

Article

Not peer-reviewed version

---

# Adaptive Load Balancing Approach to Mitigate Network Congestion in VANETS

---

Syed Ehsan Haider , [Muhammad Faizan Khan](#)<sup>\*</sup> , Yousaf Saeed

Posted Date: 15 July 2024

doi: 10.20944/preprints202407.1012.v1

Keywords: Network Congestion; Load Balancing; VANET; Clustering



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Article

# Adaptive Load Balancing Approach to Mitigate Network Congestion in VANETS

Syed Ehsan Haider <sup>1,†</sup>, Muhammad Faizan Khan <sup>2,\*,†</sup>  and Yousaf Saeed <sup>1,†</sup>

<sup>1</sup> Department of Information Technology, The University of Haripur, Haripur, 22620, Pakistan

<sup>2</sup> School of Computer Science and Cyber Engineering, Guangzhou University, China

\* Correspondence: faizan@gzhu.edu.cn

† These authors contributed equally to this work.

**Abstract:** Load balancing to alleviate network congestion remains a critical challenge in Vehicular Ad-Hoc Networks (VANETs). During route and response scheduling, RSUs risk being overloaded beyond their calculated capacity. Despite recent advancements like RSU-based load transfer, NP-Hard hierarchical geography routing, RSU-based MAC schemes, simplified clustering, and network activity control, a significant gap persists in employing a load-balancing server for effective traffic management. We propose a server-based network congestion handling mechanism (SBNC) in VANETs to bridge this gap. Our approach clusters RSUs within specified ranges and incorporates dedicated load balancing and network scheduler RSUs to manage route selection and request-response scheduling, thereby balancing RSU loads. We introduce three key algorithms: optimal placement of dedicated RSUs, a scheduling policy for packets/data/requests/responses, and a congestion control algorithm for load balancing. Using the VanetMobiSim library of Network Simulator-2 (NS-2), we evaluate our approach based on residual energy consumption, end-to-end delay, packet delivery ratio (PDR), and control packet overhead. Results indicate substantial improvements in load balancing through our proposed server-based approach.

**Keywords:** network congestion; load balancing; vehicular Ad-hoc network; clustering

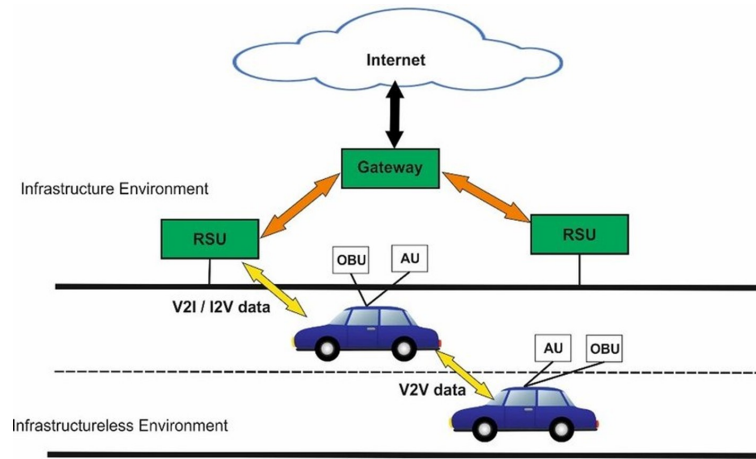
## 1. Introduction

In recent years, the development of mobile ad-hoc networks has resulted in notable progress in vehicular communication, particularly with the implementation of Vehicular Ad-Hoc Networks (VANETs) [1]. These networks provide various services, such as mitigating traffic congestion, broadcasting safety-related information, and optimizing travel routes for commuters [2]. VANETs are characterized by their dynamic topologies, fluctuating network sizes, and high mobility, and they are integral to intelligent transportation systems (ITS) [3]. Consisting of vehicles equipped with onboard units (OBUs) and roadside units (RSUs) that communicate with each other and the infrastructure, as illustrated in Figure 1, VANETs are built to enhance road safety and improve traffic efficiency by facilitating the exchange of information about road conditions, traffic congestion, accidents, and other critical data [4]. However, a primary challenge in VANETs is managing network congestion, which occurs when the data load on RSUs and OBUs surpasses their processing capacity [5].

In vehicular ad hoc networks (VANETs), network congestion stemming from data imbalance is a critical issue affecting user experience by causing slow response times. This imbalance results in certain Roadside Units (RSUs) and On-Board Units (OBUs) being overwhelmed with data requests. In contrast, others need to be more utilized, leading to inefficient use of computational resources and inconsistent Quality of Service (QoS) across the network [6,7]. These inefficiencies not only hinder VANET performance but can potentially compromise the delivery of critical information, impacting road safety and traffic management [8].

Current research in VANETs has focused on strategies such as RSU-based load transfer, hierarchical geography routing protocols, RSU-based MAC schemes, simplified clustering approaches, and network activity monitoring [9–19]. However, most studies prioritize reducing job reaction time or vehicle energy consumption overload balance, which is crucial for maintaining network performance and reliability [20]. Recent approaches, including hierarchical routing schemes with load balancing in

SD-VANETs [21] and proactive load balancing using parked vehicles in IoV-SDN [22], show promise in alleviating congestion by evenly distributing the load across the network and utilizing idle computational resources [23]. Nonetheless, a research gap still needs to be addressed to mitigate network congestion through load-balancing mechanisms effectively.



**Figure 1.** VANET Structure.[1]

Our research introduces an innovative approach to addressing network congestion in VANETs through a load-balancing server-based mechanism. The strategy involves organizing Roadside Units (RSUs) into clusters within specific ranges and integrating dedicated load balancing and network scheduler RSUs to effectively manage route selection and request-response scheduling. The primary focus of this approach is to establish a well-structured and efficient network by evenly distributing data loads to prevent any RSU from becoming a bottleneck. To achieve this, we aim to optimally position dedicated RSUs within the network to maximize coverage and efficiency, directly impacting the network's capacity to handle varying data loads and maintain seamless communication.

Furthermore, an essential aspect of our approach involves implementing a comprehensive policy for scheduling packets, data, requests, and responses. This policy determines how data is routed through the network, prioritizes critical data, and effectively manages traffic to minimize delays and prevent congestion. Finally, our research also emphasizes the development of a congestion control algorithm that continuously monitors data loads on each RSU. It dynamically adjusts routing and scheduling policies to prevent any RSU from becoming overwhelmed, ensuring network stability and optimal performance even under heavy traffic conditions.

The main research contributions are given below.

