

Article

Not peer-reviewed version

Simulation System for V2X Applications

[Andrei Vladyko](#) , [Pavel Plotnikov](#) ^{*} , [Gleb Tambovtsev](#)

Posted Date: 9 October 2024

doi: 10.20944/preprints202410.0662.v1

Keywords: vehicular communications; vehicle-to-everything; vehicular computing; simulation system



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

Simulation System for V2X Applications

Andrei Vladyko ¹, Pavel Plotnikov ^{2,*} and Gleb Tambovtsev ¹

¹ Faculty of Radioelectronic Systems and Robotics, The Bonch-Bruевич Saint-Petersburg State University of Telecommunications, Prospekt Bolshevikov 22-1, 193232 Saint Petersburg, Russia

² Faculty of Information Technology and Software Engineering, The Bonch-Bruевич Saint-Petersburg State University of Telecommunications, Prospekt Bolshevikov 22-1, 193232 Saint Petersburg, Russia

* Correspondence: plotnikov.pv@sut.ru

Abstract: The issue of organizing efficient interaction between V2X system elements has become increasingly critical in recent years. Utilizing vehicle-to-everything (V2X) technology enables achieving the necessary balance of safety, reducing system load, and ensuring a high degree of vehicle automation. This study aims to develop a simulation system for V2X applications in various element placement configurations and conduct a numerical analysis of several V2X system interaction schemes. The research analyzes various methods, including clustering, edge computing, and fog computing, aimed at minimizing system losses. The results demonstrate that each proposed model can be effectively implemented on mobile nodes. Results also provide insights into the average expected request processing times, thereby enhancing V2X system organization. The authors propose a comprehensive model that enables the distribution and prediction of system parameters and resources for diverse computational tasks, essential for the successful implementation and utilization of V2X technology.

Keywords: vehicular communications; vehicle-to-everything; vehicular computing; simulation system

1. Introduction

In recent years, vehicle-to-everything (V2X) communication technology within intelligent transportation systems (ITS) has garnered significant attention due to its potential to address contemporary ITS requirements. V2X offers the potential to mitigate the strain on transportation network infrastructure, enhance road safety, improve driving comfort, provide an alternative emergency communication system during natural disasters, and enable a high degree of vehicle automation [1–3]. To achieve these objectives, V2X technology must ensure low latency, high reliability, energy efficiency, and optimal computational resource allocation. Furthermore, the specific characteristics of the data transmission network must be considered, including its highly dynamic nature, constant topology changes, and exchange of large data volumes. Failure to meet these requirements may prevent safety-critical applications from responding effectively in potentially hazardous situations. The development of a comprehensive V2X system management framework is also of considerable interest.

For the design of transportation systems with high automation levels (eyes-off and mind-off), vehicles must possess the capacity to process substantial real-time data related to nearby object recognition, road incidents, navigation tasks, etc. However, the computational resources of most existing vehicles are limited for executing such applications. Any disruption in system functionality and stability can jeopardize human safety and vehicular integrity, potentially leading to significant recovery and maintenance costs. Consequently, ensuring data transmission reliability and high ITS performance becomes paramount.

It should be noted that these challenges can be addressed through the implementation of clustering mechanisms during roadside infrastructure deployment, as well as the utilization of edge and fog computing. The impact of these mechanisms on the overall V2X system functionality is discussed extensively in the literature [4–7].

This paper proposes a mathematical model for a simulation system and its numerical solution for V2X applications in a traditional roadside unit (RSU) deployment configuration utilizing a two-channel connection between onboard units (OBU) and RSUs. Furthermore, an evaluation of the model's operational effectiveness will be conducted based on calculations of the percentage of unprocessed requests generated by OBUs during their mobility and the latency of request transmission and processing within the RSU-OBU system. A comparison of simulation results with and without mobile roadside units (mRSUs), with targeted and random movement routes, will be performed to assess their influence on the overall system serviceability.

Our contributions can be summarized as follows:

1. Analysis of a mathematical model for V2X system element interaction based on data loss during exchange and service interaction within various roadside unit deployment schemes under low and high mRSU map saturation intensities.

2. Development of an OBU-RSU interaction model within a classical mobility scheme and utilizing clustering to ensure efficient data transmission within a dynamic system. Simulation of mRSU (relay) flows operating on a taxi principle, with mRSU movement directed towards areas of peak load, was also conducted.

3. Presentation of a developed simulation system for the analysis and synthesis of V2X system applications. This system was utilized to determine the average time required for system stabilization concerning caching and losses. Conclusions regarding the applicability of approaches to describe efficient OBU-RSU interaction within the V2X system are formulated.

The remainder of the paper is structured as follows: Section 2 provides an overview of existing literature in the relevant subject area. Section 3 formulates a description of the system model and the applied mathematical framework. The description and structure of the proposed computer model, its general scheme, key numerical simulation parameters, primary challenges, and their solution methods are presented. A description of the simulation system's operation is also provided. Section 4 presents and describes the numerical simulation results aimed at validating the theoretical findings and their analysis in various system configurations. Finally, Section 5 concludes with final remarks and discussion.

2. Related Works

This section presents a concise overview of research within similar thematic areas, highlighting the key advantages and disadvantages of approaches developed by other authors.

Studies [8–11] emphasize the utilization of a cluster-based approach for organizing ITS element interaction. A subset of transport system flow management elements is grouped into blocks with shared caches, thereby significantly increasing system throughput and reducing message processing delays. Simulation systems such as SUMO, MATLAB, and the network simulator NS-3 are frequently employed for organizing cluster structures. The primary objectives where the cluster-based approach proves most effective include minimizing data transmission delays [8,9] and constructing a comprehensive resource management scheme [10,11].

An approach based on edge computing [12–17] is often employed to address challenges related to efficient offloading, system performance enhancement, and waiting time reduction. Vehicular Edge Computing (VEC) has emerged as a promising paradigm for service provisioning and management in V2X systems [18]. VEC facilitates low latency by bringing computational resources closer to the vehicles. However, the high dynamism of vehicular networks poses significant difficulties in designing optimal V2X service provision schemes while ensuring safety and timely service delivery. Studies [12,13] primarily focus on offloading and resource allocation to minimize the overall processing time for requests from vehicles to the infrastructure. The increasing intensity of computations and latency-sensitive onboard applications makes data processing directly on vehicles while maintaining the required computational capacity, and therefore performance, quite challenging. Nevertheless, offloading schemes where all vehicles offload their tasks to the same edge server can limit system performance. Consequently, several studies [14,15] propose and implement

the concept of load balancing through the introduction of multi-user, multi-service systems. Digital twin technologies [16] and distributed machine learning [17] are gaining increasing popularity.

Another effective method for addressing optimization issues in V2X systems involves a relatively new tool: Fog Computing (FOG). The concept of fog computing was introduced in 2011 due to the need to extend cloud computing capabilities [19]. Numerous ITS devices connect to the cloud simultaneously, often overwhelming the system. The core idea of FOG is to migrate a portion of computational tasks closer to the "ground." Fog elements would collect raw data (e.g., from sensors) and perform processing using their local computers instead of transmitting it to cloud servers [20–22]. An additional advantage is that the collected data is likely required by the same devices that generated it, so local processing minimizes the latency between input and response compared to remote processing.

The deployment of stationary RSUs is hindered by several factors, including high costs, difficulties in adapting to existing roadside infrastructure, and coordination issues between the public sector, telecom operators, and the automotive industry. It is important to highlight several studies addressing these challenges by utilizing vehicles as mRSUs [23]. This approach is scalable and adaptable to current road traffic. mRSU routes can be planned to traverse the most challenging road sections, and the presence of a route control system provides a significant advantage. The dynamic scaling of computational capacity at the roadside network infrastructure level allows for mRSUs to be considered as Fog elements.

