mathematics

Article

Cluster-Based Vehicle-to-Everything Model with a Shared Cache

Andrei Vladyko [1], Gleb Tambovtsev [1], Elena Podgornaya [1], Samia Allaoua Chelloug [2,*], Reem Alkanhel [2] and Pavel Plotnikov [1,*]

1 Faculty of Fundamental Training, The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, Prospekt Bolshevikov 22-1, 193232 Saint Petersburg, Russia
2 Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
* Correspondence: sachelloug@pnu.edu.sa (S.A.C.); plotnikov.pv@sut.ru (P.P.)

Abstract: This paper presents an analysis of the effectiveness of the element interaction model in a vehicular ad hoc network (VANET). An analysis of the mathematical model and its numerical solution for the system of boundary device interactions in the traditional configuration of roadside unit (RSU) placement using single- and dual-channel connection between on-board units (OBU) and RSU is given. In addition, the model efficiency is improved using a clustering approach. The efficiency evaluation is based on calculating the percentage of unprocessed requests generated by OBUs during their mobility, the average power consumption and the magnitude of the delay in transmitting and processing the generated requests in the OBU–RSU system. The traditional and cluster models are compared. The results obtained in this paper show that each of the proposed models can be effectively implemented in mobile nodes and will significantly reduce the overall expected query processing time to improve the organization and algorithmic support of VANET. Along with this, it is shown that the developed approach allows for efficient power consumption when combining RSUs into clusters with a shared cache. The novelty of solving the problems is due to the lack of a comprehensive model that allows the distribution and prediction of the parameters and resources of the system for different computational tasks, which is essential when implementing and using V2X technology to solve the problems of complex management of VANET elements.

Keywords: vehicular communications; vehicle-to-everything; edge computing; computation capability; shared cache

MSC: 65K05

Citation: Vladyko, 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. https://doi.org/10.3390/math11133017

Academic Editor: Daniel-Ioan Curiac

Received: 30 May 2023
Revised: 3 July 2023
Accepted: 5 July 2023
Published: 7 July 2023

1. Introduction

In mobile communications environments, with cloud computing solutions and broadband communications, most computing tasks can be moved to the cloud for remote execution. Although this process is accelerating with each new generation of communications networks, offloading to the remote cloud implies a significant delay. The exponential growth of mobile data traffic, especially mobile video streaming, is proving to be a challenge in optimizing interactions within computational dynamical systems for edge computing. A number of analytical companies [1] have conducted studies and concluded that on average, the global mobile traffic will almost double over the next five years. There is no reason to believe that the growth will not accelerate in the future. It is worth noting that with the growing number of computer calculations and increase in processed information volumes, there is a need for parallel computing, both by using new algorithms and by extending the technical capabilities of communication channels. A no less important aspect of the problem in question is the significant growth in energy consumption in the process of operation of the considered systems. The increase in interaction speed with minimization of delays has a significant impact on this consumption. In connection with this, it is necessary to develop a new methodical approach to the analysis of computation delays, taking into account the energy costs and the possibilities of modification and modernization of communication channels.

One possible approach to solving the problem in question is the use of cloud computing in mobile communication environments. In this case, a proportion of the calculations is transferred from the physical device to a remote server, which allows an increase in the speed of calculations and reduces the specific power consumption. Despite the fact that due to the development of technologies, new upgraded standards of communication networks have appeared, the significant delay of calculations must be taken into account and, if possible, reduced. To this end, MEC (mobile edge computing) has been developed and successfully implemented [2].

Today, MEC has also become a new paradigm for intelligent transportation systems [3–5]. By introducing vehicular edge computing (VEC) into vehicular ad hoc networks (VANETs), service providers can provide ultra-low latency services, as edge computing tools deploy resources on the side closest to the user equipment.

Interaction within VANETs between vehicles (V2V, vehicle-to-vehicle) can be performed for speed and route data exchange, and between vehicles and edge computing devices or, for example, infrastructure elements (V2X, vehicle-to-everything) to offload local computing power, as well as to control traffic lights or parking machines [6]. VEC architecture relies on communication infrastructure and services provided by edge devices and LTE/5G technologies.