- We introduce three key algorithms. First, we devised a custom K-means clustering algorithm to optimize vehicle clusters and improve network organization. Second, we created an RSU-based Packets/Data/Request/Response Scheduling Policy algorithm to establish a robust scheduling framework driven by RSUs for efficient data prioritization. Lastly, a congestion control policy has been implemented using a load-balancing server to dynamically distribute load across clusters, paths, or vehicles to enhance network stability and performance.
- Our approach not only calculates and regulates the traffic load on specific paths within VANETs, but also brings significant benefits. It effectively manages data traffic along predetermined routes through the use of path computation techniques and predictive path tables, guaranteeing optimized data flow, reducing congestion, and improving network reliability.
- The research employs the VanetMobiSim simulator to establish a comprehensive simulation environment for VANETs. This environment enables the virtual implementation and evaluation of our proposed system. Additionally, the CANU Mobility Simulation Environment facilitates effective management of simulation parameters, addressing challenges such as request delay, load balancing, packet delivery ratio optimization, end-to-end delay minimization, and control

packet overhead management. This simulation framework validates the efficacy and performance of our proposed algorithms under various network conditions and scenarios.

The rest of the paper is organized as in section II; we define the current state-of-the-art literature study, which is based on selecting the proposed problem. In section III, we discuss the proposed system with three implemented algorithms. Section IV presents the system's proposed results that justify the evaluated parameters.

## 2. Related Work

An ad hoc network, or mobile ad hoc network (MANET), is a decentralized wireless communication network comprising individual wireless nodes capable of forming a network without any pre-existing infrastructure [1]. These networks are commonly used when traditional infrastructure-based networks are inaccessible or impractical, such as in military operations, disaster relief, search and rescue, and emergency response. Vehicular ad hoc networks (VANETs) are specialized ad hoc networks that focus on vehicle communication. VANETs have the potential to significantly impact transportation systems by enabling intelligent transportation solutions and laying the groundwork for future connected and autonomous vehicle technologies [2–4].

Vehicular Ad Hoc Networks (VANETs) face challenges due to intermittent vehicle connectivity. Roadside Units (RSUs) have shown promise in cooperative load balancing to manage data overload and outperform standalone simulation methods [24]. Efforts to optimize RSU use against limited transmission range have focused on cooperative load balancing strategies for efficient load redistribution and prompt responses to vehicle requests [23]. Research into formulating NP-hard problems for optimizing load balancing in VANETs considers constraints such as storage capacity, task completion time, transmit power, and energy consumption to develop solutions that can handle complex VANET dynamics while meeting stringent requirements [25,26].

Hierarchical geography routing protocols have been proposed to optimize routing efficiency in Software Defined Vehicular Networks (SDVNs) [10,21,27]. These protocols divide large geographic areas into smaller grids, enabling improved path selection based on real-time vehicle density and historical transfer probabilities between grids. Path cost functions with load balancing criteria further refine route selection, thereby enhancing network performance and reliability [12,28,29]. Additionally, in the domain of medium access control (MAC) strategies for VANETs, synchronous multi-channel MAC protocols coordinated by Road Side Units (RSUs) have been developed [9,19]. These protocols allocate service channels (SCHs) for data transmission, mitigating control channel congestion and improving overall network performance. Leveraging RSUs for information dissemination mitigates issues such as concealed terminals and optimizes data rates across the network [11,14].

Sealing cluster heads in VANETs is crucial for efficient network management and resource utilization. Various clustering algorithms and strategies have been researched to reduce control overhead, improve data propagation, and support targeted applications such as load balancing and traffic management [15,16]. Furthermore, research has assessed existing congestion control strategies in VANETs, emphasizing the significance of timely data dissemination and network stability during abnormal network conditions [17,18,21,30]. These studies highlight the ongoing endeavors to enhance VANET performance through innovative network protocols and management strategies.

Although significant advancements have been made, there is still a research gap that needs to be addressed in effectively managing network congestion through load-balancing mechanisms. Current studies often prioritize reducing job reaction time or vehicle energy consumption over achieving load balance, which is essential for maintaining network performance and reliability. Our research, which aims to fill this gap by proposing a novel load-balancing server-based mechanism to handle network congestion, has the potential to significantly improve load distribution and overall network efficiency in VANETs.

### 3. Proposed Methodology

We present our proposed methodology, as illustrated in Figure 2, which introduces an advanced load balancer-based architecture for managing network load in Vehicular Ad Hoc Networks (VANETs). This architecture integrates several crucial components, including servers, the internet, load balancers, Road Side Units (RSUs), and vehicular clusters. Servers connected to the internet manage data and requests, forwarding them to load balancers that efficiently distribute the network load among RSUs. Strategically positioned along roads, these RSUs manage communication with vehicles within their respective ranges, organizing vehicles into clusters to streamline data exchange.

Various communication paths, such as RSU to server, vehicle to RSU to server, RSU to cluster, cache sharing among vehicles, and content cache distribution within clusters, ensure efficient data routing, minimizing network congestion and enhancing data retrieval speed. Clusters share cached data to avoid redundant requests, while the RSUs, aided by the load balancers, manage vehicle groupings and maintain optimal network performance by balancing the load across the network. The overall traffic flow direction demonstrates how vehicles move through RSU coverage areas, ensuring robust and efficient vehicular communication, effectively managing load, and reducing latency.

The system starts with the initialization and status checking of RSUs, dynamically setting a range  $R$  and establishing a threshold value  $T$  to evaluate RSU parameters such as distance  $D$  and load capacity  $L$ . If an RSU lies within the predefined range ( $D \leq R$ ) and meets the threshold criteria ( $L \leq T$ ), it is integrated into the network. Otherwise, the system resets and reassesses, ensuring that only suitable RSUs are included to maintain optimal network coverage and performance as shown in Figure 3.

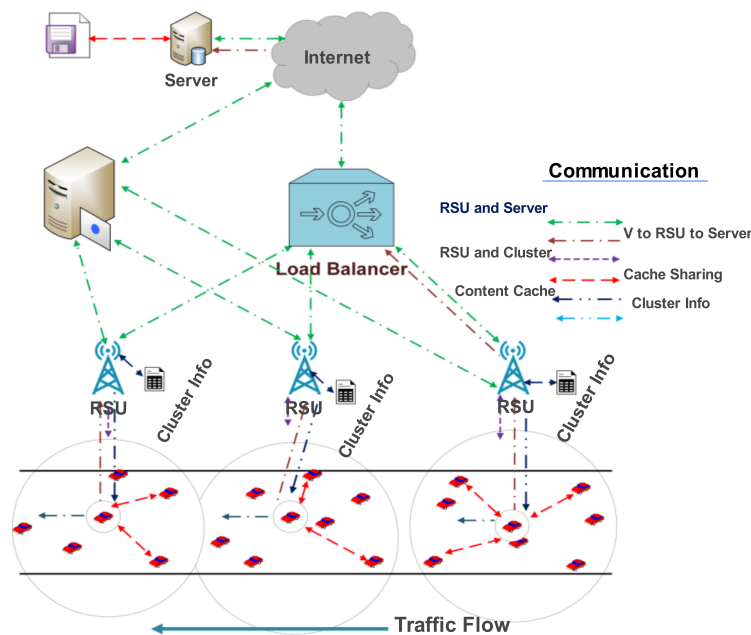


Figure 2. Proposed Load Balancing Architecture.