Study [24] is one of the first to address mRSU deployment, proposing a self-organizing network approach where specific vehicles function as RSUs. Several studies focused on fog computing primarily address optimization problems such as maximizing network coverage [25,26], minimizing latency [27,28], minimizing data loss, and cost-effectiveness [29,30]. MATLAB, supplemented by tools from SUMO and OpenStreetMap, serves as the primary simulation environment. The dominant approaches for organizing V2X system element interaction within the fog computing paradigm are based on mechanisms such as greedy algorithms, self-organizing networks, and integer linear programming. Furthermore, public transport (PT) vehicles like buses and taxis with pre-installed communication and computational equipment can function as mRSUs. This mechanism is described in detail in studies [28,31].

In addition to the aforementioned research, hybrid mRSU deployment approaches [30,32] warrant mention. Study [32] proposes a hybrid RSU deployment structure combining three distinct approaches: stationary RSU deployment, utilizing bus-mRSUs on public transport, and utilizing dedicated mRSUs. The study investigates the optimal RSU deployment strategy to maximize spatiotemporal coverage within a limited budget and formulates an optimization problem, demonstrating its NP-hardness. The mRSU concept is explored in studies by the authors of [29,33], where mobile RSUs function as dynamically repositioning RSUs. Both studies aim to minimize data loss within the V2X system. Numerical analysis of systems with and without mRSUs demonstrated that utilizing mRSUs significantly reduces the overall network load.

The lack of comprehensive models addressing the challenges in this research area, along with scientifically sound approaches, concepts, models, methods, and algorithms for targeted activities to ensure the required level of ITS automation and balance between key characteristics during V2X technology operation, also hinder its effective implementation. This highlights the scientific significance and relevance of addressing this problem comprehensively.

Several challenges arise during modeling, the solutions to which would enable the creation of a comprehensive management model, including limited computational resources of vehicles and ITS infrastructure, energy efficiency, and data transmission network characteristics.

The quest for a comprehensive model that enables the distribution and prediction of system parameters and resources under various computational tasks is a primary objective for many researchers. This issue is of considerable importance for the implementation and utilization of V2X technology.

Building a comprehensive model invariably involves selecting key optimization parameters. Study [34] identifies four main parameters:

- 1) Network coverage: A primary goal of most routing algorithms is to ensure high and efficient network coverage to deliver essential and useful information to as many network nodes as possible.
- 2) Network latency: The impact of utilizing fog and edge data processing methodologies within the network is described by calculating the average network latency. The average message delivery latency in the network is calculated by summing the latencies for each unique message across all nodes where it was first received.
- 3) Processing ratio: The ratio of data processing at the edge, fog, and cloud layers – a metric describing the effectiveness and interrelation between these layers within the FOGO architecture, which heavily depends on the number of fog nodes. The "edge/fog/cloud" processing ratio is calculated as the ratio of the total number of messages processed in a layer to the total number of messages generated during the simulation.
- 4) Processing efficiency: Data processing efficiency is calculated for each network architectural layer: edge, fog, and cloud. It is calculated as the ratio of the average processing time at a lower-level node to the average data transmission and processing time at a higher-level node.

These parameters play a crucial role in building a comprehensive and effectively functioning V2X system, thus necessitating a comprehensive approach to optimization. In response to this challenge, the authors' earlier work [33] proposed a novel conceptual scheme for organizing a V2X system, briefly presented in Figure 1.

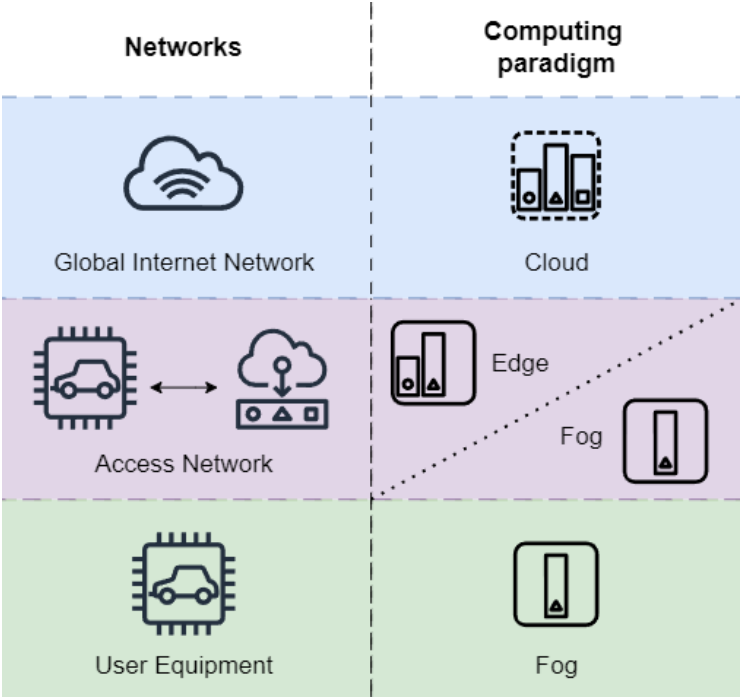


Figure 1. V2X computing paradigm.

The proposed conceptual mathematical model is based on two main principles:

Principle 1: RSUs and mRSUs will function as fog nodes, possessing the computational capabilities of edge servers to service tasks from OBUs within their coverage area. OBUs, in turn, will be considered as second-type fog nodes serviced by system elements, characterized by high mobility and significantly lower computational capabilities. Computational tasks exceeding the capabilities of edge servers will be offloaded to the cloud. Thus, a three-element V2X system architecture—OBU-RSU/mRSU-CLOUD—is considered, as illustrated in Figure 1. This concept enables the construction of a three-tier system, proposed as a hypothesis in [33].

Principle 2: The construction of a comprehensive V2X system should be based on the following optimized parameters, as classified in [35]: coverage, latency, reliability, throughput, communication range, velocity, and multi-operator's operation.

3. Materials and Methods

The primary components of a V2X system are the On-board Unit (OBU) and the Roadside Unit (RSU). The OBU is a device installed in a vehicle, responsible for communication with other vehicles (V2V, Vehicle-to-Vehicle) and infrastructure elements (V2I, Vehicle-to-Infrastructure). The OBU collects and transmits data necessary for enhancing road safety, informing the driver, and supporting various applications such as navigation and collision warning systems. The RSU is a stationary device installed along roads or on other infrastructure elements. RSUs facilitate communication with OBUs (V2I) and among themselves (I2I, Infrastructure-to-Infrastructure). The primary function of an RSU is to collect, process, and transmit data received from vehicles, as well as provide access to information necessary for the correct operation of transportation applications and the improvement of traffic coordination. Mobile RSUs (mRSUs) are increasingly utilized to organize communication between OBUs and RSUs. Their purpose is to move across the map and address tasks directly near the user, i.e., the OBU.

Furthermore, different variations may include Multi-access Edge Computing (MEC). MEC servers provide computational resources at the network edge, enabling data processing closer to its source, i.e., the vehicles. MEC servers allow for real-time processing of data received from OBUs, minimizing latency and ensuring faster responses to events. MEC can interact with both OBUs and RSUs, providing additional computational power and data storage capabilities. Within this task's framework, MEC will be considered a "cloud" that any RSU can access with a guaranteed latency for data synchronization. Figure 2 illustrates the interaction of V2X system elements.

