right|, \tag{8}$$

where  $K_{\max_{v_i}}$  denotes the maximum speed of its neighbors.

After the calculation of the three metrics (*S*, *D*, *M*), the weight value  $W_v$  of a node  $v$  is calculated as:

$$W_v = \omega_1 \frac{1}{S_v} + \omega_2 \frac{1}{D_v} + \omega_3 M_v, \tag{9}$$

where  $\omega_i$  are the weighting factors for the corresponding metrics and  $\sum \omega_i = 1$ .

#### 4. OUR NETWORK ARCHITECTURE

Our network topology is organized in clusters (Fig. 1); each node can take one of the following states: Cluster head, core member, margin member, not-decided and gateway node. Initially all nodes are in the

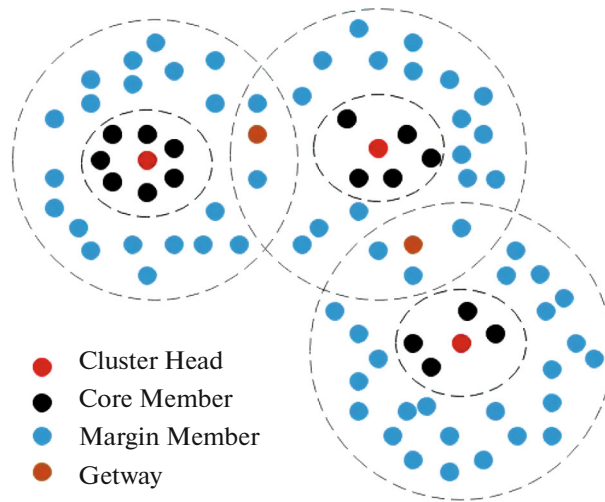


Fig. 1. General structure of our network topology.

not-decided state, as time progress each node attempts to join a cluster by being in one of the three states CH, member or gateway.

- **Cluster Head (CH):** it is the coordinator of the group (cluster). It has additional functions such as: channel access, routing data, bandwidth and channel allocation, forwarding inter-cluster packets, among many.

- **Gateway:** is a node (not CH) which works as the common access point for two or more cluster heads, when a node remains within the transmission range of two cluster heads.

- **Core Member (COM):** is an ordinary node (not a gateway or CH). The cluster-head is one of the core members of the cluster. All members of the core are 1-hop neighbors to the cluster-head.

- **Margin Member (CAM):** is an ordinary node but it is the  $k$ -hop neighbors of the cluster-head with  $k \geq 2$ .

(1) **Neighbor nodes detection.** Initially, all nodes take the not-decided state. Over time, the network nodes exchange periodically HELLO messages in order to share their states and information. The HELLO message includes: the source node address, the node state, weight value, neighbor nodes list, the cluster head address and neighbor clusters addresses. Through such information each node can establish, record and update its neighbors list (keep local topology).

(2) **Cluster heads selection process.** Once all network nodes detect themselves, we apply the cluster head selection process. In this phase, each node has to calculate its weight value ( $W$ ) as presented above and shares it by broadcasting a weight-val (my-id, my-weight) message to its  $k$ -neighbors. After receiving the weight-val message, each node is responsible of updating its table of neighbors. Once all information about  $k$ -neighbors is available, the node that has the lowest ( $W$ ) value, is elected as Cluster Head. If there is more than one node that has the same lowest ( $W$ ) value, in this case we compare by Safety value ( $S$ ), where the node that has the highest Safety value is selected as cluster head. In the case where the problem is not solved the node with highest density ( $D$ ) is elected. After selecting the Cluster Head, the elected CH shares its state to its  $k$ -neighbors by sending a CH\_elect (my-id, my-ch-id) message to all network nodes.

(3) **Cluster node join process.** At the reception of the CH\_elect messages from different CH, each node sends a Join Request (my\_id, my\_Safety\_value) message to its different Cluster Heads. The corresponding Cluster Head responds with an acceptance message, the node updates its state to either core member or margin member.

(4) **Gateway election.** Each Cluster Head is responsible of selecting its gateway nodes. The nodes that have margin member states and they belong to different clusters are candidates to be gateway nodes. The selection of gateway node process is based on the Safety value received in Join Request message. The node that has the highest Safety value  $S$  is selected as gateway node. If there are several nodes that have the same highest Safety value  $S$ , then the gateway candidate with the lowest ( $W$ ) value is selected.