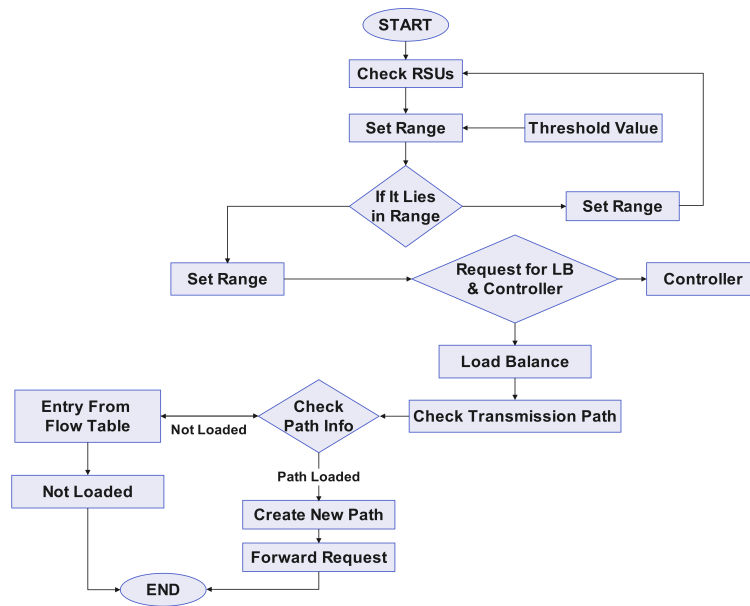


Figure 3. System Flow Diagram.

Our research introduces a task optimization and scheduling mechanism designed to address network congestion and load balancing within VANETs. The proposed load balancing mechanism is formulated as an optimization problem aimed at minimizing the maximum load on any RSU:

$$\min \max(L_1, L_2, \dots, L_n) \quad (1)$$

subject to

$$\sum_{i=1}^n L_i = L_{\text{total}}. \quad (2)$$

This architecture improves the operation of RSU-based servers and load balancers to function in parallel, effectively distributing the load according to the proposed network schemes. We propose a task scheduling load balancing-based initialization approach to mitigate load congestion. The Modified K-Mean Clustering algorithm is employed to balance the load by selecting nodes and forming clusters. Subsequently, an RSU-based packet/data/request/response scheduling policy algorithm ensures that devices do not exceed the threshold value, thereby maintaining network stability.

The system also incorporates a robust method for checking and optimizing transmission paths. If the path information is preloaded, the request is routed through the established path. Otherwise, a new path is created using entries in a flow table, leveraging graph theory to determine the shortest path in the network graph  $G = (V, E)$ . The optimization objective is to minimize the total path cost:

$$\min \sum_{(i,j) \in E} w_{ij} \quad (3)$$

where  $w_{ij}$  denotes the weight (cost, distance, or time) of the edge between vertices  $i$  and  $j$ . Additionally, RSUs are configured with a specified range and maximum capacity. When a vehicle enters an RSU's range, the RSU provides a transmission path if it can accommodate the request. If the RSU's capacity is exceeded, it seeks assistance from a load balancer and controller. If the path is absent from the flow table, a new path is generated and added. Requests from vehicles already in the flow table are forwarded if they remain within the RSU range. This comprehensive approach ensures efficient vehicular network performance, effectively managing load and congestion while enhancing overall network robustness.

**Algorithm 1** Clustering Algorithm by applying K-means

---

**Input:**  
 $L_{packets}^N$ : Number of packets received  
 $N: \{n_1, n_2, n_3, \dots, n_n\}$   
 $K$ : k-means  
 $C_1, C_2, \dots, C_n$ : Initial cluster centroids  
 $Q_{CH}$ : Centroid with the CH node  
 $P_C$ : Clusters

**Output:**  
Clusters with CH and nodes

```

1:  $N \leftarrow \text{random\_selection}(Q_{CH}, Q, K, L_{packets}^N)$ 
2: for  $j \leftarrow 1$  to  $J$  do
3:    $C_j \leftarrow N_j$ 
4:   while (criteria = threshold) do
5:     for  $i \leftarrow 1$  to  $J$  do
6:        $P_C \leftarrow \{\}$ 
7:       for  $i \leftarrow 1$  to  $I$  do
8:          $j \leftarrow \arg \min |C_j - N_m|$ 
9:          $P_C \leftarrow P_C \cup \{N_n\}$ 
10:      end for
11:    end for
12:    for  $j \leftarrow 1$  to  $J$  do
13:       $C_j \leftarrow \frac{1}{|P_C|} \sum_{P_C \in P_C} N_n$ 
14:    end for
15:  end while
16: end for return  $C_j \leftarrow \{C_1, C_2, \dots, C_n\}$ 

```

---

**3.1. Clustering Transformation**

K-Means Clustering is a widely acclaimed algorithm applied in diverse fields like Sensors, Data Mining, Ad-Hoc Networks, and IoT-based Networks. As a supervised learning technique, it initially categorizes vehicular nodes into K clusters[31]. Our adaptation of K-Means for VANET integrates RSUs as primary selectors and incorporates distance metrics into our proposed framework. The initial random selection of clusters, based on K-values, is followed by centroid determination from vehicles, considering factors such as distance from RSUs, previous packet request rates, speed, and location. Our approach emphasizes dynamic centroid positioning, facilitating efficient node enlistment and cluster head selection. Algorithm 1 delineates the Modified K-Means Clustering algorithm for our proposed framework, which empowers clusters to effectively manage network load balancing.

In Algorithm 1, the key steps involve iteratively updating the cluster centroids  $C_j$  based on the mean of their assigned points  $P_C$ . This update process continues until a certain criterion, represented by a threshold, is met. This thresholding criterion ensures convergence of the algorithm and stability of the resulting clusters, crucial for network performance.

$$C_j \leftarrow \frac{1}{|P_C|} \sum_{n \in P_C} n \quad (4)$$

where  $P_C$  represents the set of points assigned to cluster  $C_j$ , and  $n$  denotes individual data points.

Additionally, the nearest centroid selection process employs the argmin operation, seeking the closest centroid to each data point:

$$j \leftarrow \arg \min |C_j - N_m| \quad (5)$$

where  $N_m$  represents the  $m$ -th data point and  $C_j$  denotes the  $j$ -th centroid.