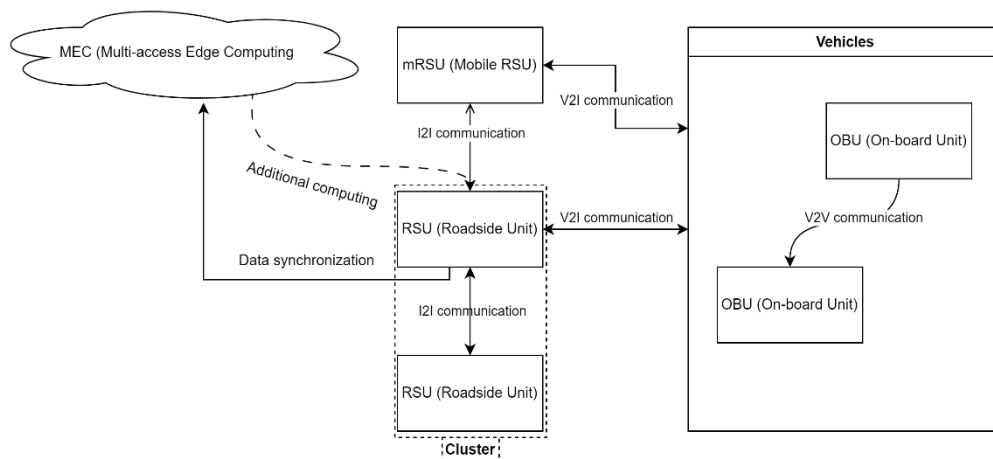


Figure 2. Interaction of V2X system elements.

When simulating a system involving vehicles, roads or lanes are required. For simplification, we assume all roads have one main lane and one opposite lane. The set of road connections can be interpreted as a graph. By mapping the graph onto a graphical interface, we obtain a map of the terrain where the vehicles will move. For model simplification, intersections will not be further burdened with traffic lights or pedestrian crossings. Therefore, it's necessary to create the following components: RSU, OBU (vehicle), and road (lane).

The well-established mathematical framework detailed in [36] can be employed to organize vehicle movement across the created map. Using a greedy algorithm [37] for appropriate device selection has proven effective, but it has drawbacks in high-load networks. In such cases, we utilize the Hungarian algorithm, which solves the assignment problem in polynomial time.

3.1. Mathematical Model

We will now briefly review the mathematical framework employed for implementing the V2X system model and its subsequent simulation.

In the authors' earlier work, a rectangular metric was used for organizing the road network due to its widespread adoption and ease of description. In the present study, the road structure model is represented generally as a graph, where intersections are vertices and roads are edges. This approach is significantly more flexible and allows for an expanded class of optimization problems. Note that the choice of metric, or the method of distance calculation, becomes secondary and depends on the specific tasks and road maps used for the simulation.

We will outline the core principles concerning traffic organization:

Vehicle generation, which will be managed subsequently, is described by Equation (1), defining the speed distribution density for each OBU on a road section within a single RSU's coverage area. The speeds represent a stream of independent, identically distributed random variables with values ranging from a specified minimum to maximum speed, i.e., $u_{min} \leq u \leq u_{max}$.

$$\varphi(u) = \frac{2}{\sigma\sqrt{2\pi}} e^{-\frac{(u-\mu)^2}{2\sigma^2}} \cdot \frac{1}{\operatorname{erf}\left(\frac{u_{max}-\mu}{\sigma\sqrt{2}}\right) - \operatorname{erf}\left(\frac{u_{min}-\mu}{\sigma\sqrt{2}}\right)} \quad (1)$$

OBUs interact with RSUs by exchanging data blocks, for example, information regarding traffic congestion on a route section. Let each OBU and each RSU generate information as a vector M of uncorrelated data elements $M = \{1, 2, \dots, m\}$.

For numerical simulation, we assume that the random variable representing the probability of an element i in the tuple M being requested by an OBU follows a Zipf distribution, calculated using Equation (2):

$$p(i) = \frac{i^{-A}}{\sum_{k=1}^m k^{-A}}, \quad (2)$$

where A is the distribution asymmetry coefficient and corresponds to the loading degree of the vehicle system.

If an OBU accelerates within a road section serviced by RSU $s = u_i$ m/s, and the coverage area of this section spans L_s meters, then the maximum number of requests that the RSU can process from a specific OBU is calculated using Equation (3):

$$k = \left\lfloor \frac{L_s}{u \cdot \alpha_s} \right\rfloor. \quad (3)$$

It's worth noting that in real-world conditions, signal attenuation occurs naturally as the vehicle moves away from the roadside unit within the RSU's coverage area. Our simulation scheme employs the well-known Hata (Okumura-Hata) model [38] to account for the building coefficient and its influence on attenuation.

Data loss occurs during OBU-RSU interaction, described by Equation (4). Let us enter the loss function when processing RSU OBU requests. Let the OBU at number i form a request $D(i)$, then the interaction loss on the length portion of the L_s will be:

$$V(i) = \begin{cases} D(i) - C \cdot k, & C \cdot k < D(i) \\ 0, & C \cdot k \geq D(i) \end{cases}. \quad (4)$$

If some of the requested information needs to be updated, then the total interaction time will be (5):

$$\sum_{i=1}^k (t_i^{(0)} + \gamma \cdot t_i^{(1)}) \leq \frac{L_s}{u}, \quad (5)$$

where $\gamma = 1$, if the data is outdated, and $\gamma = 0$, if the data does not require updating, for each pair of vehicles RSU.

Describing the mathematical framework for mRSU operation requires a more detailed explanation of pathfinding algorithms in graphs since we assume that in all scenarios, mRSUs move along dedicated lanes without encountering traffic jams.

It is convenient to utilize Estimated Time of Arrival (ETA) [39] as a cost function: the resulting route will be the shortest among all possible routes. In the simplest case, travel time comprises two factors: time spent traversing graph edges and additional time for maneuvering when transitioning between graph vertices.

$$ETA = \sum_{i=1}^n t_i + \sum_{i=1}^m c_i,$$

where the first summation represents the travel time along graph edges and the second represents penalties for maneuvers. If the mRSU moves along a dedicated lane on the previously described route, the average travel time remains constant. If mRSU movement to a specific map point is considered, then at each stage, the element of our dynamic system closest in terms of arrival time must be selected.

The problem with the chosen cost function, ETA, lies in the need to consider route parameters beyond just travel time. The identified route may be highly inconvenient for several reasons: poor road quality (e.g., poor surface, numerous potholes, lack of markings) or complex turns or intersections that should be avoided. Iterating through all possible route options would consume significant computational resources, impacting the overall system. Therefore, our study employs statistical analysis to determine preferred edges for traversal, formalized by Equation (6):

$$F_{cost}(route) = \sum_{i=1}^n \frac{t(edge_i)}{f(p_{edge_i})} + \sum_{i=1}^m c^*(turn_i) + \sum_{i=1}^k r_i^* \quad (6)$$

where $f(p)$ – is a normalized weight function approximating the attractiveness of each edge on the route, r_i^* – the portion of penalties accumulated at traversed intersections. Experimental results indicate that a linear function $f(p) = p$ yields the best results.

Systemic errors in speed determination can occur on certain roads (e.g., due to insufficient data), resulting in a suboptimal calculated route. However, the proposed approach significantly reduces the average error magnitude.

3.2. Simulation System

Selecting an environment or tools for developing a V2X system simulation requires careful consideration of both road traffic and data transmission modeling implementations.

A common and widely used combination is OMNeT++ (<https://omnetpp.org/>) coupled with VEINS (<https://veins.car2x.org/>), SUMO (<https://eclipse.dev/sumo/>), and INET (<https://inet.omnetpp.org/>). An alternative is AnyLogic (<https://www.anylogic.com/>), which also offers built-in modules for traffic and road simulation. Let's consider their respective advantages and disadvantages.

OMNeT++ is a flexible and powerful discrete event simulator widely employed for network simulation. Its modular design supports seamless integration of various components such as VEINS and INET. Its advantages include high customizability, robust community support, and powerful visualization tools. However, effective utilization of OMNeT++ requires a substantial learning curve, posing a significant constraint for this project.

