Group'n Route: An Edge Learning-based Clustering and Efficient Routing Scheme Leveraging Social Strength for the Internet of Vehicles

Naercio Magaia, Pedro Ferreira, Paulo Rogério Pereira, *Senior Member, IEEE*, Khan Muhammad, *Senior Member, IEEE*, Javier Del Ser, *Senior Member, IEEE*, and Victor Hugo C. de Albuquerque, *Senior Member, IEEE*

***Abstract*—The Internet of Vehicles (IoV) is undoubtedly at the core of the future of intelligent transportation. It will prevail over the road ecosystem, and it will have a huge impact on our lives throughout the provision of seamless connectivity among diverse transportation means. For the network to operate efficiently, the data needs to be quickly spread throughout the network, which requires low computational and bandwidth overheads. However, the dynamics of vehicular environments due to frequent node mobility poses many challenges to realize efficient data dissemination. This work addresses this type of problem by proposing a novel clustering algorithm at the edge of the network and an efficient message routing approach, which is known as Group'n Route (GnR). Both mechanisms resort to machine learning and graph metrics that reflect the social relationships between the nodes. Our performance evaluation reveals that the clustering algorithm yields stable results with varying road scenarios, which are becoming an advisable approach in the presence of mobile IoV nodes. Also, the designed routing protocol achieves two orders of magnitude smaller overhead and almost double the delivery rate when it is compared to traditional routing protocols, which thereby justify that the combination of our two proposed clustering and routing methods are a plausible alternative to support IoV communications in real-world setups.**

***Index Terms*—Internet of Vehicles, edge learning, clustering, graph theory, routing, social strength**

Manuscript received July 28, 2021; Revised January 11, 2022 and March 21, 2022, Accepted: XXX, Published: XXXX. This work was sponsored by Fundação para a Ciência e a Tecnologia (FCT) under project UIDB/50021/2020 and UIDB/04111/2020, and by Portuguese national funds through FITEC-Programa Interface, with reference CIT "INOV-INESC Inovação-Financiamento Base". This work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2016R1A2B4011712). J. Del Ser also acknowledges funding support from the Basque Government through the Consolidated Research Group MATHMODE (ref. IT1294-19).(Corresponding authors: Naercio Magaia and Khan Muhammad)

N. Magaia is with COPELABS, Universidade Lusófona, Lisbon, Portugal and with the School of Engineering and Informatics, University of Sussex, Brighton BN1 9QJ, U.K and (e-mail: naercio.magaia@ulusofona.pt).

P. Ferreira and P. R. Pereira are with INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal (e-mail: pedro.m.a.ferreira@tecnico.ulisboa.pt; prbp@inesc.pt).

Khan Muhammad is with the Visual Analytics for Knowledge Laboratory (VIS2KNOW Lab), Department of Applied Artificial Intelligence, School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul 03063, Republic of Korea (e-mail: khan.muhammad@ieee.org).

Javier Del Ser is with TECNALIA. P. Tecnologico, Ed. 700. 48170 Derio (Bizkaia), Spain, and the University of the Basque Country (UPV/EHU), Bilbao 48013, Spain (e-mail: javier.delser@tecnalia.com).

Victor Hugo C. de Albuquerque is with the Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, Fortaleza/CE, Brazil (e-mail: victor.albuquerque@ieee.org).

I. INTRODUCTION

THE Internet of Vehicles (IoV) aims to fully use information and communication technologies in order to achieve the coordinated development of humans, vehicles, and environments, which can alleviate traffic congestion, enhance transportation efficiency, and enhance the existing road capacity [1].

IoV is becoming the next transformation in the world of transportation. Its main goals include safety, comfort, and the prompt delivery of the vehicles' occupants with minimal impact on the environment. Keeping this goal in mind, there are several applications for this technology, such as the management of network traffic, the reduction of traffic jams, alerting users about any hazards, calling for specific assistance, and sending information about the victims in case of accidents [2]. The only way these goals can be achieved is through wireless communication among vehicles, pedestrians, Road Side Units (RSUs), and public networks. However, for the IoV to work at its full potential, a large amount of data has to be able to spread throughout the network. The dissemination of this huge amount of data leads to many challenges.

Most of the IoV challenges have also appeared in other fields of study, such as the Internet of Things (IoT), given the similarities between these two areas. The same architecture schema is being used to model both IoV and IoT by splitting them into three layers, which include Vehicles (Things), Edge, and Cloud [3]. The Vehicles layer is mainly responsible for the data collection and actuation in order to control the physical world. Most devices in this layer are resource-constrained in terms of the computational power, storage, and energy. The Edge layer is introduced to help end devices. First, computation-intensive tasks can be offloaded to the edge devices. Second, the edge layer can mask the communication heterogeneity among the end devices and connect them to the Internet. Third, edge devices assist with managing end devices. Finally, the Cloud layer is utilized to store, process, and analyze the collected data and provide the additional support that is needed by many applications [4], [5].

One of the main differences between IoV and IoT is the existence of RSUs, which are stationary units with a lot more computation capacity than the vehicles, which act as intermediaries between the latter and the Cloud. They are one of the main components of the Edge layer. Edge computing, which are computations at the edge, in conjunction with Machine Learning (ML), have turned into a powerful tool for the local decision-making, which is also known as Edge Learning (EL). The EL concept is based on the idea that storage and computation resources should be used at the edge of the network instead of uploading all the data that is collected by the edge devices, such as the vehicles to the cloud, which provide a low-latency, reliable, and intelligent service [6].

However, the dynamics of vehicular environments due to frequent node mobility create many challenges for efficient data dissemination in IoV [7], [8]. The time duration of each vehicle communication link is usually exceedingly limited due to these elevated dynamics. IEEE 802.11p uses a low mobility distributed coordination function (DCF) mechanism, so it is not adequate for vehicular scenarios [9]. The fairness problem that is caused by the different movement speeds of the nodes in the V2I scenarios is another issue, because they do not have equal possibilities of channel access [10].

In order to address the latter issue and guarantee stable and reliable communication between the nodes, we propose in this paper a novel clustering algorithm and message routing protocol leveraging EL and social relations among nodes, which are stable in dynamic networks.

The contributions of this paper are summarized below.

- A novel clustering algorithm at the edge of the network that leverages the social relation among nodes.
- An edge-assisted message routing protocol that uses social metrics and efficient forwarding strategies for connected vehicle networks.
- It is illustrated that both the clustering algorithm and the routing protocol achieve good performances in vehicular networking environments through extensive simulations.

The remainder of this paper is structured as follows. Section II presents the related work. In Section III, the design and implementation of the proposed solution are presented. Section IV presents the performance evaluation and discussion. Finally, Section V presents the concluding remarks.

II. Related Work

Routing on vehicular networks is not a trivial task. Different characteristics, such as the number of vehicles on the network, routing protocols, channel loss, and collisions may directly influence the packet loss and the delivery ratio [11]. However, different studies have already presented a performance comparison of routing protocols, which is described in detail in [12]–[14].

Existing solutions highlighted relevant problems with routing protocols, so vehicular networks still have problems in regards to high latency, packet loss, and high-energy consumption [15], [16]. Seeking to solve the high-latency problem, Abbas and Fan [17] proposed a clustering-based reliable low- latency multipath routing scheme in vehicular networks. The proposed solution uses a cluster head (CH), which is defined based on the link reliability, to manage all nodes' communication. The reliable low-latency routing scheme used the Ant Colony Optimization (ACO) method to reduce the latency, which is due to high mobility nodes. However, it does not consider RSU placement to improve cluster formation.

Qi et al. [18] proposed a solution for routing information to reduce delay on vehicular networks, so-called Traffic Differentiated Clustering Routing (TDCR) mechanism in a Software Defined Network (SDN)-enabled hybrid vehicular network, such as Dedicated Short-Range Communication (DSRC) and Cellular Vehicle-to-Everything (C-V2X). TDCR relies on the vehicles' position and speed, which are reported to the base stations, and a CH is then selected. Furthermore, the proposed solution uses a heuristic algorithm to achieve *a near-optimal solution*. A cluster formation is based on dynamic metrics, which renders them unstable over time, as opposed to using the social relations between pairs of vehicles, which can be more stable.