**3.2. RSU Based Packets/Data/Request/Response Scheduling Policy**

After constructing the clusters using the K-Mean clustering algorithm, redundant nodes are removed from the network. Each node is incorporated into a cluster, ensuring efficient organization. When a node exits the coverage area of a specific RSU cluster, it joins a new cluster based on the defined range for cluster transition. This dynamic reassignment helps maintain optimal network performance. The RSU-based scheduling policy is designed to balance network load and minimize congestion in packet/data transmission for both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication. The algorithm 2 employs a multi-layer conflict graph architecture in two key steps: Priority Assignment and Control and Services Channels.

**Algorithm 2** RSU based packets/data/request/response scheduling policy algorithm

---

**Require:** Channel, Priority of messages, Vehicles, CH, RSU, Messages, Scheduling type, Transmission  
**Ensure:** Scheduling Policy

```

1: for (1 to N) do
2:   setupnetwork(load-threshold)
3:   Setup - Channels()
4:   control(static, dynamic)
5:   services(Static, dynamic)
6:   prioritydecision(dynamic, static)  $\rightarrow \frac{\text{dynamic}(\text{factor}) \times \text{static message}}{\text{message\_size}}$ 
7:   Select Cluster: Algorithm-1
8:   if (static = message)
9:     set High = priority(emergency, high, contro - vanet)
10:  else
11:    for (message(1  $\rightarrow$  N) do
12:      scan(message = queues)
13:      if (message - load  $\geq$  threshold)
14:        switch(control, services, load)
15:      End for
16:    End If
17:  End For
18: End If
19: transmitmessage  $\rightarrow$  vehicle( $C_1^V, C_2^V, C_3^V, \dots, C_{n+1}^V$ )
20: Number of neighbors of vehicle:  $N_A \rightarrow \frac{2A(n+1)}{i}$ 
21: Bandwidth Utilization:  $Effective_{\text{Bandwidth}} \xrightarrow{i}$ 
22: if  $z_T \rightarrow 1$ 
23:    $z_T \rightarrow 0$ 
24: End if
25: End if
26: Return ( $z_T$ )

```

---

To achieve effective load balancing and congestion control, the proposed methodology integrates a dynamic message transmission policy. This policy operates at the application layer of the network, managing control signals and facilitating cooperative network policy management. Scheduling is prioritized based on message urgency, utilizing sender applications and neighboring contexts to ensure efficient and timely message transmission.

### 3.2.1. Priority Assignment

In this approach, packets are assigned priority by RSUs, which act as message generators for every vehicle in the network. The transmission time for each packet is considered relative to its priority for transmission. The relative transmission time for each packet is adjusted as the density of the network increases. High-priority packets are transmitted earlier, while medium- or lower-priority packets are delayed and sent according to network needs. Based on priority, two main methods are used to deploy the system's priority:

1. **Static Method:** Enhances the system version implemented within a specified range. The static field for packet priority is determined to analyze the composed mechanism.
2. **Dynamic Method:** Obtains specific content in the network. The controller determines both static and dynamic fields for priority assignment within the congestion control mechanism. These fields are combined to set up the overall comparison of priority indicators.

### 3.2.2. Control and Services Channels

In Vehicular Ad-Hoc Networks (VANETs), two main considerations are adopted for control and services channels, both used for V2V and V2I infrastructure communication. When the number of messages is smaller than control messages, scheduling policies are implemented for control signals. Conversely, when control messages are fewer than service messages, a new message sending and receiving policy is adopted to handle the number of packets sent through the proposed approach. If the service channel is overloaded with messages and the control channel is free, service queues are switched to control queues. This switch is only considered for high-priority messages related to control signals, providing a comprehensive overview of the required control signals.

A threshold is set for both service messages and control queues. The service channel is considered overloaded when the number of messages exceeds a particular threshold value, referred to as the "Threshold for Services Channel."



1. **Priority Decision:** The priority of a message is determined by combining dynamic and static factors:

$$\text{priority}_{\text{decision}}(\text{dynamic}, \text{static}) = \frac{\text{dynamic\_factor} \times \text{static\_message}}{\text{message\_size}} \quad (6)$$

2. **Number of Neighbors:** The number of neighbors of a vehicle,  $N_A$ , is calculated as:

$$N_A = \frac{2a(n+1)}{i} \quad (7)$$

where  $a$  and  $i$  are constants related to the network configuration, and  $n$  is the number of vehicles.

3. **Bandwidth Utilization:** Effective bandwidth utilization is considered and if a certain condition is met,  $z_T$  is updated:

$$z_T = 1 \Rightarrow z_T = 0 \quad (8)$$

### 3.3. Balances the VANET Congestion Control Policy

Load balancing is a crucial policy adopted to distribute the workload across RSUs and vehicles, managed by the Load Balancer and Controller components. The *Load Balancer* initiates by checking for pending requests ( $Req_i \neq \emptyset$ ). For each request, it computes a value  $C$  using parameters such as time, request access rate, speed,  $C_i$  (current load or capacity) and  $V_{id}$  (vehicle identification):

$$C = \text{compute}(\text{time}, \text{request access rate}, \text{speed}, C_i, V_{id}) \quad (9)$$

If  $C$  exceeds a predefined threshold, the request packet undergoes scheduling using a designated algorithm policy. Otherwise, the load balancer evaluates the load using a balancing formula and prioritizes the request using an alternative algorithm (Algorithm 2), offering alternate service paths for the request. This approach, as illustrated in Figure 3 of the proposed model, incorporates higher-order rates to optimize load management among multiple nodes, thereby enhancing the learning capabilities of load management methodologies.

The *Controller* procedure verifies the validity of transmission paths and consults the flow table for path information. In cases where a path is heavily loaded (indicating significant request or control signal handling), a new path is created:

$$\text{create}_{\text{new}}(\text{path}) \quad (10)$$

Subsequently, the controller directs the request through this new path. If the path remains underloaded, both paths are monitored, and control signals are dispatched accordingly. This dual-path verification and signal transmission mechanism bolster network robustness and reliability, ensuring efficient request routing without overwhelming any single path. Algorithm 3 outlines the congestion control policy, delineating how the load balancer and controller manage incoming loads from vehicles to RSUs based on content priority and data packet transmission. The algorithm's dynamic and adaptive nature underscores its efficacy in traffic management and load balancing.