(5) **Cluster Maintenance Phase.** The clusters maintenance phase attempts to update and adapt with topology changes that can be caused by the different movements of nodes. In our proposal there are three distinct kinds of events to apply cluster maintenance algorithm: the node movement, the Safety value change in gateway nodes and the weight value change in the cluster head. In each event, each node updates the information required.

(a) Node Movement

A node in the network has two types of movement. The first one when it enters a range of a new cluster, in this case the new node sends Join Request message to the corresponding cluster head and waits for the response. The second type when a node stays away from its cluster, in this case, the concerned node modifies its state to not-decided and tries to find another cluster to join. For either type, the state of the node should be updated.

(b) Safety Value Change

The parameters values change over time based on interactions between nodes. In our algorithm, all parameters values of each node are verified periodically. When a node acquires new neighbors or loses some due to its movement, the Safety value of this node changes. In the case where Safety value changes, cluster maintenance algorithm should be applied. If the malicious node is the CH, the member nodes have to change their cluster head locally to avoid the re-clustering (more overhead) of the whole network. The nodes seeking to change their malicious cluster head, find the node with lowest weight value amongst the  $k$ -hop neighbors to be their CH. In the case where the malicious node is a member node or gateway then, the CH eliminates this node from its cluster.

(c) Weight Value Change

The nodes movement changes the weight value of nodes by changing their own (safety, mobility, density). In our proposal, the weight value of each node is verified periodically. In the case where the weight value of a member node becomes lower than the weight value of its cluster head then the re-election process of a new cluster head is applied locally. The precedent CH becomes a member node and the member node that has the lowest weight value becomes a CH.

Algorithm 1. Cluster Heads Election Algorithm	Algorithm 2. Gateways Election Algorithm
<p><i>Input:</i> <math>G = (V, E), v_i \in V</math>  <i>Output:</i> CHs = Cluster Heads  <math>S_{v_i} = D_{v_i} = M_{v_i} = 0</math> // initialization  <math>W(v_j) = 0, Cid - v_i = 0</math>  <math>K = 5</math>  <b>For</b> (All <math>v_i \in V</math>) <b>Do</b>          <math>S_{v_i} = \text{Safety\_calculation}(v_i)</math>          <math>D_{v_i} = \text{Density\_calculation}(v_i)</math>          <math>M_{v_i} = \text{Mobility\_calculation}(v_i)</math>          <math>W(v_i) = \omega_1 \frac{1}{S_{v_i}} + \omega_2 \frac{1}{D_{v_i}} + \omega_3 M_{v_i}</math>          Distribute Neighbors (weight-val-msg (<math>W(v_i)</math>))          <b>For</b> ((All <math>v_j \in \mathcal{N}(v_i)</math>) <b>And</b> (<math>v_i \neq v_j</math>)) <b>Do</b>              —Extract and Store the information received from the weight-val messages in its neighbors table.              —Compares the received weights of all its <math>k</math>-hop neighbors with its own weight value.          <b>If</b> (<math>W(v_j)</math> is the Minimum) <b>then</b>              <math>CH v_j = v_j</math> // <math>v_j</math> elected as CH              Status (<math>v_j</math>) = Cluster Head              Cid- <math>v_j = CH v_j</math>              Broadcast_K-Neighbors (<math>CH\_elect(v_j)</math>)          <b>Else</b>              <math>v_j</math> waits reception of <math>CH\_elect</math> message sends by the node that has the highest weight value in its <math>k</math>-neighborhood          <b>End If</b>  <b>End For</b>  <b>End For</b></p>	<p><i>Input:</i> <math>G = (V, E), v_i \in \mathcal{N}(CH v_i), v_i = \text{margin member}</math>  <i>Output:</i> <math>g_i = \text{Gateway node}</math>  <b>For</b> (each neighbor cluster) <b>Do</b>          <b>For</b> ((All <math>v_i \in \mathcal{N}(CH v_i)</math>) <b>And</b> (<math>state(v_i) = \text{margin member}</math>)) <b>Do</b>              <b>If</b> (<math>S_{v_i}</math> is the Maximum) <b>Then</b>                  <math>g_i = v_i</math>                  State (<math>v_i</math>) = Gateway.              <b>Else</b>                  <b>For</b> (((All <math>v_j \in \mathcal{N}(CH v_i)</math>) <b>And</b> (<math>state(v_j) = \text{margin member}</math>) <b>And</b> (<math>v_i \neq v_j</math>)) <b>Do</b>                      <b>If</b> (<math>S_{v_i} = S_{v_j}</math>) <b>Then</b>                          <math>CH v_i</math> chose the gateway node with the lowest weight.                      <b>End If</b>                  <b>End For</b>              <b>End If</b>          <b>End For</b>  <b>End For</b></p>