Nowadays, different technologies may be used in conjunction with vehicular networks. For example, Edge Computing (EC) is a promising technology that helps reduce network and device resource consumption. In [19], the use of Mobile Edge Computing (MEC) in order to cope with the application issues, which are raised by high energy consumption in the Internet of Things (IoT), is investigated. Moreover, the proposed solution provides low-latency processing of visual data and offload decision-making. To allow distributed learning for vehicle routing decisions, Lin et al. [20] used centralized cloud computing to process the transmitted data. Also, an architecture for software-defined Internet of Vehicles (SDIoV) is proposed using edge intelligence technology, which uses a distributed multi-agent reinforcement learning model in order to make real-time vehicle routing decisions, such as the Distributed-Learning-Based Vehicle Routing Decision algorithm, which is based on the asynchronous and parallel behavior of edge devices. Furthermore, the SDIoV solution uses vehicles as terminal devices to start the learning process for routing decisions, and it uses the RSUs as edge nodes to respond to vehicle requests and cloud services. Nonetheless, no message aggregation approach is used to reduce the control overhead by making routing decisions at the edge. In addition, these types of learning models often end up highly biased to the scenario that is being tackled given the variability of the vehicular environment. Furthermore, they demand large training transients, which hinder their generalization and use in practical vehicular settings

Zhou et al. [21] proposed a secure data sharing method to manage smart vehicles' access control by using identity-based encryption and a cyber-attack detection method using deep learning techniques to analyze the anomalies in the traffic and filter out the malicious packets of 6G-enabled VANETs. Implementing appropriate security measures in such networks is challenging because of the large amount of data generated, the limited computation power and network compatibility. Nevertheless, Edge learning in 6G-enabled networks can address such limitations.

Groups of vehicles can generate an entire ecosystem, and they have social characteristics, such as the drivers' behavior and the vehicles' mobility. Qi et al. [22] used a semi-Markov model in order to perform pattern prediction that was aimed at "the future state transition direction according to the current state," such as vehicle movement and position by applying the SDN-Enabled Social-Aware Clustering approach for VANET-

TABLE I
DISTRIBUTION OF RSUS

1 node	centre (0*radius)
5% of the RSU	circle (1*radius)
10% of the RSU	circle (2*radius)
15% of the RSU	circle (3*radius)
20% of the RSU	circle (4*radius)
25% of the RSU	circle (5*radius)
25% of the RSU	circle (6*radius)
rest	circle 7*radius

based 5G. Moreover, the clustering process occurs based on the vehicle social pattern and the CH, which is selected based on the inter-vehicle distance, the relative speed, and other facets. By exploiting the social patterns prediction model to enhance the stability of the clusters, the CH selection is based on dynamic metrics, such as distance and speed, which again yield unstable cluster formations as exposed above.

Furthermore, Paranjothi et al. [23] proposed a message authentication using social networks in VANET (MAvanet). The main strategy for the message authentication uses a Quick Response code (QR code) technique to allow only sending and receiving vehicles to read the content of the message, which involves performing QR encryption and decryption, respectively. In the encryption phase, the sender vehicle verifies the receiver's ID and location. Once the message is successfully verified, it is encoded using the Reed-Solomon algorithm with error setting correction level. In the decryption phase, to solve the QR code security issues, an authenticating method was used to verify the sender's ID at the receiver end. The proposed system only supports V2I communications.

Moreover, applying the clustering approach in vehicular networks is a known technique for routing [24], [25]. The proposed solutions aim to reduce delay, packet loss, and network and device resources consumption, but these problems are still present in these type of networks.

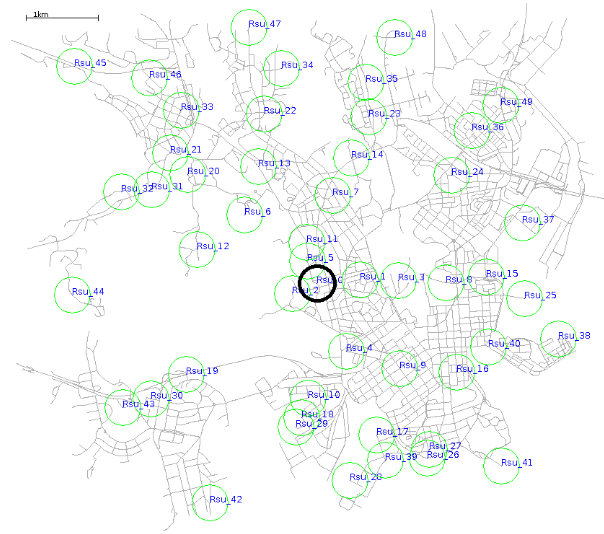
MMEC technology has been used to create geo-clustering for the reduction of resource consumption [26]. Han et al. [27] used EC technology for the CH selection on the Internet of Medical Things (IoMT). Collecting data on IoV can generate a relevant data source. The most straightforward approach to clustering only takes into account the geographical location of the nodes, which mostly occur without any consideration of the role of every node (vehicle) in the relational dynamics of the network.

III. DESIGN AND IMPLEMENTATION

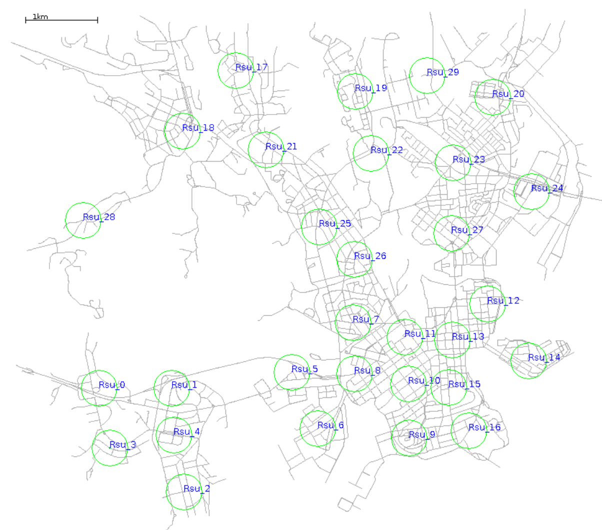
A. Network Architecture

This section describes the IoV network architecture. It covers the types of network nodes, the communication that is used, and the RSU placement.

1) System Model: Figure 2 presents the proposed IoV network that is composed of three layers, which include Vehicles, Edge, and Cloud. There are mainly two groups of nodes: mobile and stationary. In the first group, there are cars and buses in the Vehicle layer. In the second group, there are the RSUs in the Edge layer. It is assumed that all the nodes are equipped with a wireless interface that uses an 802.11p



(a) Automatic placement option



(b) Manual placement option

Fig. 1. Distribution of RSUs.

Wireless Access in Vehicular Environments (WAVE) interface [28] with a transmission range of 100 m and a transmission rate of 10 Mbps in order to connect to each other. In addition, each vehicle is also equipped with a more powerful wireless interface that can only be used to connect with the RSUs. This interface has a transmission range of 250 m, and the same transmission rate.

Mobile nodes can offload data to the nearest RSU to be processed. Once the computation is finished, the output will be sent through other nodes in the direction of the desired node. Nevertheless, the RSUs are all connected through the Edge, and, thus, some data can be shared between them.