Edge devices can include on-board units (OBU), roadside units (RSU) and MEC-servers. Note that OBUs have small computational and storage resources, and the computational power of RSUs is also relatively limited due to the size of computing tasks and the high demand for real-time system responses. When OBUs generate tasks, they can connect to roadside devices, selecting the right time and the right MEC servers to offload tasks.

However, how and in what order to efficiently offload computing power remains an incompletely solved problem. Since MEC servers operate at the edge of the radio access network, connecting to RSUs and performing transmission tasks with them, their service area may be limited to the radio coverage of the RSU. Because of their high mobility, moving OBUs can pass through several RSUs and MEC servers and offload their computational tasks to any server they have access to, with the load on them varying greatly over time due to the uneven number of connected users.

The complexity of the problem under consideration is determined by its dynamic nature, as well as by the weak predictability of the spatial configuration of the system, even at medium-term anticipation intervals. Therefore, it is not clear a priori how and in what order the computational tasks should be unloaded. An additional limitation is that MEC servers can operate at the edge of the radio access network, which makes them difficult to use. An additional bottleneck remains the connection and caching at the level of OBU–RSU edge devices [7].

It is worth noting that the described interaction imposes additional costs in the form of increased power consumption. Computational latency occurs during content processing and transmission when MEC servers, RSUs and OBUs interact. Increased traffic flow in the network can cause unacceptable overhead and congestion. To reduce the network overhead, two main approaches related to increasing network heterogeneity with current traffic, as well as optimizing edge computation using caching and clustering, have been outlined above.

Thus, this work is aimed at solving the scientific problem of analyzing the stability of the interaction structure of VANET model elements and the research and design of the computing task management system based on vehicle-to-everything technology. This project is aimed at solving a scientific problem by developing a complex of models and methods for research and design of a vehicle-to-everything task management system in conditions of limitations imposed on data network characteristics, energy efficiency and computing resources of intellectual transport systems.

In response to this problem, in this paper, we propose a mathematical model and its numerical solution for an edge device interaction system in a traditional RSU placement configuration using single- and dual-channel connectivity between an OBU and RSU. In addition, an improvement of the model performance using the clustering approach was carried out. The efficiency evaluation was based on the calculation of the percentage of unprocessed requests generated by the OBU during their mobility, the average power consumption and the value of the delay in the transmission and processing of generated requests in the MEC–RSU–OBU system. The traditional and clustered models were compared. The results showed that each of the proposed models can be effectively implemented in mobile nodes and will significantly reduce the overall expected query processing time, improving the organization and algorithmic support of VANETs. At the same time, the developed approach allowed for efficient power consumption costs when combining RSUs into clusters with a shared cache.

Our contributions can be summarized as follows:

1. Analysis of a mathematical model of the interaction of VANET network elements by indicators: the value of data loss during exchange and energy consumption for servicing the interaction of network elements was determined.
2. A model of OBU interaction with RSUs and MEC in a classical mobility scheme and using clustering (two and four elements in a cluster) was developed and designed to ensure the efficiency of data transmission in a dynamic system.
3. Modeling and applicability analyses of two types of OBU and RSU interaction with single- and dual-channel connectivity were performed.
4. A comparative analysis of all the above-described schemes in terms of average energy costs is given.

The remainder of the paper is organized as follows. In Section 2, we provide a literature review on the topics of V2X systems, clustering and caching. Section 3 describes the general model of the dynamic system, describing the general parameters of the class of problems under consideration. In Section 4, we first formulate the problem, which is to minimize the data loss in the interaction of elements of the system, then we consider the question related to the energy consumption of the elements of the system in the exchange and transfer of data. In Section 5, a general scheme of the developed optimization algorithm is given, and the main numerical parameters of the simulation are given. Section 6 presents the results of numerical simulation, aiming to confirm the theoretical conclusions and analyze them, and Section 7 contains concluding remarks.