VEINS (Vehicles in Network Simulation) is a framework that bridges road traffic simulation with network interaction simulation. Operating in conjunction with OMNeT++ and SUMO, VEINS provides high-fidelity modeling of both vehicular movement and data transmission. Its strength lies in integrating network models with vehicle models. However, utilizing VEINS necessitates familiarity with both OMNeT++ and SUMO, thereby increasing development complexity.

SUMO (Simulation of Urban Mobility) is a widely adopted open-source road traffic simulator, enabling vehicle movement modeling in urban and suburban environments. SUMO offers a high degree of detail and supports complex traffic scenario simulations. Its advantages encompass

integration capabilities with other tools like OMNeT++ and VEINS, alongside a comprehensive feature set for scenario customization. A primary drawback of SUMO is its complex configuration and the prerequisite for preparing road network data.

INET is a library for OMNeT++ providing models of network protocols and devices. INET supports various network technologies, including Wi-Fi, Ethernet, and TCP/IP, making it well-suited for modeling data transmission networks in V2X systems. Its advantage lies in its extensive functionality and the ability to model network interactions in detail. However, similar to OMNeT++, INET requires considerable knowledge and experience for effective use.

AnyLogic is a comprehensive simulation platform offering built-in modules for road traffic and infrastructure modeling. AnyLogic's distinguishing feature is its combination of various modeling paradigms, including discrete event, system dynamics, and agent-based modeling. Its integrated traffic and road simulation module enables the creation and customization of traffic scenarios without requiring external simulator integration. AnyLogic's intuitive interface and capacity for rapid development of complex models are advantageous. A potential drawback is the high cost of licensing, although AnyLogic provides a free educational license with limited features compared to the OMNeT++, VEINS, and SUMO combination for specialized V2X system simulation.

Analysis of the presented tools revealed that each environment possesses both advantages and disadvantages. However, considering the complexity of comprehensive integration and utilization of these tools collectively, the need arose for an alternative approach. Furthermore, extending the functionality of certain closed systems presented challenges. Given these circumstances, we opted to explore an alternative approach for creating the simulation system using Python.

Python is well-known for its simplicity and powerful libraries for scientific computing and simulation, such as NumPy, SciPy, and SimPy. However, Python's interpreted nature, leading to slower execution speeds compared to compiled languages, can be problematic when simulating complex real-time systems. To address these limitations, we selected GDScript, the scripting language integrated within the Godot Engine (<https://godotengine.org/>). GDScript's syntax resembles Python, facilitating a smooth transition for developers familiar with Python. Crucially, unlike Python, GDScript is compiled, providing a significant performance advantage. This makes it suitable for simulation tasks where high performance is essential.

Godot is a modern, open-source game engine offering a robust environment for creating both 2D and 3D applications. While primarily designed for game development, Godot's flexibility and extensibility allow for diverse applications, including simulating network systems like V2X. The compiled nature of GDScript and Godot's optimizations enable real-time simulation with minimal latency. Furthermore, Godot facilitates the visualization of vehicle movement and network interactions, enhancing the clarity of simulation results. Godot's modular architecture and open-source nature allow the system to be easily extended and customized to specific project requirements. Thus, the choice of Godot and GDScript effectively addresses the key challenges encountered, ensuring high quality and performance within the simulated system. Importantly, working with Godot offers the possibility of future development in 3D, incorporating the effects of network interference from surrounding buildings.

As described earlier, roads are fundamental elements within the V2X system. While manual road rendering is possible, developing a dedicated editor is resource-intensive. Therefore, we opted to utilize readily available cartographic data, significantly streamlining the development process. OpenStreetMap (OSM) was chosen for this purpose.

OpenStreetMap (OSM) is an open and freely available geographic database, providing detailed information on roads, buildings, and other infrastructure elements globally. OSM's community-driven nature ensures data relevance and accuracy. To integrate cartographic data into our simulation system, we employed an approach where the map is automatically generated from OSM data. This allows for rapid retrieval of current data for any location, readily applicable in the simulation. Python, with its powerful geospatial analysis libraries, was chosen for processing OSM data.

Initial location data extraction is performed with Python code utilizing the osmnx library, simplifying the download and processing of OpenStreetMap data. The following Python code snippet demonstrates this data extraction process:

```
import osmnx as ox

# Define the location
place_name = "Manhattan, New York City, New York, USA"
G = ox.graph_from_place(place_name, network_type='drive')

# Extract data
nodes, edges = ox.graph_to_gdfs(G, nodes=True, edges=True)

# Save data
nodes.to_csv("nodes.csv")
edges.to_csv("edges.csv")
```

The extracted files contain the following columns:

- nodes.csv: osmid, y, x, highway, street_count, ref, geometry
- edges.csv: u, v, key, osmid, name, highway, maxspeed, oneway, reversed, length, geometry, lanes, ref, access, bridge, tunnel, width, junction

Godot provides the Path2D element, suitable for this task. It connects points (nodes) to form roads. Subsequently, PathFollow2D can be used to move vehicles along these Path2D elements, which will be detailed below. This process resulted in the map depicted in Figure 3, with a scale of 1 meter to 1 pixel. Camera adjustments within the scene enable map zooming.



Figure 3. Graph mapping (MAP).

The chosen approach offers flexibility by enabling users to create custom maps and utilize them within the traffic flow simulation environment.

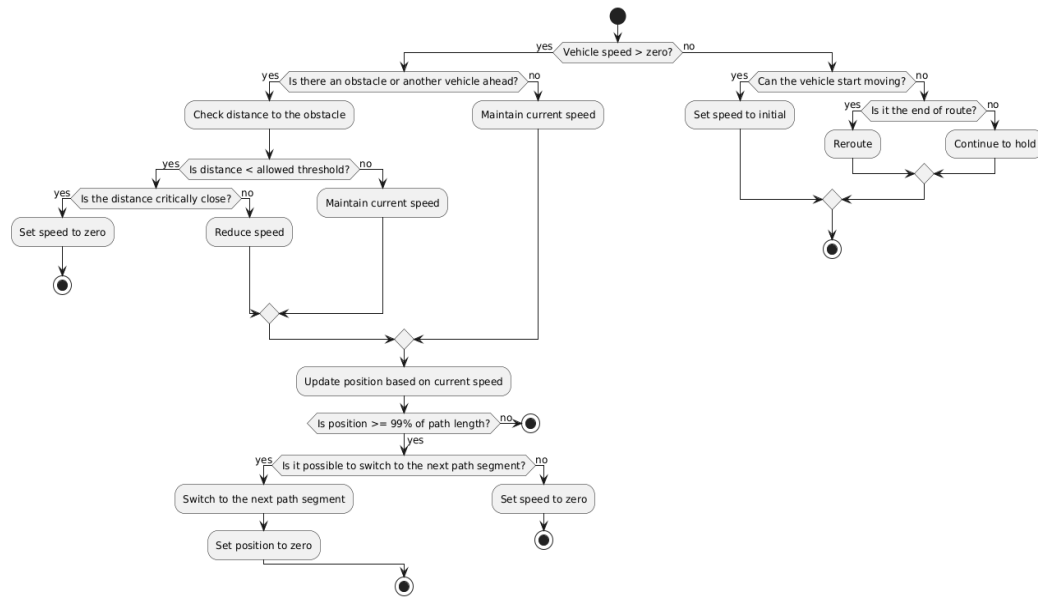
We utilize PathFollow2D in Godot to represent vehicles, treating the path (Path2D) as a continuous straight line, even with multiple points and curves. For clarity, such "paths" will be referred to as sections, and the combination of sections will be a route. By obtaining the length in pixels, we move the object along the "straight line" at a set speed. We assume an average speed of 30 km/h, which translates to approximately 8.5 m/s. With a 1:1 ratio, vehicles move at 8 and 12 pixels per second.