2) RSU Placement: RSUs are stationary nodes that are placed near road intersections [29]. Two options were considered when the number of RSUs was decided and their position. In the first option, the user only chooses the number of RSUs and a position. From this position, concentric circles with an increasing radius are created, and the RSUs are distributed

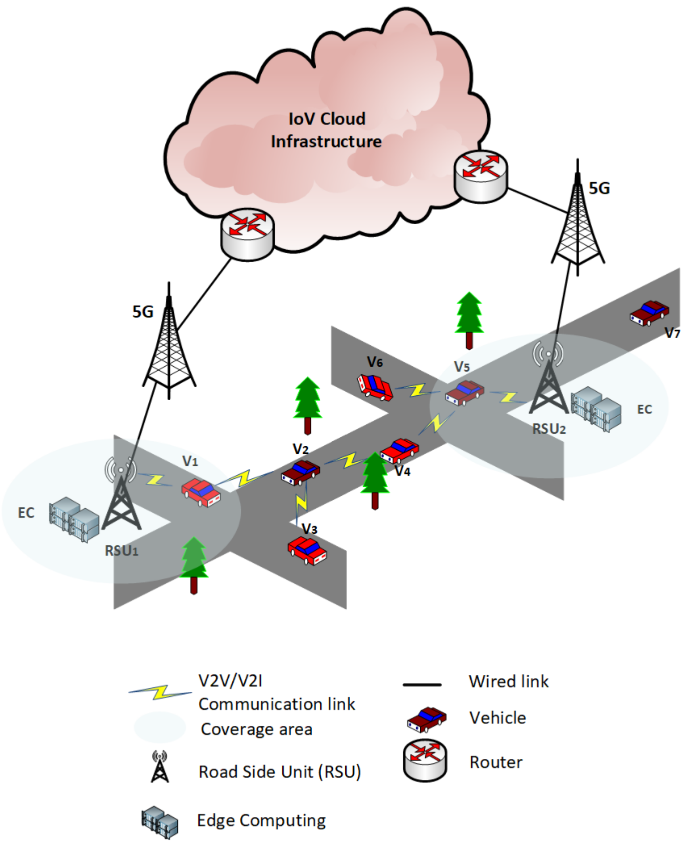


Fig. 2. The proposed IoV network.

among them, which are shown in Table I.

The circles are divided according to the number of nodes that are assigned to each. A position for each RSU is then computed. With these coordinates, a check is made for a valid position in their proximity, and the RSU will assume the valid location when it is found. If there are not any valid map positions, the RSU will be placed in a 7th circle. An example of this distribution is shown in Figure 1(a) with 50 RSUs centered in RSU_0, which has the transmission range marked in black.

It was noticed during the evaluation of this option that some important points and routes were not covered by this model. Therefore, a different approach was needed in order to take the latter into account. The map was consequently analyzed, and the RSUs were placed one by one in strategic locations, which included the main road interceptions and access to the city points of interest, such as the city center. An example of this option is shown in Figure 1(b).

This RSU distribution only used a portion of the total number of RSUs to cover all the city, and it presents much better results. However, this option requires prior knowledge about the city. Therefore, the first distribution option suffices for a simple setup. The second approach could be transformed into an optimization problem with more data, which reduces the number of nodes and improves the network performance.

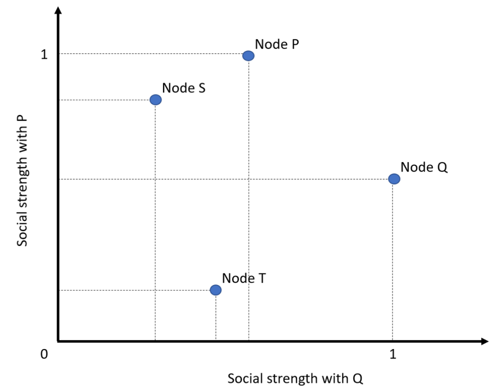


Fig. 3. Simple example with only 2 dimensions P and Q.

B. Edge Learning based Clustering

Our goal in this study is to cluster the IoV nodes. It is assumed that mobile nodes move in completely different directions.

There are countless ways to cluster nodes in a network, and clustering them based on their position is the most trivial method. However, this leads to a network stability problem.

For instance, consider that at a given time, a picture of all cars in a big metropolis, such as Lisbon was taken, and they were assigned to clusters according to their positions based on that picture. How different would these clusters be if the photo was taken 1 minute later?

In light of the above scenario, a more stable metric should be considered, such as the nodes' social relationships. Now, consider that every time node i connects with node j, their social strength is increased. This metric decreases if node i and node j do not connect during a given period. It is also assumed that every node updates its social relationship metric periodically, such as every hour. Thus, nodes can be clustered according to the similarity of their relationships, which is explained below.

Each node has a data structure with a social strength between it and each one of its contacts. An example of a network with nodes P, Q, S, and T is is provided. The corresponding data structure is shown below, where Q_P is the social strength between nodes Q and P in node Q's data structure.

Node Q	Node P
$Q_P = 0.5$	$P_P = 1$
$Q_Q = 1$	$P_Q = 0.5$
$Q_S = 0.3$	$P_S = 0.7$
...	...

These nodes can be represented in a Euclidean space with N dimensions, where each dimension represents the relations with a given node, which is shown in Figure 3. Therefore, the similarity between the nodes will be measured by the distance between them in this Euclidean space.

Let $\mathbf{x}_i$ be an embedding for node i, and x_i^n (i.e., the n-th component of $\mathbf{x}_i$ for all $n = 1, \ldots, N$). The Euclidean

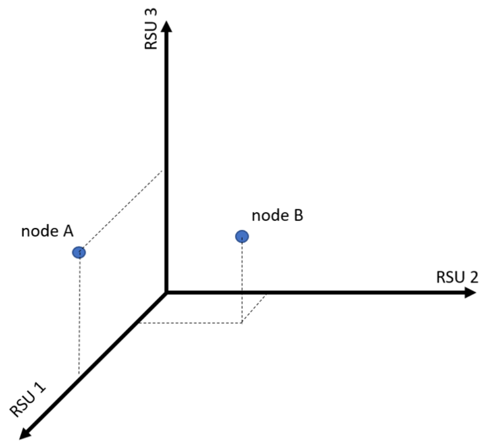


Fig. 4. Simple Euclidean space with 3 RSUs.

distance is given by:

$$d_{i,j} = d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{n=1}^{N} (x_i^n - x_j^n)^2}, \quad (1)$$

from which it follows that $d_{Q,P}$ is the similarity between nodes Q and P.

By considering that RSUs also store their social strengths, if the Edge merges all these data, it could be presented as a Euclidean space with N dimensions, where N is the number of RSUs, and each node would be represented by its social strength, which is shown in Figure 4. Node A has never connected with RSU 2, and node B has already connected with all RSUs.

Based on this representation, which takes into consideration the algorithm complexity, for a number of mobile nodes that is much higher than the number of stationary ones, which is suitable for live computation, the K-Means algorithm (Algorithm 1) is used to compute the clusters. This algorithm will run a predefined number of iterations each time it is called. The number of iterations is a balance between the resource consumption and the algorithm accuracy.

The occupation of all clusters will be checked just after K-Means finishes, and if one or more of the clusters are empty, their centroid will be placed over a point of the largest cluster, and the algorithm will run a few more iterations. There will then be no empty clusters, and their sizes will be balanced.