2. Related Works

Recently, a number of works have been carried out on the analysis of approaches to minimizing energy consumption under the constraints imposed on the delay value in the systems under consideration. In [8,9], a new approach for optimal network resource allocation based on convex optimization for minimizing the weighted total computational energy consumption under the computational latency constraints was proposed.

In [10], a group of authors proposed an algorithm for offloading distributed computations based on a model construction based on Nash equilibrium, and a comparative analysis of the efficiency of the algorithm with known approaches to optimization for two parameters, consumed energy and processing time, was performed. The numerical results confirmed that the proposed algorithm was able to achieve an excellent performance when offloading computations and good scaling as the number of users increased. However, the proposed model did not allow us to analyze systems with mobile users who can dynamically change their position during the computation offloading period.

In recent years, there have been many studies dedicated to the analysis of data caching approaches in mobile networks. The classic popularity-based caching scheme (PBCS), studied in detail in [11–14], was first applied to homogeneous systems, which can include cellular networks and social wireless network VANETs. In these schemes, RSUs do not interact with each other and always fill the free memory space with the most popular content. If the OBU cannot obtain the desired content from the local RSU, the request is redirected to the MEC server.

In [15], a new caching scheme was proposed. Considered a multithreaded caching scheme, the so-called small cell network supports multicast formation and distribution of requests within the network architecture. In this scheme, macrocell base stations (MBSs) and small cell base stations (SBSs) collect requests from OBUs during a certain period of data collection, while the SBSs process only those requests whose responses are stored in the device cache; the rest are forwarded to MBSs. From the base stations, macrocells are generated and processed, and interact with the MEC server. This multicast-enabled scheme reduces the interaction time for resource delivery with high latency in the interaction with the MEC server. Further research should focus on optimizing the interaction process with the remote server.

A caching scheme called femtocaching was proposed in [16]. It was assumed that the layout of RSUs was arranged in such a way that the coverage areas of base stations overlapped. In this case, the OBU could be served by any RSU in the coverage area in which it was located. Thus, in the absence of the requested information, the request from the OBU would be redirected to the other nearest RSU. This approach suggests that the cluster approach scheme for dividing RSUs into groups can lead to an optimal solution for minimizing the latency of interactions in the OBU–RSU system, and achieve the maximum percentage of served devices in the network and the optimal energy consumption.

Edge caching is a promising approach to reduce the load on transport networks. It plays an important role in VANET performance by providing cached data at the edge and reducing the load on the core network caused by the number of vehicles and the volume of data. However, due to the limited computational and storage capabilities of edge devices, it is difficult to guarantee that all content is cached and all device requirements are met for all users.

Here are some more works on caching. A hierarchical preemptive caching scheme, which jointly considers caching in vehicles and RSUs, is proposed in [17]. The optimization problem is formulated to find the optimal caching solution for RSUs and vehicles. The numerical results show that preemptive caching provides significant performance gains over the baseline reactive scenario. In addition, the results show that caching in vehicles with small cache sizes effectively affects the overall latency in the network. Paper [18] presents a preemptive caching strategy that uses information-centric network (ICN) flexibility of caching data anywhere in the network, not just at the edge as in conventional content delivery networks. The main contribution of the paper is the use of entropy to measure mobility prediction uncertainty and determine the best node for prefetching, which eliminates redundancy. Although prefetching at higher levels of the network hierarchy causes higher latency than at the periphery, our evaluation results show that increased latency does not negate the performance gain due to preemptive caching. Moreover, the gain is enhanced by reducing server load and achieving cache redundancy.

Studies on ICNs are conducted by a wide range of authors and are quite relevant. In [19], we develop an ICN with a mobility-aware anticipatory caching scheme to provide delay-sensitive services in Internet of Vehicles (IoV) networks. The real-time state and interaction of vehicles with other vehicles and RSUs is modeled using a Markov process. A mobility-aware preemptive border caching solution is applied that maximizes network performance with minimal transmission delay. Our numerical simulation results show that the proposed scheme outperforms related caching schemes by 20–25% in latency and 15–23% in cache hits. Paper [20] investigates a method for optimizing content caching solutions in the IoV to minimize content loading latency for vehicles, which is based on vehicle-to-vehicle communication. We propose a delay-aware content caching (DCC) algorithm in the IoV, which consists of vehicle associations, content caching and optimization of pre-caching solutions. First, the delay-aware vehicle associations (DVAs) algorithm is proposed to optimize vehicle associations. Hence, based on the results of vehicle associations, content caching decisions are optimized in two network scenarios according to the availability of handover vehicles.