## 4. Experimental Results

### 4.1. Simulation Setup

We implemented our proposed technique using the VanetMobiSim simulator, an extension of the NS-2 simulator environment[32]. This simulator was chosen for its capability to model mobility in vehicular networks (VANETs) with features such as speed-based shortest path discovery, path detail verification, load computation across paths, and dynamic street segment configuration. VanetMobiSim operates within the CANU Mobility Simulation Environment, known for its flexibility in mobility modeling and simulation. Our methodology and algorithms were evaluated using VanetMobiSim to validate our system implementation plan. We conducted 60 independent simulations to ensure

**Algorithm 3** Load Balancing Strategy

**Require:** Controller, Balancer, Algorithm 1, Algorithm 2, Packets, RSU, Vehicles, Links, Time, Transmission, Priority, Control Signals.

**Ensure:** Load Balancer

```
1: procedure LOAD_BALANCER
2:   while ( $Req_i \neq \emptyset$ ) do
3:      $C = \text{compute}(\text{time, request access rate, speed, } C_i, V_i d)$  for  $OP_{req}^n$ 
4:     if ( $C == \text{threshold}$ ) for  $Req_{packet}$  then
5:       schedule using Scheduling algorithm policy
6:     else
7:       (Load Balancer) computes
8:       corresponding load(.) using load balancing formula
9:       use priority schedule the request (based on algorithm 2)
10:      provide alternate services path for request
11:    end if
12:  end while
13: end procedure
14: procedure CONTROLLER
15:   if ( $\text{Transmission}_{path} - \text{valid}$ ) then
16:     check path info
17:     entry from flow table()
18:     if (path is loaded == more request of services or control are handled) then
19:       createnew(path)
20:       forward(request)
21:     else
22:       (check both paths)
23:       sendsignals()
24:     end if
25:   end if
26: end procedure
```

robustness and reliability of the proposed approach, utilizing a computer system equipped with an Intel(R) Core(TM) i5 2.3 GHz 10th generation CPU, 14 GB of RAM, and running Windows 10 OS. Simulation configurations and results are detailed in Table 1, which presents the setup specifics and outcomes of our proposed scenario.

**Table 1.** Simulation Parameters

Parameter	Values
Popularity Model	Zipf
Simulator	VanetMobiSim simulator
Environment	Network Simulator (NS-2)
Extension	CANU Mobility Simulation Environment
Simulation Area	250 x 250
Wireless Connectivity	Wi-Fi
Publisher/RSUs	10
Subscribers/Vehicles	100
Cache Size	100
File Size	Chunks on Demand Basis
Message Transformation	15-18 Sec
Mobility Model	2D-Random Directions
Number of Simulations Run	60
Simulation Time	1000s
Road Conditions	One Way Highway
Broadcast Interval Time	50s

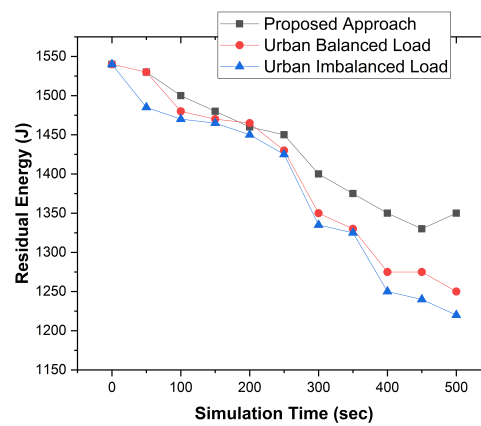
4.2. Simulation Parameters

The simulation results in this study effectively consider the identified parameters within an urban vehicular scenario. In urban areas, vehicles typically travel at speeds ranging from 50 to 60 km/h. Due to traffic signals and congestion, vehicle speeds can drop to zero, resulting in a stopped scenario. The proposed approach leverages vehicles connected to a Load Balancing server via RSUs. The mobility factor is integral to the connection and decision-making processes for vehicular nodes. The main parameters considered in the simulations are Residual Energy consumption, End-to-End Delay, Packet Delivery Ratio (PDR), and Control Packet Overhead. These results are benchmarked against those from [14,33], facilitating a comparison between imbalanced and balanced simulation parameters.

### 4.3. Simulation Results

#### 4.3.1. Residual Energy Consumption

Initially, we evaluated the nodes' energy consumption throughout the entire duration, taking into account the nodes' mobility factors. In Figure 4, the energy utilization of the proposed approach is compared with existing methods. Over time, the energy consumption of the proposed approach is expected to decrease, but it will be limited compared to existing approaches. The efficient load distribution algorithm in this scenario contributes to the proposed approach's energy consumption reduction. Network resources are optimized to manage the energy usage of the nodes effectively. The proposed approach demonstrates a decrease in energy consumption from zero to 500 ms over time. Energy is measured in Joules, the standard unit of energy. Initially, energy consumption started at zero and decreased over time. Consequently, the suggested load balancer-based VANET approach exhibits lower residual energy than cluster-based and imbalanced load-handling schemes. This increase in residual energy improves network response time and maximizes the battery life for vehicular nodes in urban scenarios, making battery life a crucial factor in the proposed scenario.

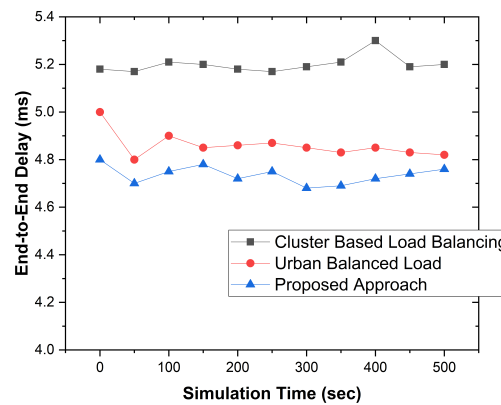


**Figure 4.** Residual of Energy Consumption of the Proposed Technique compared with existing approaches.

#### 4.3.2. End-to-End Delay

End-to-end delays can impact communication between vehicles and roadside units (RSUs). This parameter, as shown in Figure 5, defines the transmission delay and packet forwarding ratios. The end-to-end delay parameter also plays a crucial role in identifying underutilized nodes, as it illustrates the strain on the chosen vehicular nodes. In scenarios of cluster-based and urban imbalanced load distribution, the outcomes reveal an overload on certain nodes and offer a comprehensive approach to load distribution. When the load on nodes increases, some nodes become imbalanced and underutilized. Increased processing time leads to greater congestion in the network. Consequently, congestion raises the response time of nodes and can result in missed job deadlines.

The proposed approach shows a lower delay when the load on the nodes increases over time. The increased load values enhance the working principle for nodes. The proposed approach effectively balanced the load from one part of the network towards the other part with the distribution of the network load to idle vehicles. However, the load balancer plays a significant role in this process, as it significantly reduces the end-to-end delay in the packet sending among the vehicular nodes. This reduction in delay is a crucial benefit of our approach. Reducing congestion in the network may enhance network distribution and congestion handling. As congestion and end-to-end delay are directly proportional, the end-to-end delay decreases inside the network once we minimize congestion.