Once all the nodes are divided into clusters, the edge will choose the Cluster Head (CH), which is a node that represents the cluster and gathers all the cluster's messages, to deliver them elsewhere in the network, which reduces the network overload. The CH selection is based on two metrics that include (i) the similarity between each node and the centroid that represents the cluster and (ii) the ego betweenness centrality of each node in the network. The similarity value varies between 1 and 0. However, the value that represents the ego does not. Thus, a logistic function is used in order to map the ego and the similarity values, which is called the *normalized ego*, and it is illustrated in Figure 5. The ego metric concerns the centrality of a node in a graph of all the nodes and their connections. This function is chosen, because only a

Algorithm 1: The K-Means Algorithm

```
begin
    numCluster ⟵ k (number of clusters)
    for 1 : numCluster do
        create centroid
    for each node do
        minDist = ∞
        for each centroid do
            dist ⟵
              distance between node and centroid
            if dist < minDist then
                minDist ⟵ dist
                assign node to centroid
    for each centroid do
        centroid position ⟵
          mean position of nodes assigned to centroid
```

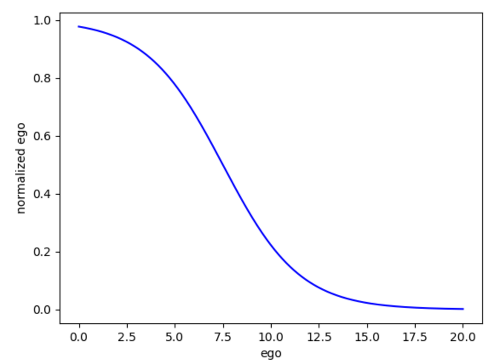


Fig. 5. Function to map the ego.

small percentage of the nodes have an ego that is larger than 7.5, which should be considered to be the CH. The CH is node i, which has metric mi and produces the lowest value.

$$\mathfrak{m}_i = 2 \cdot \hat{c}_{\mathrm{EBC}_i} + d_{i,centroid}. \quad (2)$$

where $\hat{c}_{\mathrm{EBC}_i}$ is the normalized ego and $d_{i,centroid}$ is similarity between i and the centroid.

Eq. (2) gives extra weight to the ego metric in order to increase the importance of a smaller normalized ego over the similarity to the centroid.

The main disadvantage of this approach is that when a node does not connect with an RSU for a significant time, it gets closer and closer to the origin of the Euclidean space. If this happens with a significant number of nodes, a cluster will contain inactive nodes that do not have any relations between them. Therefore, if the distance between the representation of the node and the origin of the Euclidean space is two orders of magnitude below the clustering threshold, which represent the nodes that do not get in contact with an RSU for a long time, these nodes will not be eligible for clustering, and they are considered nodes without a cluster.

Figure 6 shows an example of a network that is composed of 30 RSUs, which are marked with a blue colored label, and 56 vehicles, which are marked with a colored label, that

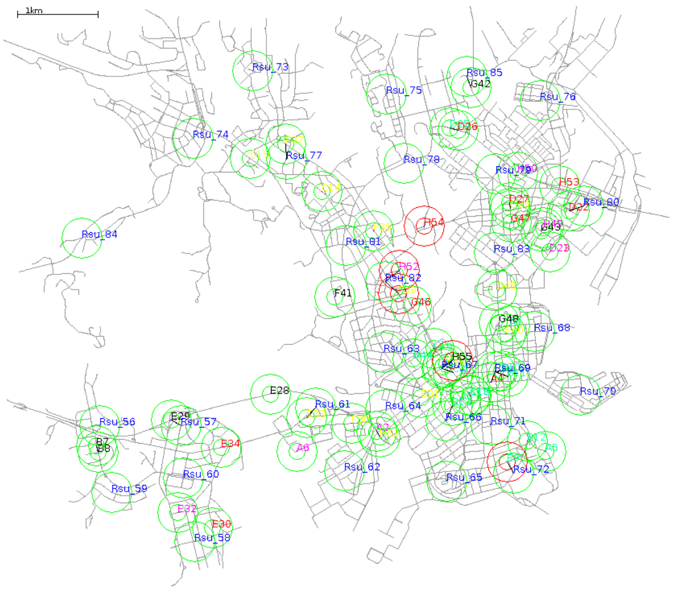


Fig. 6. Nodes divided into clusters.

represent one of the 5 clusters the vehicle belongs to, which are colored using red, yellow, black, pink, and cyan. Each RSU has one circle around it that represents its transmission range. Each vehicle has two circles around it that represents the transmission ranges of their wireless interfaces. The CH vehicles are marked with red transmission circles, whereas the other nodes are marked with green transmission circles.

C. The Routing Protocol

It is essential to fully use the information and communication technologies in order to achieve the coordination of humans, vehicles, and the environment, which was previously mentioned. Therefore, a reliable and efficient routing algorithm is mandatory, and it should comply with the IoV paradigm that is presented above. The following aspects need to be taken into account while designing it.

First of all, it is assumed that the messages are carried through the nodes and not the edge. This is important. The RSUs can communicate among themselves, but they mainly share network state data. In regards to the impact of clustering with routing, some constraints are in place to ensure that CHs are preferably the connection point between the clusters and the RSUs. Nonetheless, if two nodes are within the communication range and one node has a message for the other node, the message will be delivered even if they belong to different clusters.

Our routing protocol, which is called Group'n Route (GnR), was built upon [29] leveraging on its routing metrics and forwarding policies. The following routing metrics are considered: the inter-contact time, average separation period, similarity to the destination (*Sim*), ego betweenness centrality (*EgoBC*), and the mean time to encounter (*MTTE*). The inter-contact time metric is estimated for each pair of nodes using an exponential weighted moving average to update the values, which are based on the previous data. The average separation period is represented by the mean time to encounter.

Our protocol also uses three forwarding policies: direct single copy, direct multi copy, and limited multi copy. In the first policy, each message can only be sent from one node to another node without the sender keeping a copy. While on the second one, all the nodes keep copies of all the messages that pass through them. Finally, the last policy has a limited number of times a message can be copied.

The Routing Algorithm: Each node runs the Routing Rules algorithm (Algorithm 2) in order to select which of the nodes in range can receive messages. With this set of nodes, the Basic Routing algorithm (algorithm 3) runs to check if the other node is preferable compared to the current node.

The connection between RSUs is only used to share (i) what is the closest node to the destination and (ii) what is the RSU with the highest ego betweenness centrality. This information will be used to forward messages in the direction of the best RSU or to the most central one when the destination node has never made contact with any RSU. This connection is never used to send messages from one RSU to another.

In addition to the Routing Rules algorithm and to avoid routing circles, the messages are not transferred to a node that has already seen them.

The following aspect is also considered by our routing approach. When the limited multi copy policy is used, there are times that instead of just forwarding the message, the message carrier just replicates it to the other node. The goal is to replicate messages to the nodes with a higher probability of meeting the destination node. However, the decision to do this is based on the type and cluster of the message carrier. If the node is an RSU or it is from the same cluster of the destination node, it always replicates the message. Furthermore, if the similarity between the carrier of the message and the destination is higher than a threshold, the message is also replicated and transferred. Otherwise, the node will just transfer it.

IV. Performance Evaluation

A. The Scenarios

The proposed IoV network was implemented using the ONE simulator [30]. The following simulation scenarios were considered:

- *Helsinki Wworkday (HW):* This scenario consisted of 56 nodes that are divided into 5 groups, which included home, office, and meeting spots that followed the workday movement (WDM) model. The WDM simulates the usual movement of a person that spends the night at home and then goes to work for 8 hours. After that, there are a few meeting points, which replicate stores, cinemas, restaurants, and other places where people may go after work. There are also 2 buses for each of the 8 routes available on the scenario. The cars' buffers vary from 64 to 256 MB, whereas all the buses have 256 MB. In addition, there are also 30 RSUs.
- *Helsinki Fast Movement (HFM):* This scenario uses the same number of nodes with the same configurations as the previous scenario. However, the shortest path movement is used instead of the WDM. This movement model

Algorithm 2: Routing Rules

```
begin
  if destination cluster = mycluster or destination node has no cluster then
    if the destination node has a cluster then
      run basic routing algorithm only with nodes from the same cluster and RSU
    else
      run basic routing algorithm with all nodes
  else
    if current node is a cluster head then
      send the message only to RSU
    else
      if current node is a normal node in a cluster then
        using the basic routing algorithm considering nodes in the same cluster, RSU
        or the cluster head
      else
        if current node is a RSU then
          use the basic routing algorithm to send the message in the direction of the RSU
          that has the highest social strength with the destination node
          if it is already the best RSU send the message to a node from the same cluster as
          the destination
          if the destination has never connected with a RSU send to a node that has already
          connected with the destination node or in the direction of the most central RSU
        else
          if current node does not have a cluster then
            use the basic routing algorithm to send the message considering all nodes
```

Algorithm 3: Basic Routing Algorithm

```
begin
  for all node's connections do
    otherNode ⟵ nodeconnected
    if
     currentNode MTTE < otherNode MTTE
     or
    currentNode Sim < otherNode Sim then
      send the message to the otherNode
    else if
     currentNode Sim == otherNode Sim
     then
      if currentNode EgoBC *
       currentNode MTTE <
       otherNode EgoBC * otherNode MTTE
       then
        send the message to the otherNode
```

assigns a set of points of interest to each group of nodes, and they go from one point of interest to another in random order. With this movement model, each node would only stay in the same spot for a few minutes, which increases the average movement of the mobile nodes and reduces the time the network takes to converge.