In paper [21], efficient task offloading schemes in vehicular boundary computational networks are studied. Vehicles optimally perform offload time choices and communication, and computational resources, vehicle mobility and maximum task latency are taken into account. To minimize system costs, including the cost of required communication and computational resources, the offloading scheme is first analyzed in a scenario with independent MEC servers. The offloading tasks are handled by MEC servers deployed at the access point (AP) independently of each other. A mobility-aware task offloading scheme is proposed. Then, in a collaborative MEC server scenario, MEC servers can further offload collected offloading tasks to neighboring servers at the next AP in the direction of vehicle traffic. A location-based offloading scheme is proposed. Both scenarios mainly consider the trade-off between task latency and required communication and computational resources. The numerical results show that the proposed schemes can effectively reduce the system cost while the latency constraints are met.

A detailed review of the application of the cluster approach to solving problems in VANETs was given in [22–24]. Separate clustering solutions were obtained in [25–27]. In one study [25], the FOREL clustering algorithm was applied to minimize the data delivery delay. The delay component was considered, which was characterized by the propagation time of electromagnetic or optical signals in a dynamic system. Optical cable communication lines were considered as the propagation medium. However, this architecture did not take into account the problems associated with the fact that cable lines are laid along existing roads, which cannot always be specified by a straight-line segment. In [26], the authors proposed the use of spectral clustering. The objects of the study were vehicle-to-infrastructure and vehicle-to-vehicle links in a multi-lane highway scenario, where an RSU network provided coverage. A mechanism for the optimal selection of TCs that have a quality connection with RSUs was proposed, which allowed unloading of TCs with a low signal-to-noise ratio. Numerical simulation results were given, demonstrating a significant improvement in the overall performance of the dynamic system. The authors of [27] studied a city model that focused on the effect of preemptive caching on unclustered and clustered schemes. It was shown that the clustered caching scheme was the more efficient as more RSUs were included in the cluster. In this regard, it is important to test whether clustering will minimize the overall computational latency of the edge devices.

Below, in Table 1, we summarize the approaches to solving this class of problems.

Table 1. Overview of the existing solutions.

Papers	Time Loss	Data Loss	Clustering	Caching	Energy
[8,9]	+				+
[10]	+			+	+
[11–14]		+		+	
[15]	+			+	
[16]		+	+	+	
[17,18]		+		+	
[19,20]	+			+	
[21]	+	+			
[22–24]			+		
[25,26]	+		+		
[27]			+	+	
this paper		+	+	+	+

As can be seen from Table 1, the reviewed works provide a partial analysis of the parameters and constraints arising in the design of the VANET system, so a comprehensive approach to the analysis of parameters the authors consider relevant.

3. System Model

Suppose the city model is defined by the Manhattan mobility model in a rectangular metric (L_1-metric). The distance between two points with coordinates $A(x_1;y_1)$ and $B(x_2;y_2)$ in this model was calculated using the Formula (1)

$$d(A,B) = |x_2 - x_1| + |y_2 - y_1|. \quad (1)$$

The city highway system forms a grid with horizontal and vertical elements. Two-way traffic is organized on each street. In addition, we assume that the mobility model is considered in 2D. This assumption is considered by the authors as a significant limitation, because the model will seriously change when considering the third (vertical) component, as it will be necessary to additionally consider the reflection of signals from buildings and moving cars. Here, we will limit ourselves to the consideration of the two-dimensional model in L_1-metric, additionally assuming that the difference between the height of the car and the RSU is zero. The study of three-dimensional models is the direction of further research. A set of V OBU-equipped vehicles travel through the city, where each OBU selects the shortest route to travel from the start point to a predetermined destination. Before starting, each OBU generates its own individual route.