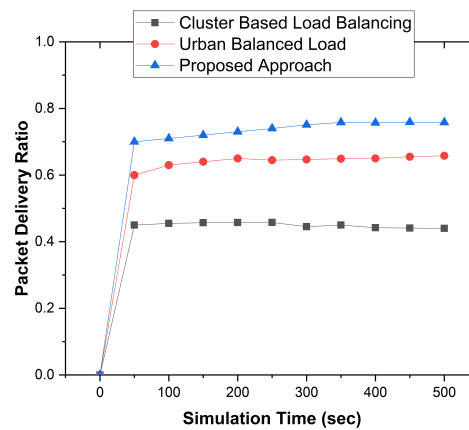
**Figure 5.** End-to-End Delay in the Network compared with the proposed approach.

The proposed load balancer distributes the load towards idle nodes over the network, utilizing the network's capabilities with load distribution. The job completed in one node reduces the job enhancements to other nodes with capabilities to reduce network congestion. Due to this dependency, the proposed approach reduces the node's response time with an additional balanced load over the network.

#### 4.3.3. Packet Delivery Ratio (PDR)

The packet delivery ratio is another property that can handle network congestion in the urban VANET environment. The packet sends and receives ratio is the packet delivery ratio with an improved load balancing server to balance the number of requests sent and received back from the system. The proposed load-balancing approach is compared with the existing approach with a practical load-balancing scenario in a vehicular environment. Figure 6 shows the PDR results.

The PDR of the proposed load-balancing server approach and the existing state-of-the-art techniques are compared. The load distribution was initially the same because it was closer to the gateway server. After an initial time, the load distribution and practical work handling parameters utilize the load balancing capacities for the network scenario. The fog nodes are used in the proposed scheme according to the capacity made. The cluster-based routing approach shows lower packet delivery due to network congestion needing to be appropriately handled in that proposed approach. In urban-based load handling, the load is distributed but not effectively managed, and control is not effectively utilized. On the other hand, in our proposed approach, cluster-based load balancing and handling is another scenario that aims to utilize network effectiveness and provide effective load balancing. In the proposed approach, the PDR value for the network is reduced, which indicates the values presented in the proposed scenario.



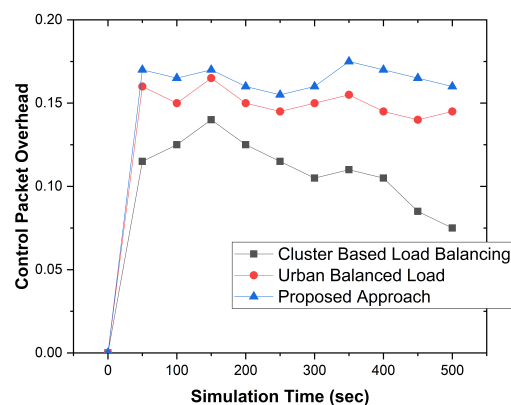
**Figure 6.** Packet Delivery Ration (PDR) for proposed load balancing technique in Urban Area VANET.

The proposed load-balancing approach effectively balances the load among clusters and enhances network throughput. The cluster-based transformation is based on the network scenario and provides effective load-balancing strategies.

#### 4.3.4. Control Packet Overhead

The control packet overhead scenario shows the delay in sending packets in a cluster-based proposed load-balancing approach. Overhead uses time and overhead values to utilize the network performance and scenario. Figure 7 shows the control packet overhead in the proposed scenario to control the network's overall performance. The control packet overhead of the proposed approach is higher than that of the existing Cluster-Based Load Balancing and Urban-balanced Load approaches.

The overhead is caused in the proposed network due to velocity, information contained in messages for vehicles, direction, current position, and proposed network system delay. Initially, control packet overhead involves the overhead time.



**Figure 7.** Control Packet Overhead in Urban Vehicular Area Network to control the traffic congestion and load balancing.

With time, the proposed technique shows higher packet control packet overhead in the system development. The resources are efficiently utilized, and control packets enhance the load-balancing approach by effectively utilizing requests. The control packet overhead increases due to handling more requests from internodes behavior.



## 5. Performance Analysis

In Vehicular Ad-Hoc Networks (VANETs), tasks are frequently offloaded from vehicles to RSUs for processing, leading to network congestion [5]. While traditional scheduling policies have aimed to tackle these issues, there is a need for better congestion management and resource utilization optimization [6]. Our study presents an Adaptive Load Balancing Approach (SBNC) to alleviate network congestion and enhance load balancing in VANETs [26]. This method involves a centralized load-balancing server integrated with RSUs and base stations to optimize network load distribution dynamically. Our approach boosts network efficiency and traffic flow control in urban vehicular environments by employing advanced scheduling algorithms and dynamic message transmission policies. This leads to notable reductions in energy consumption, end-to-end delay, and improved packet delivery ratios between nodes [20].

In contrast to the cooperative load balancing methods reviewed in [15,24,31], which distribute load across multiple vehicles and RSUs, our approach utilizes a centralized load balancing server to manage network load efficiently. This ensures a more effective and responsive congestion control mechanism. Our methodology, as compared to the hierarchical load balancing approach [12,27], employs a unified server-based system that optimizes both V2V and V2I communications, resulting in reduced latency and improved traffic flow. While energy-efficient strategies, highlighted in [25,31], primarily focus on extending network lifespan, our approach goes beyond by integrating advanced scheduling algorithms and dynamic message transmission policies, significantly reducing energy usage while maintaining high performance across other metrics. The heterogeneous network strategies discussed by Lyamin et al. [5] involve managing the load through different types of RSUs. In contrast, our centralized load balancing server allows for real-time adjustments and more precise management of network resources, effectively handling peak traffic conditions.

In addition, the congestion control mechanisms reviewed by Al Mallah et al. [6,7] offer fundamental insights into managing network traffic to prevent congestion. Expanding on these principles, our SBNC mechanism introduces innovative clustering techniques for RSUs and advanced scheduling algorithms, resulting in significant improvements in load balancing and congestion management. Compared to the distributed multi-path computation algorithms proposed by Zhang et al. [26], our centralized control approach reduces complexity and ensures more reliable load distribution. Finally, while Dai et al. [8] investigate the use of machine learning for intelligent load balancing, our deterministic and structured method offers a robust solution suitable for real-world implementation. By incorporating a dedicated load-balancing server and sophisticated scheduling policies, our approach establishes a new standard for future research and practical applications in intelligent transportation systems and vehicular communication technologies.