B. Network Architecture Evaluation

Two RSU placement implementations were made, which is mentioned in Section III-A. One implementation is where the user gives the number of RSUs and the central position, and the remaining RSUs are placed automatically. The other implementation is where the positions were assigned one by one. In order to compare these approaches, the following metrics are considered: (i) the average number of nodes that are seen per RSU, (ii) the number of RSUs that never had contact with the other nodes, and (iii) the frequency of the contacts. The first metric indicates how close the average of the RSUs is to the center of the network neighboring graph. An RSU with a high number of nodes is seen as having more connections than the other. The second metric shows the number of RSUs that never made contact with a node, such as an RSU that has not contributed to the network at the time when the data was acquired. Finally, the frequency of the contacts shows on average how close the RSUs are from the crowded areas.

The following tests were made on the HW scenario. All the simulations were run several times, and the results below refer to the average that was obtained with a 95% confidence interval.

A study that regards the number of RSUs that are needed for the auto-placing approach is first performed. The goal is to find where the bottleneck point for this problem occurs as well as the minimum number of RSUs that are needed in order to obtain the best results. Five different simulation configurations, which included 20, 30, 40, 50, and 60 RSUs, were analyzed

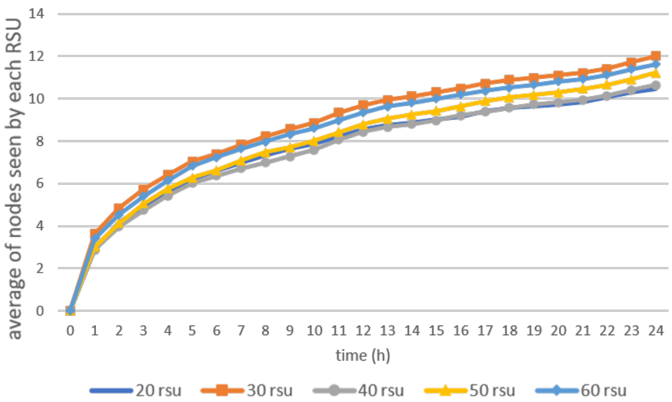


Fig. 7. Average number of nodes seen by the RSUs over time.

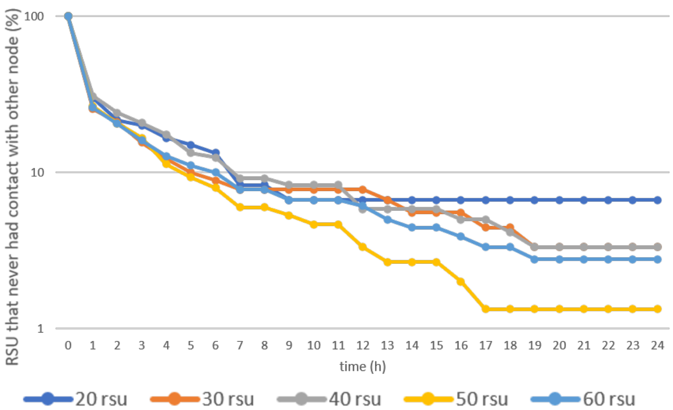


Fig. 8. Percentage of RSUs that never had contact with other nodes over time.

by keeping this in mind.

1) RSU Network Coverage: The first test measured the RSU network coverage by counting the number of contacts with different vehicles each RSU had, which is shown in Figure 7. This metric measures how well the RSUs are positioned. The highest values represent a network where the average RSU have multiple contacts with different vehicles. Therefore, they are placed next to the main roads. Moreover, if there are a lot of RSUs located at places without any movement, the average will also be lower.

Figure 7 shows that the network coverage was similar for all the simulations. There are 56 nodes, but the average number of nodes seen by each RSU is only 12. This is due to the movement model that was used, which assigns a routine to each mobile node. Thus, there is a high probability of a node moving only on a part of the map, which leads to not making contact with many RSUs. In regards to the results that are obtained for each configuration, using only 30 RSUs presented the best results. This is explained by the fact that having a reduced number of RSUs, less of them have zero contacts and some were placed closer to optimal positions.

2) Average Number of Mobile Nodes seen by RSUs: The percentage of RSUs that never had contact with other nodes over time is measured to identify how many RSUs are being used as well as the nodes that do not contribute to the network. The corresponding results are shown in Figure 8 using a logarithmic scale. The percentage of RSUs that never had contact with the other nodes is below 10% for all the simulations after 7 hours. Figure 8 shows a higher percentage of RSUs for the test with 20 nodes. However, 10% of 20 is just 2 nodes. In addition, the relation between the values of the simulations for 30, 40, and 60 nodes should be noticed, because they are close.

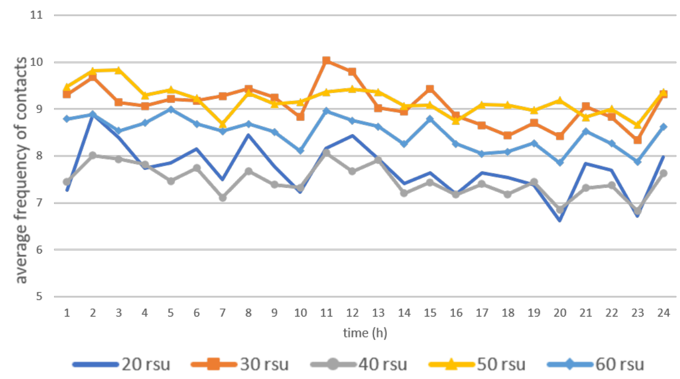


Fig. 9. Frequency of RSU contact over time.

3) Frequency of contacts: The frequency of contacts made by each RSU was also measured. Figure 9 shows the data for all the different configurations that were tested. The higher the frequency, the better the RSU position is. Therefore, if the average frequency is high, the RSU is at an optimal position. It should be noted that the variation in the values obtained to the frequency of contacts is due to the movement model that was used. The results illustrate, which are shown in Figure 9, that the configurations with the best frequencies are with 50 and 30 RSUs. For these scenarios, the average number of nodes that are seen by each RSU is the same. However, in regards to the percentage of RSUs that is placed in irrelevant places, which is shown in Figure 8, the results for 30 nodes are similar to the results with 60 RSUs, which is only beaten by 50 by a small margin.

The frequency of contacts for the scenarios with 30 and 50 nodes presented the best results. The number of nodes that had the best performance was for 50 nodes. Nevertheless, the results were similar and the deployment is less expensive when 30 nodes are used.

4) Further analysis of configuration with 30 RSU: Figure 10 presents a comparison of the cumulative number of nodes seen by each RSU over one day for the two RSU placement approaches that were studied, which included automatic placement and manual placement. As shown, the average number of nodes that made contact with each RSU always increased for both implementations. However, when the nodes where hand placed in strategic locations, the average number of nodes that were seen by each RSU was always 30% higher than it was for the automatic approach.