<i>Algorithm 3. Local Cluster Maintenance Algorithm</i>	<i>Algorithm 4. Node Join Cluster Algorithm</i>
<p><b>For</b> (All active <math>v_i \in V</math>) <b>Do</b>  <b>If</b> (state (<math>v_i</math>) = Cluster Head) <b>Then</b>  —<math>v_i</math> distributes an abandon role message to its cluster members.  —<math>v_i</math> changes its state to not-decided and looks for a new cluster to join it (algorithm 4).  <b>For</b> (All <math>v_j \in \mathcal{N}(v_i)</math>) <b>Do</b>  —<math>v_j</math> changes its cluster information.  —<math>v_j</math> calls the CH election algorithm to reselect a new CH.  <b>End For</b>  <b>Else</b>  <b>If</b> (state (<math>v_i</math>) = core member <b>Or</b> state (<math>v_i</math>) = margin member <b>Or</b> state (<math>v_i</math>) = gateway) <b>Then</b>  —<math>v_i</math> sends a moving message to its CH.  —<math>v_i</math> changes its state to not-decided and looks for a new cluster to join it (Algorithm 4).  <b>For</b> (All <math>v_j \in \mathcal{N}(v_i)</math>) <b>Do</b>  <b>If</b> (state(<math>v_j</math>) = Cluster Head) <b>Then</b>  —changes information about the moving node.  —informing member nodes of the moving node by sending an information message to them.  <b>If</b> (state (<math>v_i</math>) = gateways) <b>Then</b>  —CH node calls the gateway election Algorithm to re-elect a new gateway.  <b>End If</b>  <b>Else</b>  —Updates information related to the moving node.  <b>End If</b>  <b>End For</b>  <b>End if</b>  <b>End if</b>  <b>End For</b></p>	<p><b>For</b> (All moving <math>v_i \in V</math>) <b>And</b> (state (<math>v_i</math>) = notdecided) <b>Do</b>  —<math>v_i</math> sends a request message to the corresponding Cluster Head to join the cluster and waits a <math>T</math> period of for response.  <b>If</b> (no response arrives after time <math>T</math>) <b>Then</b>  —<math>v_i</math> increases its transmission power to reach more CHs.  —<math>v_i</math> sends a request message to other Cluster Head  <b>Else</b>  <b>If</b> (<math>v_i</math> receives one Join Accept message) <b>Then</b>  —<math>v_i</math> joins the corresponding Cluster and updates its state to member node (Core or Margin) and all important information.  <b>Else</b>  —<math>v_i</math> joins the Cluster that has the highest weight value among all CHs that have replied with Join Accept message.  —<math>v_i</math> changes its state to member node (Core or Margin) and all important information.  <b>End If</b>  <b>End If</b>  <b>End For</b></p>

## 5. SIMULATION AND PERFORMANCES EVALUATION

In this section we performed different simulation scenarios and comparison between our proposed work and the recent existing proposals MADCCA [10] and CAVDO [11]. We used a combination of simulation tools in order to obtain a realistic vehicular environment where we used Network simulator NS3 [14] for network nodes interaction, SUMO [15] was used to simulate vehicles mobility. We generated realistic scenarios using MOVE [16], where speed of vehicles varies between 60 to 120 km/h, we used the IEEE 802.11p as a media access type with setting 300 m as the transmission range for nodes with data rate of 1 Mbps.