### 5.1. Limitations of the Research

This study primarily focuses on congestion control in VANETs using vehicle-centric mechanisms without addressing direct congestion mitigation strategies at RSUs. The scope of the study is limited by time constraints, which may affect the depth of analysis and experimentation. While the research suggests approaches to reduce energy consumption during congestion, it should also thoroughly investigate other essential factors, such as bandwidth utilization and data transmission efficiency. This comprehensive exploration is crucial to ensure the robustness and effectiveness of the proposed strategies. Additionally, the study does not consider variations in network traffic density or environmental conditions that could impact the effectiveness of the proposed congestion control mechanisms. Furthermore, the evaluation is based on simulations, which may only partially replicate real-world complexities and performance metrics. These limitations indicate future research opportunities to expand the scope and strengthen the effectiveness of congestion management strategies in VANETs.

### 5.1.1. Impact

This study makes a significant contribution to VANETs by introducing a load-balancing server-based network congestion (SBNC) handling mechanism. The innovative approach involves clustering RSUs and integrating dedicated load balancing and network scheduler RSUs to address the critical issue of RSU overload during route and response scheduling. By optimizing the placement of RSUs and implementing advanced scheduling and congestion control algorithms, the proposed methodology enhances network efficiency, reduces end-to-end delay, and improves the packet delivery ratio (PDR). The use of the VanetMobiSim library within the Network Simulator-2 (NS-2) environment for evaluation provides a robust framework for validating the effectiveness of the proposed solutions. This study demonstrates a marked improvement in load balancing and energy consumption and sets a precedent for future research to explore comprehensive congestion control mechanisms encompassing vehicle and RSU-centric strategies. The findings have the potential to inspire future research and influence the design and deployment of more resilient and efficient VANET infrastructures, ultimately contributing to the advancement of intelligent transportation systems and vehicular communication technologies.

## 6. Conclusion and Future Directions

This research identifies load balancing as a critical challenge in Vehicular Ad-Hoc Networks (VANETs), significantly impacting network performance and congestion management. The proposed approach addresses this issue by implementing a comprehensive load-balancing strategy involving the formation of clusters, an RSU-based packets/data/request/response scheduling policy, and a robust VANET Congestion Control Policy. The effectiveness of these techniques is demonstrated through simulations using the VanetMobiSim Library of Network Simulator-2 (NS-2). Key performance metrics, including Residual Energy Consumption, End-to-End Delay, Packet Delivery Ratio (PDR), and Control Packet Overhead, show substantial improvements, validating the proposed approach.

Future work will focus on further improvements, such as adopting deep learning-based pre-caching content prediction to improve load balancing and incorporate security aspects to address potential vulnerabilities.

## References

1. Toor, Y.; Muhlethaler, P.; Laouiti, A.; La Fortelle, A.D. Vehicle Ad Hoc networks: applications and related technical issues. *IEEE Communications Surveys & Tutorials* **2008**, *10*, 74–88. <https://doi.org/10.1109/COMST.2008.4625806>.
2. Balico, L.N.; Loureiro, A.A.F.; Nakamura, E.F.; Barreto, R.S.; Pazzi, R.W.; Oliveira, H.A.B.F. Localization Prediction in Vehicular Ad Hoc Networks. *IEEE Communications Surveys & Tutorials* **2018**, *20*, 2784–2803. <https://doi.org/10.1109/COMST.2018.2841901>.
3. Ullah, A.; Yao, X.; Shaheen, S.; Ning, H. Advances in Position Based Routing Towards ITS Enabled FoG-Oriented VANET—A Survey. *IEEE Transactions on Intelligent Transportation Systems* **2020**, *21*, 828–840. <https://doi.org/10.1109/TITS.2019.2893067>.
4. Rettore, P.H.; Maia, G.; Villas, L.A.; Loureiro, A.A.F. Vehicular Data Space: The Data Point of View. *IEEE Communications Surveys & Tutorials* **2019**, *21*, 2392–2418. <https://doi.org/10.1109/COMST.2019.2911906>.
5. Lyamin, N.; Bellalta, B.; Vinel, A. Age-of-Information-Aware Decentralized Congestion Control in VANETs. *IEEE Networking Letters* **2020**, *2*, 33–37. <https://doi.org/10.1109/LNET.2020.2970695>.
6. Al Mallah, R.; Quintero, A.; Farooq, B. Distributed Classification of Urban Congestion Using VANET. *IEEE Transactions on Intelligent Transportation Systems* **2017**, *18*, 2435–2442. <https://doi.org/10.1109/TITS.2016.2641903>.
7. Nourani, N.; Seno, S.A.H.; Etehadi, S.S.S.; Hosseinpour, M.; Yaghmaee, M.H. Distributed Traffic Congestion Classification in Intelligent Transportation Systems based on SDN and Fog Computing. In Proceedings of the 2023 7th International Conference on Internet of Things and Applications (IoT), 2023, pp. 1–6. <https://doi.org/10.1109/IoT60973.2023.10365360>.