In regards to the number of RSUs that never had contact with a mobile node, Figure 11 shows a comparison between the values of the two approaches per hour. In the first analysis, a tendency to zero can be seen for both placement approaches. However, when the nodes were placed automatically, the RSUs

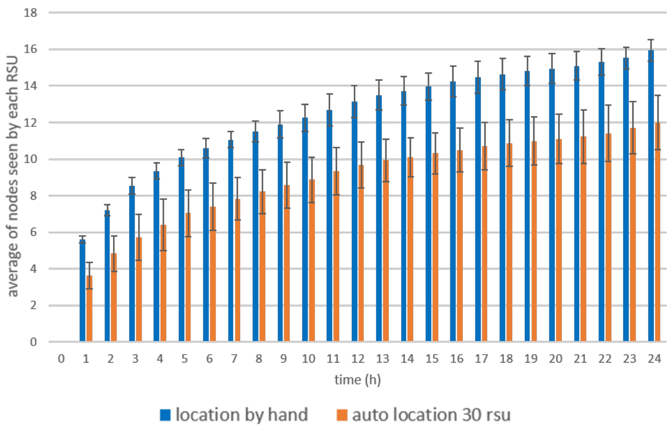


Fig. 10. Average number of nodes seen by 30 RSUs over time.

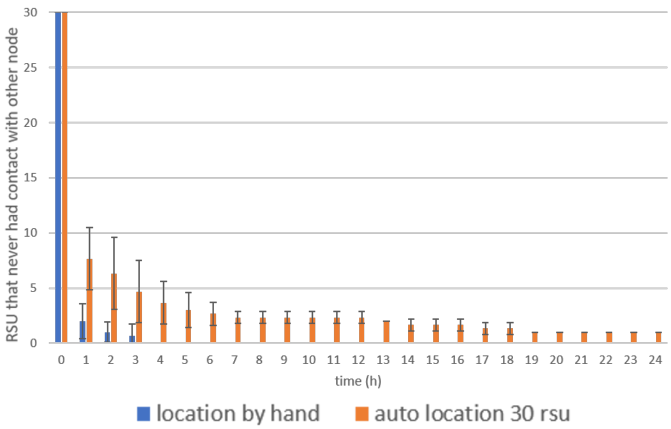


Fig. 11. Number of RSUs that never had contact with other nodes over time.

needed more than 24 hours to reach that value. On the other hand, when the nodes were placed in strategic locations, zero is reached in only 4 hours.

Figure 12 presents a comparison between the two approaches in regards to the average number of contacts per hour that were made by each of the RSUs. This figure shows the main differences between the two techniques and the impact of choosing optimal locations. When the RSUs are hand placed, the frequency of contacts is more than twice as high compared to using the auto placing approach. The RSU placement can consequently have a major influence on the network.

In summary, the RSUs have a huge impact on the network's performance, and they should be accessible to the highest number of mobile nodes that are possible. Moreover, the complexity and deployment costs of several IoV applications are conditioned by the number of RSUs.

C. Edge-based Clustering Evaluation

The IoV network nodes were grouped into clusters aiming to cover as many nodes as possible and creating clusters that were as stable as possible, which was previously stated. In addition, the higher the number of clusters, the more specific is the information that a cluster gives about its nodes. Nevertheless, the size of each cluster shall be taken into consideration, that

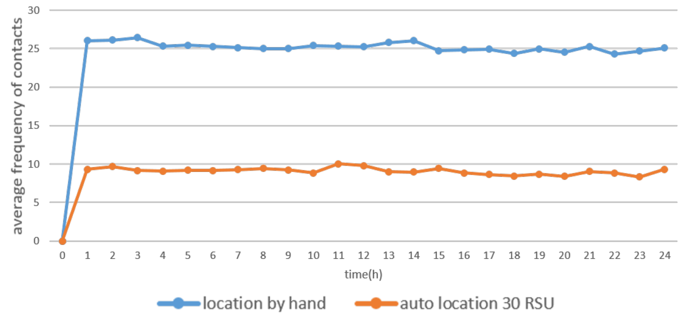


Fig. 12. Frequency of RSU contacts over time.

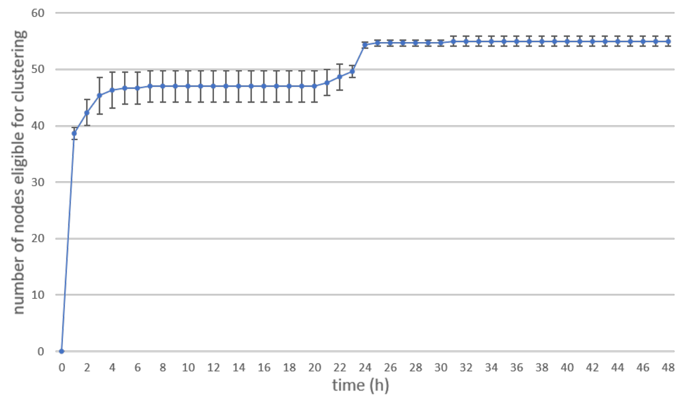


Fig. 13. Number of nodes eligible for clustering over time.

is, the number of clusters shall be much lower than the number of nodes.

In order to evaluate the performance of the proposed algorithm, the number of nodes eligible for clustering and the number of cluster changes per hour were measured. Each simulation was conducted several times with different seeds, and the results presented refer to their average for statistical confidence.

In regards to the number of nodes that are covered by the algorithm, it is important to mention that only the nodes that already made contact with an RSU are known by the Edge and are eligible for clustering.

Figure 13shows that during the first two days, the nodes start making contact with RSUs, which leads to an increase in the number of nodes that are covered. The most significant changes occur in the morning and in the evening, which is when the nodes are supposed to move between home, office, or meeting spots. The values are much more stable during work time. This behavior needs to be taken into account when studying the stability of the algorithm. However, there are a few times that the value decreases, which is not illustrated in Figure 13. This is explained by a node that did not make contact with an RSU, and its relationship decayed below the clustering threshold. These nodes are excluded from the algorithm, and they were not assigned to any cluster.

In order to further analyze the clustering algorithm, a cluster change metric is defined where all cluster changes count as one: (i) when a node does not have a cluster and joins one, and (ii) when a node changes from one cluster to another.

Both situations only count as one cluster change. In the simulation with only one cluster, all the nodes that are eligible for clustering belong to the same cluster. Therefore, all the changes will be from the nodes that were added to the cluster set. In addition, the number of clusters is varied between one and eight, which is based on the total number of nodes in the network.

In Figure 14, each color represents a different simulation configuration, and each vertical bar shows the sum of the nodes that had a cluster change during each hour. For example, during the second hour, there was a total of 65 cluster changes, but there was just an average of 7 changes when using two clusters. It can be observed that there are not any cluster changes counted in the first hour. This is because all the changes in this period were nodes that made first contact with the RSUs.

To give a clearer image of when the network starts to converge, an average of all these simulations was done, which can be seen in Figure 15. In this figure, all the cluster changes are represented, which include nodes that connect for the first time with an RSU and the nodes that change from one cluster to another. This is why there are values during the first hour of the simulation.

Figure 15 also shows that it takes an average of six to seven hours to reach a stable point. Furthermore, with the increase in the number of clusters, a decrease of the stability can be seen, which is shown in Figure 14. However, it can also be seen that even for the less stable configurations, the number of changes per hour tend to be less than 1 after 48 hours, which shows that the approach that was developed leads to a stable clustering solution.

Furthermore, the stability of clustering based on proximity is also analyzed. The goal is only to give a comparison with the most trivial approach to the problem. Therefore, only a superficial analysis is done. The contacts that a node had here during two consecutive time slots were compared, and the number of changes were counted. During the first slot, the number of changes was 11, which then dropped to 8, and every time the node moved the values were similar. When the node was not moving, these values were near 1. Nevertheless, these values are just for one node, which means that it represents only 1/50 of the cluster changes. Therefore, the difference between the stability of the social based approach and this approach is evident

D. The Routing Protocol Evaluation

In order to evaluate the performance of the proposed protocol, which is the GnR, the following metrics were considered: delivery probability, which is the probability of a message reaching its destination, the overhead ratio, which is excessive messages being exchanged, the average latency, which is the delay between the creation of a message and its delivery to the destination, the average hop count, which is the number of nodes that are needed to deliver a message, and the average buffer time, which is the time that a message stays in the buffer of a node.