**Average Number of Clusters:** It is defined as the number of clusters that are already formed in the network since inception to any instance of time with different number of nodes. As shown in Fig. 2, each time the number of nodes increases, the average number of clusters also increases for all algorithms which is logical. The high number of nodes makes network more divisible into several clusters. However, our proposal approach creates continuously fewer clusters compared to other simulated clustering algorithms, this is owed to density metric used in our proposed algorithm that allows to select the CHs that have a high number of neighbors. Moreover, our proposed approach used the concept of  $k$ -hops distance to form clusters that allows to create less clusters comparing to 1-hop gathering process.

**Average Cluster Duration:** It is the time duration between a vehicle becoming a CH and the moment it discharges the responsibility of CH. Figure 3 shows the period of time that clusters can resist with different

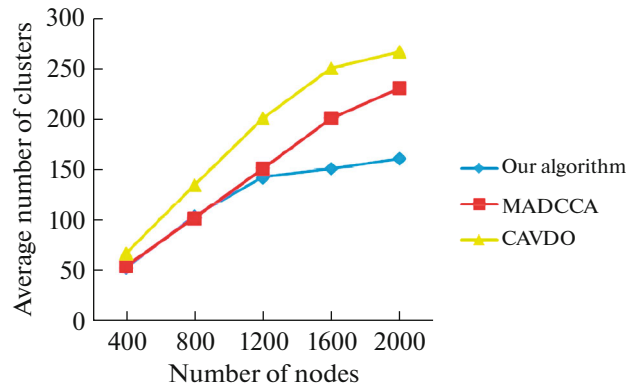


Fig. 2. Average number of CHs vs. number of nodes.

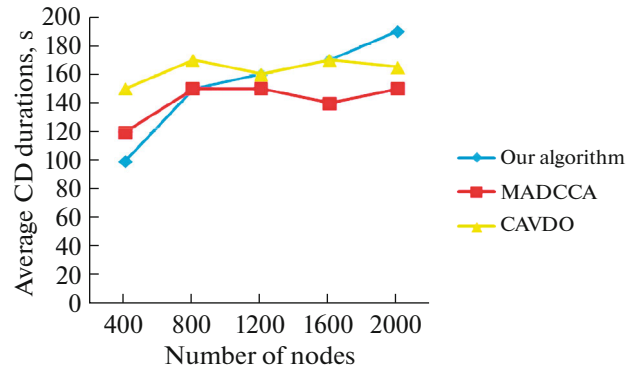


Fig. 3. Average CHs duration vs. number of nodes.

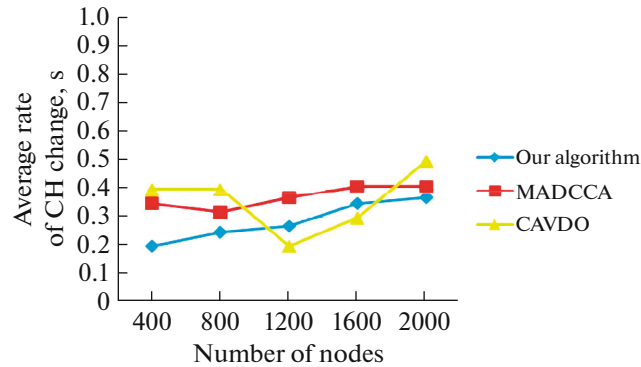


Fig. 4. Average rate of CH change vs. number of nodes.

number of nodes. As shown, when having less than 800 nodes, the MADCCA [9] and CAVDO [10] algorithms perform better than our algorithm, and this can be explained by the small number of nodes that form clusters which led to fast disconnections. Otherwise, when the number of nodes increases to more than 800, our proposal shows better clusters stability compared to other clustering algorithms. Such a good performance is owed to the mobility metric used during CHs election process that keeps connection between neighbors.

**Average Rate of CH Change:** Average Rate of CH Change is defined as the total number of CH changes per second with variation of nodes numbers. Average Rate of CH Change shows the level of cluster’s stability, low Average Rate of CH change means that the selected CH has a robust connection with its cluster members. As depicted in Fig. 4, our proposed algorithm shows the most stable changes and the least variation compared to the other two clustering algorithms. The good behavior of our algorithm is related to

the use of mobility metric in CHs selection process, where the node that has a close velocity to its neighbors is selected as CH. Moreover, Safety metric used in our proposal to calculate weight value allows to choose the trusted nodes to be CHs which in turn ensures stable connection.