8. Dai, Y.; Xu, D.; Maharjan, S.; Zhang, Y. Joint Load Balancing and Offloading in Vehicular Edge Computing and Networks. *IEEE Internet of Things Journal* **2019**, *6*, 4377–4387. <https://doi.org/10.1109/JIOT.2018.2876298>.
9. Sahoo, S.R.; Patra, M.; Gupta, A. AALB: Application Aware Load Balancing Algorithm for Road Side Units. *Vehicular Communications* **2022**, *36*, 100475. <https://doi.org/10.1016/j.vehcom.2022.100475>.
10. Bitam, S.; Mellouk, A.; Zeadally, S. Bio-Inspired Routing Algorithms Survey for Vehicular Ad Hoc Networks. *IEEE Communications Surveys & Tutorials* **2015**, *17*, 843–867. <https://doi.org/10.1109/COMST.2014.2371828>.
11. Thapa, S.; Sahoo, S.R.; Patra, M.; Gupta, A. A Novel Cost-Aware Load Balancing Algorithm for Road Side Units in Internet of Vehicles. In Proceedings of the 2022 18th International Conference on Network and Service Management (CNSM), 2022, pp. 359–363. <https://doi.org/10.23919/CNSM55787.2022.9964580>.
12. Wang, M.I.C.; Wen, C.H.P.; Chao, H.J. Hierarchical Cooperation and Load Balancing for Scalable Autonomous Vehicle Routing in Multi-Access Edge Computing Environment. *IEEE Transactions on Vehicular Technology* **2023**, *72*, 6959–6971. <https://doi.org/10.1109/TVT.2023.3236783>.
13. Song, Y.; Jiang, K.; Cao, Y.; Zhou, R.; Suthaputchakun, C.; Zhuang, Y. STALB: A Spatio-Temporal Domain Autonomous Load Balancing Routing Protocol. *IEEE Transactions on Network and Service Management* **2023**, *20*, 73–87. <https://doi.org/10.1109/TNSM.2022.3208025>.
14. Saemi, B.; Halataei, F.; Ahmadi, R.; Ashkaran, A.; Mirkamali, S.; Hosseinabadi, A.A.R. Improving the quality of real-time data transmission service in VANETS by balancing the load on road side units. *Cluster Comput.* **2024**.
15. Cooper, C.; Franklin, D.; Ros, M.; Safaei, F.; Abolhasan, M. A Comparative Survey of VANET Clustering Techniques. *IEEE Communications Surveys & Tutorials* **2017**, *19*, 657–681. <https://doi.org/10.1109/COMST.2016.2611524>.
16. Kassir, S.; de Veciana, G.; Wang, N.; Wang, X.; Palacharla, P. Analysis of Opportunistic Relaying and Load Balancing Gains Through V2V Clustering. *IEEE Transactions on Vehicular Technology* **2022**, *71*, 9896–9911. <https://doi.org/10.1109/TVT.2022.3178129>.
17. Driss Laanaoui, M.; Lachgar, M.; Mohamed, H.; Hamid, H.; Gracia Villar, S.; Ashraf, I. Enhancing Urban Traffic Management Through Real-Time Anomaly Detection and Load Balancing. *IEEE Access* **2024**, *12*, 63683–63700. <https://doi.org/10.1109/ACCESS.2024.3393981>.
18. Ahmed, U.; Lin, J.C.W.; Srivastava, G.; Yun, U.; Singh, A.K. Deep Active Learning Intrusion Detection and Load Balancing in Software-Defined Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 953–961. <https://doi.org/10.1109/TITS.2022.3166864>.
19. Haddad, M.; Muhlethaler, P.; Laouiti, A.; Zagrouba, R.; Saidane, L.A. TDMA-Based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis, and Open Research Issues. *IEEE Communications Surveys & Tutorials* **2015**, *17*, 2461–2492. <https://doi.org/10.1109/COMST.2015.2440374>.
20. Wu, Y.; Wu, J.; Chen, L.; Yan, J.; Han, Y. Load Balance Guaranteed Vehicle-to-Vehicle Computation Offloading for Min-Max Fairness in VANETs. *IEEE Transactions on Intelligent Transportation Systems* **2022**, *23*, 11994–12013. <https://doi.org/10.1109/TITS.2021.3109154>.
21. Malakar, M.; Mahapatro, J.; Ghosh, T. A survey on routing and load-balancing mechanisms in software-defined vehicular networks. *Wirel. Netw.* **2024**, *30*, 3181–3197.
22. Kadhim, A.J.; Naser, J.I. Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN. *China Communications* **2021**, *18*, 271–289. <https://doi.org/10.23919/JCC.2021.02.019>.
23. Cao, D.; Wu, M.; Gu, N.; Sherratt, R.S.; Ghosh, U.; Sharma, P.K. Joint Optimization of Computation Offloading and Resource Allocation Considering Task Prioritization in ISAC-Assisted Vehicular Network. *IEEE Internet of Things Journal* **2024**, pp. 1–1. <https://doi.org/10.1109/JIOT.2024.3360962>.
24. Zhang, W.; Zhang, G.; Mao, S. Joint Parallel Offloading and Load Balancing for Cooperative-MEC Systems With Delay Constraints. *IEEE Transactions on Vehicular Technology* **2022**, *71*, 4249–4263. <https://doi.org/10.1109/TVT.2022.3143425>.
25. He, Y.; Wang, D.; Huang, F.; Zhang, R. An MEC-Enabled Framework for Task Offloading and Power Allocation in NOMA Enhanced ABS-Assisted VANETs. *IEEE Communications Letters* **2022**, *26*, 1353–1357. <https://doi.org/10.1109/LCOMM.2022.3162603>.
26. Zhang, J.; Guo, H.; Liu, J.; Zhang, Y. Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution. *IEEE Transactions on Vehicular Technology* **2020**, *69*, 2092–2104. <https://doi.org/10.1109/TVT.2019.2959410>.

27. Gao, Y.; Zhang, Z.; Zhao, D.; Zhang, Y.; Luo, T. A Hierarchical Routing Scheme With Load Balancing in Software Defined Vehicular Ad Hoc Networks. *IEEE Access* **2018**, *6*, 73774–73785. <https://doi.org/10.1109/ACCESS.2018.2884708>.
28. Ortiz, M.T.; Salient, O.; Camps-Mur, D.; Escrig, J.; Nasreddine, J.; Pérez-Romero, J. On the Application of Q-learning for Mobility Load Balancing in Realistic Vehicular Scenarios. In Proceedings of the 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), 2023, pp. 1–7. <https://doi.org/10.1109/VTC2023-Spring57618.2023.10201091>.
29. Naderi, M.; Ghanbari, M. Adaptively prioritizing candidate forwarding set in opportunistic routing in VANETs. *Ad Hoc Networks* **2023**, *140*, 103048. <https://doi.org/https://doi.org/10.1016/j.adhoc.2022.103048>.
30. Roy, C.; Misra, S.; Maiti, J.; Nag, S. EdgeSafe: Dynamic Load Balancing Among Edge Nodes for Provisioning Safety-as-a-Service in Vehicular IoT Applications. *IEEE Transactions on Vehicular Technology* **2021**, *70*, 9320–9329. <https://doi.org/10.1109/TVT.2021.3097557>.
31. Abdulrazzak, H.N.; Hock, G.C.; Mohamed Radzi, N.A.; Tan, N.M.L. A New Unsupervised Validation Index Model Suitable for Energy-Efficient Clustering Techniques in VANET. *IEEE Access* **2023**, *11*, 67540–67555. <https://doi.org/10.1109/ACCESS.2023.3281302>.
32. Chanal, P.M.; Kakkasageri, M.S.; Jahagirdar, R.G.; Sataraddi, M.J. Performance Analysis of Ant Colony based Routing Approach for VANETs using VanetMobiSim and NS2. In Proceedings of the 2019 11th International Conference on Advanced Computing (ICoAC), 2019, pp. 129–133. <https://doi.org/10.1109/ICoAC48765.2019.246828>.
33. Tripp-Barba, C.; Urquiza-Aguiar, L.; Igartua, M.A.; Rebollo-Monedero, D.; De la Cruz Llopis, L.J.; Mezher, A.M.; Aguilar-Calderón, J.A. A Multimetric, Map-Aware Routing Protocol for VANETs in Urban Areas. *Sensors* **2014**, *14*, 2199–2224. <https://doi.org/10.3390/s140202199>.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.