As it was seen in the previous section, it takes almost 8 hours for the network to converge. Most nodes stay still for long periods of time using the WDM. Therefore, the HFM with 30 hand placed RSUs was chosen for this study instead of using the HW scenario, As a result, all the nodes were eligible for clustering after one hour, and the relations between the housing, working, and meeting locations were preserved. Nevertheless, the clustering algorithm was chosen to run with 6 clusters considering the number of nodes in the network as well as the results that were obtained in the previous section.

To evaluate this algorithm, the following tests were performed for different forwarding policies. One test specifically used the direct single copy policy (DSCP) and two tests used the limited multi copy policy (LMCP) with 6 and 10 message copies [29]. The results are presented in Table II.

The discussion below focuses on the delivery rate and the overhead ratio of each solution. The results in II are according to what is expected, because an increment of the number of copies leads to a boost of the delivery rate. However, this increment also had a negative impact on the overhead. The overhead increased 55%, whereas the delivery rate only increased 3% by comparing the gain of having 10 over 6 copies.

Therefore, the routing algorithm that used the LMCP with six copies was chosen for GnR. This evaluation is performed by comparing its delivery rate and the overhead ratio with four different routing protocols, which include Epidemic [31], Prophet [32], BubbleRap [33], and ePRIVO [29].

The following tests were performed under the same circumstances in order to compare the forwarding policies. All the tests were run several times with different seeds to generate pseudo random movements, and all the results presented are the averages. All the simulations included both vehicles and RSUs. However, the RSUs had a different behavior in our routing protocol than the other nodes of the other routing protocols. The only difference between them is the buffer size and the movement, as the RSUs are static.

In the first instance, the delivery rate of each protocol is compared. As shown in Table III, all protocols have delivery rates between 45 and 93%. Prophet had the best performance, followed by GnR, ePRIVO, Epidemic and BubbleRap.

There is an enormous difference between the protocols in regards to the overhead ratio with values that range between 5 and 2753, which GnR and ePRIVO present the best values by far. This disparity of values was expected, because Epidemic and Prophet do not assign any limit to the number of copies. Epidemic replicates messages to every node it makes contact with, whereas Prophet replicates messages only to all the nodes that have a better metric to the destination than itself. By reducing the overhead, Prophet achieves a better delivery rate than Epidemic. BubbleRap uses social metrics in order to replicate the messages to a more central node until a node that belongs to the community of the destination is met, which will eventually deliver the messages. ePRIVO and GnR have a limit on the number of copies, which is 6, that limits the overhead.

The same tests were also performed using the HW scenario with 30 hand placed RSUs. This scenario represents a situation that is closer to reality with less movement, which has less contacts than the previous scenario.

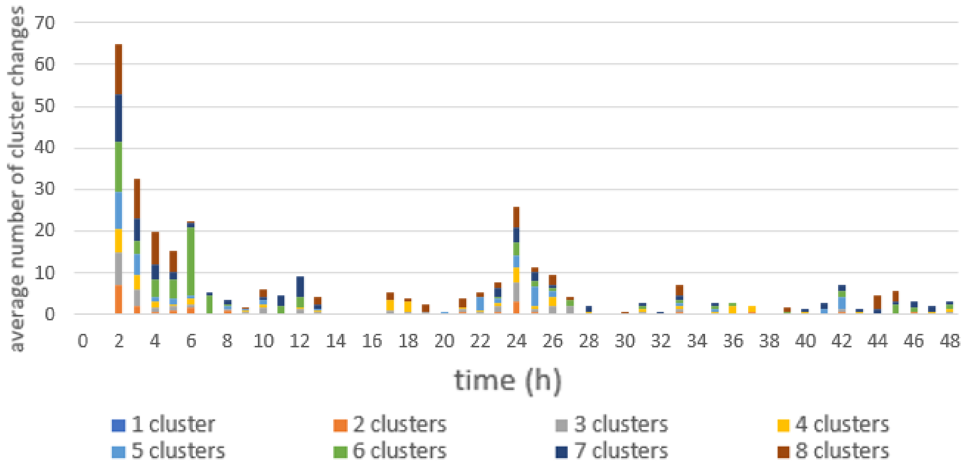


Fig. 14. Sum of the average cluster changes over time.

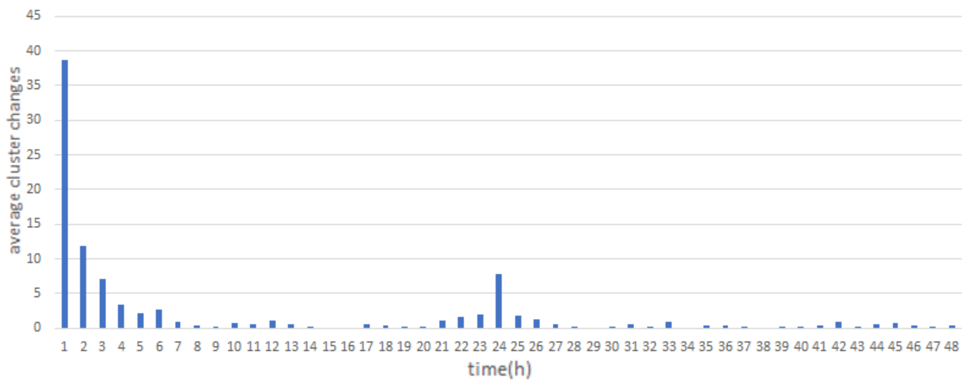


Fig. 15. Average cluster changes over time.

TABLE II
ROUTING POLICY PERFORMANCE FOR THE HELSINKI FAST MOVEMENT (HFM) SCENARIO

Policy	Delivery rate	Overhead	Latency	Hopcount	Buffertime
DSCP	0.6288 ± 0.077	0.0 ± 0.0	9682 ± 996	1 ± 0.0	9682 ± 996
LMCP 6	0.8172 ± 0.077	5.53 ± 0.65	6225 ± 972	1.8 ± 0.018	4188 ± 522
LMCP 10	0.8486 ± 0.05	8.62 ± 0.59	6343 ± 1367	1.86 ± 0.03	3459 ± 720

TABLE III
ROUTING PERFORMANCE FOR THE HFM SCENARIO

HFM scenario	Delivery rate	Overhead
BubbleRap	0.45 ± 0.029	453 ± 41
Epidemic	0.78 ± 0.024	2753 ± 142
Prophet	0.93 ± 0.029	1161 ± 38.9
GnR	0.82 ± 0.077	5.53 ± 0.65
ePRIVO	0.78 ± 0.017	4.97 ± 0.11

TABLE IV
ROUTING PERFORMANCE FOR THE HW SCENARIO

HW scenario	Delivery rate	Overhead
BubbleRap	0.38 ± 0.006	219 ± 8.2
Epidemic	0.31 ± 0.010	1311 ± 190
Prophet	0.45 ± 0.010	350 ± 48
GnR	0.58 ± 0.052	5.79 ± 0.10
ePRIVO	0.64 ± 0.009	5.47 ± 0.11

As seen in Table IV, the delivery rate of the protocols is reduced, which is due to the nodes having less contacts. This change emphasizes the best protocols to use. Epidemic and Prophet had a considerable drop on their metrics, whereas GnR, ePRIVO, and BubbleRap, which are social protocols, had a smaller drop on the delivery rate and had similar overheads.

GnR presented slightly worse results than ePRIVO. This is due to the fact that one of the main differences between these algorithms relies on the restrictions to transfer a message. GnR will only share the messages within its cluster, whereas ePRIVO shares them with any node with a greater chance to meet the destination.

In summary, GnR was developed to take advantage of the IoV architecture. It can be seen as an evolution of ePRIVO to this paradigm. It presents a similar delivery rate as the other standard routing protocols but with a much lower overhead ratio. In addition, it needs only 1.8 hops on average in order to deliver a message, whereas the others need 3.0, 4.7, and 3.8 for Bubblerap, Epidemic, and Prophet, respectively.