## 6. CONCLUSIONS

Clustering can provide large-scale networks like IoV with a hierarchical network structure to facilitate routing operations and network management. In this paper we proposed a new clustering algorithm for IoV environments, where we used a combination of three metrics, namely: node safety, node density and node mobility. For each group of nodes, a Cluster Head must be elected based on the three metrics previously cited. The variety of metrics used to select Cluster Heads may guarantee a high level of Quality of Service in IoV environment such as: high security, long connectivity, links stability and so on. The performance evaluation phase showed a good behavior of our proposed work when compared to other clustering algorithms in different terms and demonstrate that we achieved 95% of our goals to make IoV network more manageable and stable. As future perspectives to improve our solution in the short and long term, we expect:

- Adding more metrics to fit with other road applications like multimedia applications.
- Analyzing the proposed protocol with analytical methods.

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## REFERENCES

1. Contreras-Castillo, J., et al., Internet of Vehicles: Architecture, protocols, and security, *IEEE Internet Things*, 2017, vol. 5, pp. 3701–3709.
2. Gasmi, R., et al., Vehicular Ad Hoc NETWORKS versus Internet of Vehicles – a comparative view, *International Conference on Networking and Advanced Systems (ICNAS)*, Annaba, 2019.
3. Gasmi, R., et al., A Stable Link Based Zone Routing Protocol (SL-ZRP) for Internet of Vehicles environment, *Wireless Pers. Commun.*, 2020, vol. 112, pp. 1045–1060.  
<https://doi.org/10.1007/s11277-020-07090-y>
4. Bodyanskiy, Ye.V., et al., Kernel fuzzy Kohonen’s clustering neural network and it’s recursive learning, *Autom. Control Comput. Sci.*, 2018, vol. 52, no. 3, pp. 166–174.
5. Pavlenko, E.Yu., et al., Application of clustering methods for analyzing the security of Android applications, *Autom. Control Comput. Sci.*, 2017, vol. 51, no. 8, pp. 867–873.
6. Kerimova, L.E., et al., On an approach to clustering of network traffic, *Autom. Control Comput. Sci.*, 2007, vol. 41, no. 2, pp.107–113.
7. Bali, R.S, et al., Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions, *Veh. Commun.*, 2014, vol. 1, pp. 134–152.
8. Zhang, D., et al., New multi-hop clustering algorithm for vehicular ad hoc networks, *IEEE Trans. Intell. Transp. Syst.*, 2019, vol. 20, no. 4, pp. 1517–1530.
9. Tseng, H., et al., A stable clustering algorithm using the traffic regularity of buses in urban VANET scenarios, *Wireless Networks*, 2020, vol. 26, pp. 2665–2679.
10. Ram, A., et al., Mobility adaptive density connected clustering approach in vehicular ad hoc networks, *Int. J. Commun. Networks Inf. Secur.*, 2017, vol. 9, p. 222.
11. Aadil, F., et al., Clustering algorithm for internet of vehicles (IoV) based on dragonfly optimizer (CAVDO), *J. Supercomput.*, 2018, vol. 74, pp. 4542–4567.
12. Bentaleb, A., et al., A weight based clustering scheme for mobile ad hoc networks, *11th International Conference on Advances in Mobile Computing & Multimedia (MoMM2013)*, 2013.
13. Chen, M., et al., A novel mobility-based clustering algorithm for VANETs, *Sens. Transducers*, 2014, vol. 176, no. 8, pp. 189–195.
14. Riley, G.F., et al., The ns-3 network simulator, in *Modeling and Tools for Network Simulation*, Wehrle, K., Güneş, M., and Gross, J., Eds., Berlin–Heidelberg: Springer, 2010.
15. Behrisch, M., et al., Sumo-simulation of urban mobility: An overview, *The Third International Conference on Advances in System Simulation*, 2011, pp. 63–68.
16. Karnadi, F.K., et al., Rapid generation of realistic mobility models for VANET, *Wireless Communications and Networking Conference*, 2007, pp. 2506–2511.