A vehicle's route from one point to another consists of a set of sections (Path2D) determined using the A* algorithm [40]. Thus, a vehicle initially has the shortest route to its destination.

Vehicle movement alone is insufficient for simulation. We must also account for these vehicles generating traffic. For simplicity, vehicles cannot overtake each other and may experience congestion if the leading vehicle stops.

Upon entering a new section (Path2D), the following algorithm is executed: For every vehicle within the current section, we determine its location as the distance traveled from the section's start. If the distance allows for a transition, the vehicle moves to the next section and continues at its set speed, maintaining a distance of 20 pixels from the preceding vehicle. If the distance is insufficient, the transition does not occur.

Vehicles spawn at their route's starting point. Upon reaching their destination, they randomly select a new destination and continue moving.

**Figure 4.** Traffic algorithm scheme.

RSUs are crucial system components. Their implementation in the software must consider not only their physical properties, adhering to the IEEE 802.11bd standard [41], but also their placement on the map.

The RSU operation and OBU interaction algorithm is as follows: Using an Area2D element, a monitored zone is established. When an OBU enters this Area2D, it is considered within the coverage area and capable of data transmission. Accurate simulation also requires incorporating distance-based attenuation and accounting for data loss. Information transmitted by the RSU can be classified as either useful or overhead. Useful information consists of commands for the OBU, while overhead information constitutes a specific volume of bits. If messages need to be sent, the RSU allocates transmission speed equally among connected OBUs. Message transmission is simulated with packet-based modeling, meaning a single message may comprise several packets. A primary channel is

assumed for multimedia traffic, while a secondary channel is used for transmitting critical information like instructions.

The primary interface provides information about the selected element. Currently, only RSUs are supported, though minor modifications could incorporate additional placeable objects. Upon clicking "Select Folder," the user is prompted to open a folder containing simulation configurations. Upon successful selection and loading, a second window containing the map opens.

A dedicated block within the interface displays information about the selected map element. However, it currently remains empty, as the specific information display requirements have yet to be defined at this development stage. Defining these requirements would delay prototype creation beyond the scope of this practical implementation.

A system accepting an event list as input has been designed for creating simulation scenarios. Events can be specified for each simulated object within the system.

3.3. Description of the Simulation Environment and Experimental Scenarios

The simulation was conducted using a Windows 11 operating system with the following software specifications: RTX 4060 graphics card with 8 GB of video memory, AMD Ryzen 7 7735HS processor with 8 cores and 16 threads at 3.2 GHz, 16 GB of DDR5 RAM at 4800 MHz.

Key system parameters for the model were derived from the IEEE 802.11bd standard documentation [41], some of which are presented in Table 1.

Table 1. Model system characteristics.

Parameter	OBU/mRSU (On-board unit/ Mobile On-board unit)	RSU (Roadside Unit)
Frequency	5.9 GHz	5.9 GHz
Bandwidth	10 Mb/s	10 Mb/s
Noise Figure	9 dB	9 dB
Transmitter Power	20 dBm	25 dBm
Antenna	1 meter	3 meters
Range radius	500 meters	250 meters

Initially, we will describe the simulated system. The average speed of vehicles (OBUs and mRSUs) is 30 km/h. RSUs have access to MEC, and if they cannot provide up-to-date information upon request, they query the MEC for updates. If the RSU belongs to a cluster, it queries the cluster for information before sending a request to the MEC.

Even if two or more RSUs are located on a section, we assume OBU-RSU interaction occurs with only one station, thereby avoiding redundant requests from the OBU. In other words, in cases of multiple connection possibilities, the OBU autonomously selects a single RSU for connection. If an OBU isn't connected to any RSU, it connects to the first one encountered along its route. If multiple connection options exist, it retrieves information about the current number of OBUs connected to each RSU. If this number is less than 20% of the current RSU's active connections, the OBU switches to the less loaded RSU. OBUs perceive mRSUs as RSUs, while RSUs perceive mRSUs as OBUs. The primary difference between mRSUs and OBUs is the ability to process requests and store data, while mRSUs are not connected to the MEC and consequently redirect requests to a connected RSU according to the logic described above. The primary distinction between mRSUs and RSUs is their mobility along OBU movement paths.

Within the simulation framework, a vehicle (OBU) may encounter an emergency situation, triggering a message to the nearest RSU, which then propagates this information. Initially, all vehicles connected to this RSU are notified about the incident, followed by the remaining infrastructure, including mRSUs.

Modulated event-case. After a period of continuous movement, every tenth vehicle experiences an "accident," resulting in a complete stop and blocking the road. After waiting for 10 to 60 seconds, the vehicle resumes movement. In the event of an accident, the system is designed to send a critical message. Subsequently, vehicles within RSU coverage are expected to update their routes, while other vehicles continue on their initial paths. Once the accident is cleared, a message is transmitted to the RSU, enabling subsequent vehicles to proceed to utilize unblocked route. Vehicles spawn at random locations across the map, simulating a more realistic and heavily loaded traffic system.

Our model considers 10^6 unique content requests. The size of the response to each request ranges from 4194304 to 8388608 bits.

It's important to note that the simulation was conducted on a map representing the infrastructure of Manhattan, NYC, USA (Figure 3), which facilitated subsequent comparison of results and evaluation of the developed simulation system's effectiveness.

The program's output includes three scenarios. Each scenario employs two RSU placement schemes:

Scheme 1: RSUs are placed as densely as possible, with a data collection and transmission device at each road section or block. Data transmission to the higher level is carried out independently by each RSU over a dedicated communication channel (possibly fiber optic). The coverage areas of different RSUs may overlap. OBUs can sequentially query the least loaded RSU within their coverage

area for data exchange. The main drawback of this arrangement is the high cost of network organization and its "noise" due to numerous cross-requests.

Scheme 2: RSUs are sparsely placed on the map, with the sole condition that the entire map is covered by RSU coverage areas. For centralized interaction with the MEC, roadside units are grouped into clusters with a shared cache. A primary element is selected within each cluster to interact with the MEC. While significantly reducing system management costs, this scheme may increase data loss within the system and potentially lead to complete system blockage.

1. The interaction model includes only four system elements: OBU-RSU-MEC-CLOUD.
2. As previously mentioned, using mRSUs significantly improves system efficiency. Mobile RSU movement routes are chosen according to designated public transport routes. mRSUs move on a schedule, enabling prediction of their location and anticipated appearance at specific points. The main disadvantage of this scheme is the presence of idle mobile RSUs not involved in system operation during periods of low network load. However, this is expected to be mitigated by high traffic flow or long mRSU travel routes. Within this model, all mRSUs are assumed to be en route according to the schedule.
3. This scenario proposes a new mRSU management scheme where each system element is responsible for serving a specific map area. Alternatively, it can be stationed in a depot until notified of increased load within its designated area. In this case, the mRSU travels to the call point via the shortest route, the mathematical description of which is provided in Section 3.1.

4. Results

We will now analyze the results obtained through the simulation. For each scenario described in Section 3.3, ten generations are considered for two average speed variants: 30 km/h and 50 km/h.