V. Conclusions and Future Work

There are some aspects to take into consideration when clustering an IoV network, such as (i) what are the clusters representing, (ii) the size of the clusters and the network, and (iii) the stability that the algorithm used will provide. Clustering based on social strength leads to stable results, which is advisable in the presence of mobile nodes with some movement patterns.

Our proposed routing protocol takes advantage of the edge layer and the location of the RSUs in order to comply with the IoV paradigm. It divides the network nodes into clusters that are based on their social relationships, which groups nodes that will most likely share the same connections.

Two metrics, which included similarity and ego betweenness centrality, were used for clustering nodes in IoV, and a higher importance is given to similarity when choosing the cluster head. The performance evaluation results showed that our proposed clustering, which is based on social relationships between the nodes, is an acceptable approach to this problem, because it provides a stable solution, which is due to the choice of the RSUs as anchors. It is also robust to an increment of the number of nodes. In addition, GnR performed better than the traditional routing protocols, which include Epidemic, Prophet, and BubbleRap, and it presented comparable results to ePRIVO, which presented itself as a good protocol for IoV. GnR had an overhead of about two orders of magnitude smaller than the traditional routing protocols with a comparable delivery rate for the HFM scenario and about twice the delivery rate for the HW scenario.

The future research lines are being planned from the findings that are reported in this manuscript. The use of other clustering algorithms, such as X-Means [32], G-Means [33] or DBSCAN [34] will be explored as a replacement for K-Means, which estimates the number of clusters while the cluster formation is composed, such as without setting it beforehand. In addition, the RSU placement problem will be addressed as an optimization problem, which pursues finding their optimal location by maximizing the coverage and communication-related metrics.

Acknowledgment

The authors would like to thank Breno Sousa for the fruitful discussions in regards to the early conception of this research study.

References

[1] N. Magaia, G. Mastorakis, C. Mavromoustakis, E. Pallis, and E. K. Markakis, Eds., *Intelligent Technologies for Internet of Vehicles*, ser. Internet of Things. Cham: Springer International Publishing, 2021.

[2] L. Silva, N. Magaia, B. Sousa, A. Kobusińska, A. Casimiro, C. X. Mavromoustakis, G. Mastorakis, and V. H. C. de Albuquerque, “Computing paradigms in emerging vehicular environments: A review,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 3, pp. 491–511, 2021.

[3] N. Magaia, P. Gomes, L. Silva, B. Sousa, C. X. Mavromoustakis, and G. Mastorakis, “Development of mobile IoT solutions: approaches, architectures, and methodologies,” *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[4] K. Sha, R. Errabelly, W. Wei, T. A. Yang, and Z. Wang, “EdgeSec: Design of an edge layer security service to enhance IoT security,” *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pp. 81–88, 2017.

[5] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, and A. Mouzakitis, “A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021.

[6] J. Mills, J. Hu, and G. Min, “Communication-efficient federated learning for wireless edge intelligence in IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2020.

[7] L. Liu, C. Chen, T. Qiu, M. Zhang, S. Li, and B. Zhou, “A data dissemination scheme based on clustering and probabilistic broadcasting in VANETs,” *Vehicular Communications*, vol. 13, pp. 78–88, 2018.

[8] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim, and C. S. Hong, “Deep learning based caching for self-driving cars in multi-access edge computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2862–2877, 2021.

[9] W. Alasmary and W. Zhuang, “Mobility impact in IEEE 802.11p infrastructureless vehicular networks,” *Ad Hoc Networks*, vol. 10, no. 2, pp. 222–230, mar 2012.

[10] E. Karamad and F. Ashtiani, “A modified 802.11-based MAC scheme to assure fair access for vehicle-to-roadside communications,” *Computer Communications*, vol. 31, no. 12, pp. 2898–2906, jul 2008.

[11] S. Malik and P. K. Sahu, “A comparative study on routing protocols for VANETs,” *Heliyon*, vol. 5, p. e02340, 8 2019.

[12] P. Rani, N. Sharma, and P. K. Singh, “Performance comparison of VANET routing protocols,” in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, 2011, pp. 1–4.

[13] A. Bujari, O. Gaggi, C. E. Palazzi, and D. Ronzani, “Would current ad-hoc routing protocols be adequate for the internet of vehicles? a comparative study,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3683–3691, 2018.

[14] Y. H. Ho, A. H. Ho, and K. A. Hua, “Routing protocols for inter-vehicular networks: A comparative study in high-mobility and large obstacles environments,” *Computer Communications*, vol. 31, pp. 2767–2780, 7 2008.

[15] H. Zhou, B. Liu, T. H. Luan, F. Hou, L. Gui, Y. Li, Q. Yu, and X. Shen, “Chaincluster: Engineering a cooperative content distribution framework for highway vehicular communications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2644–2657, 2014.

[16] H. Fatemidokht, M. K. Rafsanjani, B. B. Gupta, and C.-H. Hsu, “Efficient and secure routing protocol based on artificial intelligence algorithms with uav-assisted for vehicular ad hoc networks in intelligent transportation systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4757–4769, 2021.

[17] F. Abbas and P. Fan, “Clustering-based reliable low-latency routing scheme using ACO method for vehicular networks,” *Vehicular Communications*, vol. 12, pp. 66–74, 4 2018.

[18] W. Qi, B. Landfeldt, Q. Song, L. Guo, and A. Jamalipour, “Traffic differentiated clustering routing in DSRC and C-V2X hybrid vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7723–7734, 2020.

[19] H. Trinh, P. Calyam, D. Chemodanov, S. Yao, Q. Lei, F. Gao, and K. Palaniappan, “Energy-aware mobile edge computing and routing for low-latency visual data processing,” *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2562–2577, 2018.

[20] K. Lin, C. Li, Y. Li, C. Savaglio, and G. Fortino, “Distributed learning for vehicle routing decision in software defined internet of vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3730–3741, 2021.

[21] Z. Zhou, A. Gaurav, B. B. Gupta, M. D. Lytras, and I. Razzak, “A fine-grained access control and security approach for intelligent vehicular transport in 6g communication system,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.

[22] W. Qi, Q. Song, X. Wang, L. Guo, and Z. Ning, “SDN-enabled social-aware clustering in 5G-VANET systems,” *IEEE Access*, vol. 6, pp. 28 213–28 224, 2018.

[23] A. Paranjothi, M. S. Khan, M. Nijim, and R. Challoo, “MAvanet: Message authentication in vanet using social networks,” in *2016 IEEE*

7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), 2016, pp. 1–8.

[24] S. R. Thangiah and S. Salhi, "Genetic clustering: An adaptive heuristic for the multidepot vehicle routing problem," *Applied Artificial Intelligence*, vol. 15, no. 4, pp. 361–383, apr 2001. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/08839510151087293

[25] Ö. Kayiş and T. Acarman, "Clustering formation for inter-vehicle communication," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2007, pp. 636–641.

[26] M. Bouet and V. Conan, "Mobile edge computing resources optimization: A geo-clustering approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 787–796, 2018.

[27] T. Han, L. Zhang, S. Pirbhulal, W. Wu, and V. H. C. de Albuquerque, "A novel cluster head selection technique for edge-computing based IoMT systems," *Computer Networks*, vol. 158, pp. 114–122, jul 2019.

[28] N. Magaia and Z. Sheng, "ReFIoV: A novel reputation framework for information-centric vehicular applications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1810–1823, 2019.

[29] N. Magaia, C. Borrego, P. R. Pereira, and M. Correia, "ePRIVO: An enhanced PRIvacy-preserVing Opportunistic routing protocol for vehicular delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 154–11 168, 2018.

[30] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. New York, NY, USA: ICST, 2009.

[31] D. B. A. Vahdat, "Epidemic routing for partially connected ad hoc networks," *Technical Report CS-200006, Duke University*, 2000.

[32] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Service Assurance with Partial and Intermittent Resources*, P. Dini, P. Lorenz, and J. N. de Souza, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 239–254.

[33] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.