1. Scenario 1.

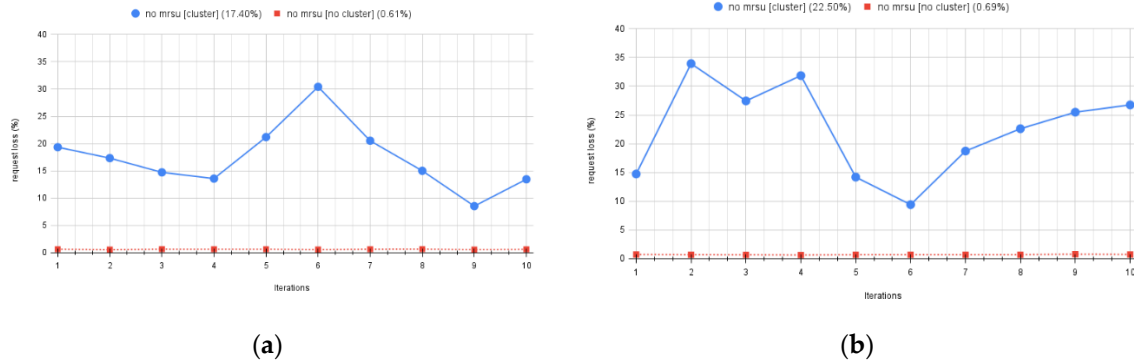


Figure 5. Average percentage of losses at 10 iterations (no mRSU): (a) average speed 30 km/h; (b) average speed 50 km/h.

2. Scenario 2.

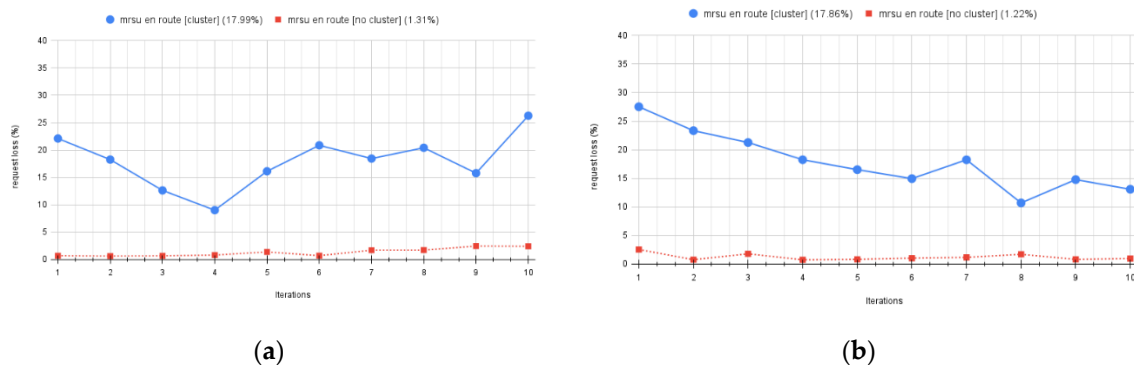


Figure 6. Average percentage of losses at 10 iterations (floating mRSU): (a) average speed 30 km/h; (b) average speed 50 km/h.

3. Scenario 3.

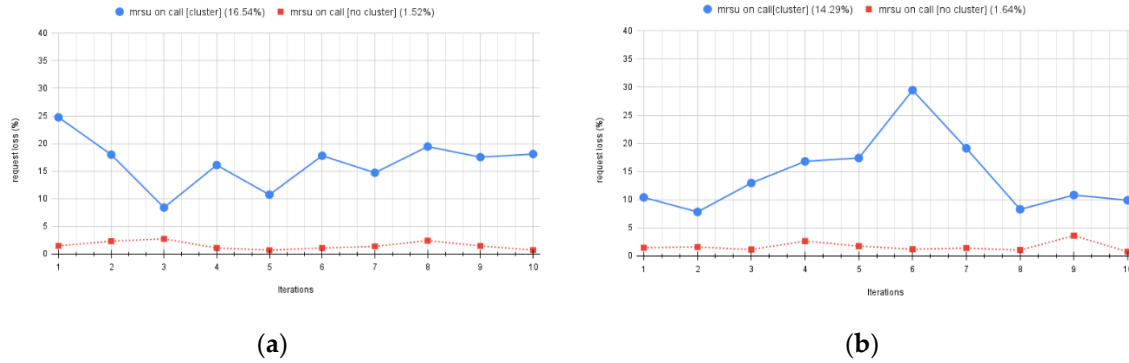


Figure 7. Average percentage of losses at 10 iterations (on call mRSU): (a) average speed 30 km/h; (b) average speed 50 km/h.

The simulation results indicate that in all three scenarios, the direct connection of each RSU to the MEC yields lower losses compared to the cluster-based approach. It should be noted that data loss, even in an overloaded system, tends towards 0%. This is because the data transmission rate directly from the RSU to the MEC is high and without attenuation. However, as previously mentioned, the maintenance and operation of such a system are costly. Therefore, further comparison should be limited to utilizing clustering.

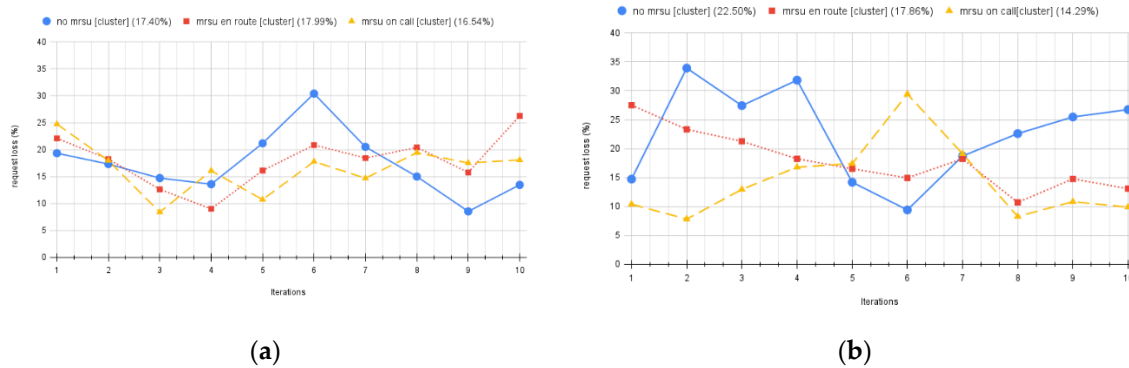


Figure 8. Average percentage of losses at 10 iterations (cluster): (a) average speed 30 km/h; (b) average speed 50 km/h.

In this case, the most effective scenario in terms of losses involves a high average speed for mRSUs following on-call route. Detailed examination of each graph pair leads to the following conclusions:

1. In the scenario without mRSUs, lower speeds result in lower losses. This stems from OBUs having sufficient time at lower speeds to receive responses to most of their requests while within a specific RSU's coverage area.
2. The benefit of employing mRSUs becomes substantial only at higher speeds. This is attributed to the mRSU scheme being more flexible with respect to speed and not increasing the percentage of losses with speed variations.
3. The reduction in losses by introducing mRSUs into the system is less significant than anticipated during the modeling phase. Despite achieving the best results with mRSUs, the improvement in terms of losses is not substantial under conditions of uniform map coverage by RSUs. A potential reason for the limited efficiency increase is the choice facing OBUs: which entity to

interact with. The introduction of mRSUs into the coverage area offloads some connections onto them, causing current connections with RSUs to break and be replaced by mRSU connections. This leads to unplanned losses. Some of these broken connections might have completed successfully with a longer average waiting time. Furthermore, an mRSU to which an OBU connects continues moving irrespective of the main traffic flow, potentially causing their routes to diverge and generating additional losses.

An essential system parameter is stability. A stable simulation system is expected to reach an equilibrium state in terms of losses. The developed system is stable, as shown in Figure 9. The horizontal axis represents system operation time, while the vertical axis represents the percentage of losses. A stable pattern emerges after an average of 100 seconds, suggesting that calculation times for obtaining statistics can be reasonably reduced.

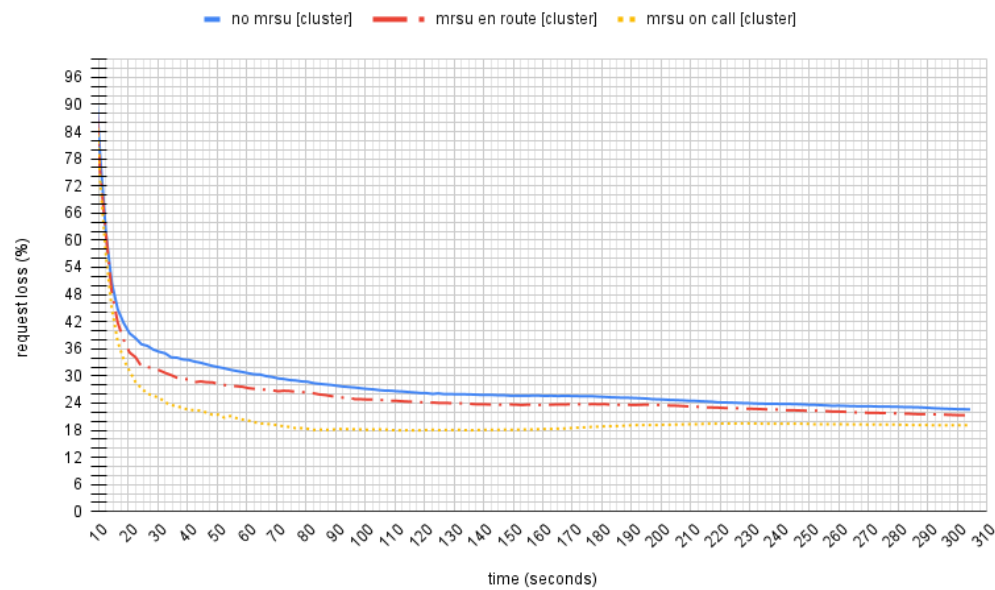


Figure 9. Percentage of request losses over simulation time.

5. Discussion

The rapid development of modern technologies in automotive transport necessitates addressing several challenges regarding the efficiency of interaction between road infrastructure elements. There are two primary paths towards tackling this problem: enhancing the technical characteristics of utilized equipment (particularly constructing networks based on the new 5G mobile communication standard) and improving the organizational and algorithmic aspects of V2X infrastructure operation.

This paper presents a new simulation system enabling map construction for the simulation, defining the position of key V2X system elements, specifying their characteristics, conducting simulations considering transmitted message sizes, and evaluating results accounting for interference and noise through loss parameters.

Several approaches are utilized for organizing the dynamic system's operation, including clustering, fog computing, and edge computing. Introducing mRSUs to assist stationary RSUs can significantly reduce V2X system maintenance costs while controlling minor increases in losses. Therefore, selecting a model requires finding a balance between the number of RSU clusters and the number of mRSUs saturating the system. With high-efficiency map coverage using RSUs, the benefits of mRSUs moving with traffic flow become negligible.

Numerical simulation results demonstrate that the final outcome is significantly influenced by the average flow speed, RSU coverage density, and the number of RSUs requiring direct MEC connection.

In summary, the proposed approach offers a flexible and efficient platform for V2X simulation. Leveraging OpenStreetMap data enables realistic road network representation, while the simulation system allows for detailed modeling of OBU-RSU interactions, including data transmission, attenuation, and losses. Simulations demonstrated that direct RSU-MEC connections minimize data loss, although clustering offers a more cost-effective solution. The integration of mRSUs, particularly at higher speeds, can further enhance system performance, albeit with limited gains under uniform RSU coverage due to dynamic connection management challenges. Furthermore, system stability analysis confirmed equilibrium within a reasonable timeframe. However, known limitations exist due to the simplified mobility model as potential complexity increases with 3D modeling.

Future research directions include assessing the energy efficiency of the proposed V2X system models and identifying optimal offloading algorithms for OBUs and RSUs within clusters.

The authors' implementation of the proposed approach is publicly available on GitHub [42].

Author Contributions: Conceptualization, A.V.; methodology, A.V. and P.P.; software, G.T. and P.P.; validation, GT and P.P.; formal analysis, A.V. and P.P.; data curation, G.T.; writing—original draft preparation, GT and P.P.; writing—review and editing, A.V.; visualization, G.T.; project administration, A.V.; funding acquisition, A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Russian Science Foundation, grant number 24-29-00304, <https://rscf.ru/en/project/24-29-00304/>.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mueck, M.; Karls, I. Networking Vehicles to Everything: Evolving Automotive Solutions. *Walter de Gruyter*, 2018. 233 p.
2. Abdelkader, G.; Elgazzar, K.; Khamis, A. Connected vehicles: Technology review, state of the art, challenges and opportunities. *Sensors* **2021**, *21*, 7712. DOI: 10.3390/s21227712
3. Wang, J.; Shao, Y.; Ge, Y.; Yu, R. A survey of vehicle to everything (V2X) testing. *Sensors* **2019**, *19*, 334. DOI: 10.3390/s19020334
4. Bali, R.S.; Kumar, N.; Rodrigues, J.J. Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions. *Vehicular Communications* **2014**, *1*, 134–152. DOI: 10.1016/j.vehcom.2014.05.004
5. Cooper, C.; Franklin, D.; Ros, M.; Safaei, F.; Abolhasan, M. A Comparative Survey of VANET Clustering Techniques. *IEEE Communications Surveys and Tutorials* **2017**, *19*, 657–681. DOI: 10.1109/COMST.2016.2611524
6. Raza, S.; Wang, S.; Ahmed, M.; Anwar, M.R. A survey on vehicular edge computing: architecture, applications, technical issues, and future directions. *Wireless Communications and Mobile Computing* **2019**, *2019*, 3159762. DOI: 10.1155/2019/3159762
7. Hou, X.; Li, Y.; Chen, M.; Wu, D.; Jin, D.; Chen, S. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures. *IEEE Transactions on Vehicular Technology* **2016**, *65*, 3860–3873. DOI: 10.1109/TVT.2016.2532863
8. Jameel, F.; Javed, M.A.; Zeadally, S.; Jantti, R. Efficient Mining Cluster Selection for Blockchain-Based Cellular V2X Communications. *IEEE Transactions on Intelligent Transportation Systems* **2021**, *22*, 4064–4072. DOI: 10.1109/TITS.2020.3006176
9. Paramonov, A.; Khayyat, M.; Chistova, N.; Muthanna, A.; Elgendy, I.A.; Koucheryavy, A.; Abd El-Latif, A.A. An Efficient Method for choosing Digital Cluster Size in Ultralow Latency Networks. *Wireless Communications and Mobile Computing* **2021**, *2021*, 9188658. DOI: 10.1155/2021/9188658
10. Abbas, F.; Liu, G.; Fan, P.; Khan, Z. An efficient cluster-based resource management scheme and its performance analysis for V2X networks. *IEEE Access* **2020**, *8*, 87071–87082. DOI: 10.1109/ACCESS.2020.2992591
11. Vladkyo, A.; Tambovtsev, G.; Podgornaya, E.; Chelloug, S.A.; Alkanhel, R.; Plotnikov, P. Cluster-Based Vehicle-to-Everything Model with a Shared Cache. *Mathematics* **2023**, *11*, 3017. DOI: 10.3390/math11133017
12. Vladkyo, A.; Elagin, V.; Spirkina, A.; Muthanna, A.; Ateya, A.A. Distributed Edge Computing with Blockchain Technology to Enable Ultra-Reliable Low-Latency V2X Communications. *Electronics* **2022**, *11*, 173. DOI: 10.3390/electronics11020173
13. Fan, W.; Su, Y.; Liu, J.; Li, S.; Huang, W.; Wu, F.; Liu, Y. Joint Task Offloading and Resource Allocation for Vehicular Edge Computing Based on V2I and V2V Modes. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 4277–4292. DOI: 10.1109/TITS.2022.3230430

14. Hou, P.; Jiang, X.; Lu, Z.; et al. Joint Computation Offloading and Resource Allocation based on Deep Reinforcement Learning in C-V2X Edge Computing. *Applied Intelligence* **2023**, *53*, 22446–22466. DOI: 10.1007/s10489-023-04637-x
15. 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. DOI: 10.1109/JIOT.2018.2876298
16. Cai, G.; Fan, B.; Dong, Y.; Li T.; Wu, Y.; Zhang, Y. Task-Efficiency Oriented V2X Communications: Digital Twin Meets Mobile Edge Computing. *IEEE Wireless Communications* **2024**, *31*, 149–155. DOI: 10.1109/MWC.012.2200465
17. Ye, D.; Yu, R.; Pan, M.; Han, Z. Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach. *IEEE Access* **2020**, *8*, 23920–23935. DOI: 10.1109/ACCESS.2020.2968399
18. Liu, L.; Chen, C.; Pei, Q.; Maharjan, S.; Zhang, Y. Vehicular edge computing and networking: A survey. *Mobile Networks and Applications* **2021**, *26*, 1145–1168. DOI: 10.1007/s11036-020-01624-1
19. Bonomi, F. Connected vehicles, the internet of things, and fog computing. In Proceedings of the Eighth ACM International Workshop on VehiculAr Inter-NETworking (VANET 2011), Las Vegas, USA, 23 September 2011.
20. Khattak, H.A.; Islam, S.U.; Din, I.U.; Guizani, M. Integrating fog computing with VANETs: A consumer perspective. *IEEE Communications Standards Magazine* **2019**, *3*, 19–25. DOI: 10.1109/MCOMSTD.2019.1800050
21. Sarrianiannis, I.; Contreras, L.M.; Ramantas, K.; Antonopoulos, A.; Verikoukis, C. Fog-Enabled Scalable C-V2X Architecture for Distributed 5G and Beyond Applications. *IEEE Network* **2020**, *34*, 120–126. DOI: 10.1109/MNET.111.2000476
22. Alvi, A.N.; Javed, M.A.; Hasanat, M.H.A.; Khan, M.B.; et al. Intelligent task offloading in fog computing based vehicular networks. *Applied Sciences* **2022**, *12*, 4521. DOI: 10.3390/app12094521
23. Lee, J.; Ahn S. Adaptive Configuration of Mobile Roadside Units for the Cost-Effective Vehicular Communication Infrastructure. *Wireless Communications and Mobile Computing* **2019**, *2019*, 6594084. DOI: 10.1155/2019/6594084
24. Tonguz, O.K.; Viriyasitavat, W. Cars as roadside units: A self-organizing network solution. *IEEE Communications Magazine* **2013**, *51*, 112–120. DOI: 10.1109/MCOM.2013.6685766
25. Reis, A.B.; Sargento, S.; Tonguz, O.K. Parked cars are excellent roadside units. *IEEE Transactions on Intelligent Transportation Systems* **2017**, *18*, 2490–2502. DOI: 10.1109/TITS.2017.2655498
26. Heo, J.; Kang, B.; Yang, J.M.; Paek, J.; Bahk, S. Performance-cost tradeoff of using mobile roadside units for V2X communication. *IEEE Transactions on Vehicular Technology* **2019**, *68*, 9049–9059. DOI: 10.1109/TVT.2019.2925849
27. Evariste, T.; Kasakula, W.; Rwigema, J.; Datta, R. Optimal exploitation of on-street parked vehicles as roadside gateways for social IoV – a case of Kigali City. *Journal of Open Innovation: Technology, Market, and Complexity* **2020**, *6*, 73. DOI: 10.3390/joitmc6030073
28. Jiang, X.; Du, D.H.C. Bus-VANET: A bus vehicular network integrated with traffic infrastructure. *IEEE Intelligent Transportation Systems Magazine* **2015**, *7*, 47–57. DOI: 10.1109/MITS.2015.2408137
29. Plotnikov, P.V.; Tambovtsev, G.I.; Vladyko, A.G. Performance Evaluation of V2X Model with a Mobile Road Side Units. In Proceedings of the Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED), Moscow, Russian Federation, 15–17 November 2023. DOI: 10.1109/TIRVED58506.2023.10332617
30. Kim, D.; Velasco, Y.; Wang, W.; Uma, R.N.; Hussain, R.; Lee, S. A new comprehensive RSU installation strategy for cost-efficient VANET deployment. *IEEE Transactions on Vehicular Technology* **2016**, *66*, 4200–4211. DOI: 10.1109/TVT.2016.2598253
31. Li, G.; Ma, M.; Liu, C.; Shu, Y. Routing in taxi and public transport based heterogeneous vehicular networks. In Proceedings of the IEEE Region 10 Conference (TENCON), Singapore, Singapore, 22–25 November 2016, pp. 1863–1866. DOI: 10.1109/TENCON.2016.7848344
32. Ni, Y.; Zhao, C.; Cai, L. Hybrid RSU management in cybertwin-IoV for temporal and spatial service coverage. *IEEE Transactions on Vehicular Technology* **2022**, *71*, 4596–4606. DOI: 10.1109/TVT.2021.3138749
33. Plotnikov, P.V.; Tambovtsev, G.I.; Vladyko, A.G. Numerical Analysis of roadside Units Deployment Models in V2X Communication System. In Proceedings of the Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 12–14 March 2024. DOI: 10.1109/IEEECONF60226.2024.10496720
34. Balen, J.; Damjanovic, D.; Maric, P.; Vdovjak, K. Optimized Edge, Fog and Cloud Computing Method for Mobile Ad-hoc Networks. In Proceedings of the 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, USA, 15-17 December 2021, pp. 1303–1309. DOI: 10.1109/CSCI54926.2021.00266
35. V2X White Paper. Next Generation Mobile Networks Ltd.: San Jose, 2018.

36. Naeem, A.B.; Soomro A.M.; Saim, H.M.; Malik, H. Smart road management system for prioritized autonomous vehicles under vehicle-to-everything (V2X) communication. *Multimedia Tools and Applications* **2024**, *83*, 41637–41654. DOI: 10.1007/s11042-023-16950-1
37. Jungnickel, D. The greedy algorithm. *Graphs, Networks and Algorithms* 2013. pp. 135-161. DOI: 10.1007/978-3-642-32278-5_5
38. Shemshaki, M.; Lasser, G.; Ekiz, L.; Mecklenbrauker, C. Empirical path loss model fit from measurements from a vehicle-to-infrastructure network in Munich at 5.9 GHz. In Proceedings of the 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Hong Kong, China, 30 August 2015 - 02 September 2015, pp. 181-185. DOI: 10.1109/PIMRC.2015.7343291
39. Sun, Y.; Fu, K.; Wang, Z.; Zhou, D.; Wu, K.; Ye, J.; Zhang, C. CoDriver ETA: Combine Driver Information in Estimated Time of Arrival by Driving Style Learning Auxiliary Task. *IEEE Transactions on Intelligent Transportation Systems* **2022**, *23*, 4037–4048. DOI: 10.1109/TITS.2020.3040386
40. AStar2D – Godot Engine 4.3 documentation. 2024. Available online: https://docs.godotengine.org/en/stable/classes/class_aster2d.html (accessed on 01 October 2024).
41. Next Generation V2X – IEEE 802.11bd as fully backward compatible evolution of IEEE 802.11p. CAR 2 CAR Communication Consortium, **2023**.
42. Plotnikov, P.; Tambovtsev, G.; Vladyko, A. VANET Ver. 2.0.0 –Software Module for Modeling the Interaction of Edge Devices, 2024. Available online: <https://github.com/quanuhs/VANET> (accessed on 01 October 2024).

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.