Consider a set of S RSUs located in the middle of each segment between two intersections. Each RSU $s \in S$ is equipped with a limited Z_s size cache, which is used to cache processed data. We considered that each RSU has a coverage area in the form of a circle with a diameter of L_s.

Vehicle speeds are a stream of independent, equally distributed random variables. These velocities are distributed over a truncated normal distribution law. Let the speed u of each OBU in a section of the road (within the range of one RSU) be in the range from pre-set minimum to maximum speed, that is, $u_{min} \leq u \leq u_{max}$.

The velocity distribution density function u with average μ and σ^2 dispersion is given by Formula (2) in [28]:

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

where $erf()$ is the normal distribution function or Laplace function.

The OBU interacts with the RSU through the exchange of data blocks, for example, information about the congestion of a section of the road on the route. Let each OBU and each RSU form information as a vector M of uncorrelated data elements $M = \{1, 2, \ldots, m\}$. Assume that all data items have the same C byte size. Each RSU serves the connected OBU at α_s bytes per second.

For numerical modeling, we assumed that the random variable specifying the probability that the cell of tuple M can be requested by OBU is given by a Zipf distribution, calculated using Formula (3).

$$p(i) = \frac{i^{-A}}{\sum_{k=1}^{m} k^{-A}}, \quad (3)$$

where A is the distribution asymmetry coefficient and corresponds to the loading degree of the vehicle system. The average value of the amount of query to be processed by the OBU–RSU system can be calculated as a mathematical expectation of a randomly generated quantity, and is given by Equation (4).

$$D(i) = \sum_{k=1}^{m} C \cdot p(i). \quad (4)$$

The location and number of OBUs involved in the model in question vary dynamically over time. Therefore, it is necessary to update the data in the RSU with some frequency, requesting it from the MEC server. Let the processing time of the request from a particular

OBU over the data block i be $t_i^{(0)}$, if the data in that memory location are current, and $t_i^{(0)} + t_i^{(1)}$, where $t_i^{(1)}$ is the time required to update the out-of-date data, if the data are irrelevant and need to be updated.

Assume that each RSU can serve the OBU sequentially. Each road infrastructure device can simultaneously process up to two requests from the OBU. To support V2X applications, IEEE developed the IEEE 802.11p standard in 2010. These systems do not meet the very low data latency requirements of today's V2X applications. The new IEEE 802.11bd standard has an important advantage in the form of channel connection technology. This technique allows data to be transmitted over two adjacent channels simultaneously. This increases the data rate and can reduce delays and packet loss [29], based on the effectiveness of standards from IEEE 802.11.

If the OBU leaves the RSU coverage area, the interaction is terminated. The OBU can connect to the next RSU if it is free. It is believed that the RSU, when finished with any OBU, can start servicing the next instantly. Blocks of data that were not fully processed in the information interaction with the previous OBU are lost.

If the OBU speed on a section of the road served by the RSU s is u_i m/s, and the coverage area of this section is L_s meters long, then the maximum number of requests that the RSU from a particular OBU can process is calculated using Formula (5).

$$k = \left\lfloor \frac{L_s}{u_i \cdot \alpha_s} \right\rfloor. \tag{5}$$

Let us introduce a loss function for query processing in the OBU–RSU system. 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 calculating using Equation (6).

$$V(i) = \begin{cases} D(i) - C \cdot k,\ C \cdot k < D(i) \\ 0,\ C \cdot k \geq D(i) \end{cases} \tag{6}$$

If some of the requested information needs to be updated, then the total interaction time is given by Equation (7):

$$\sum_{i=1}^{k} \left(t_i^{(0)} + \gamma \cdot t_i^{(1)} \right) \leq \frac{L_s}{u}, \tag{7}$$

where $\gamma = 1$ if the data are outdated and $\gamma = 0$ if the data do not require updating for each pair of vehicle RSUs.

The diagram of solving the task is shown in Figure 1.

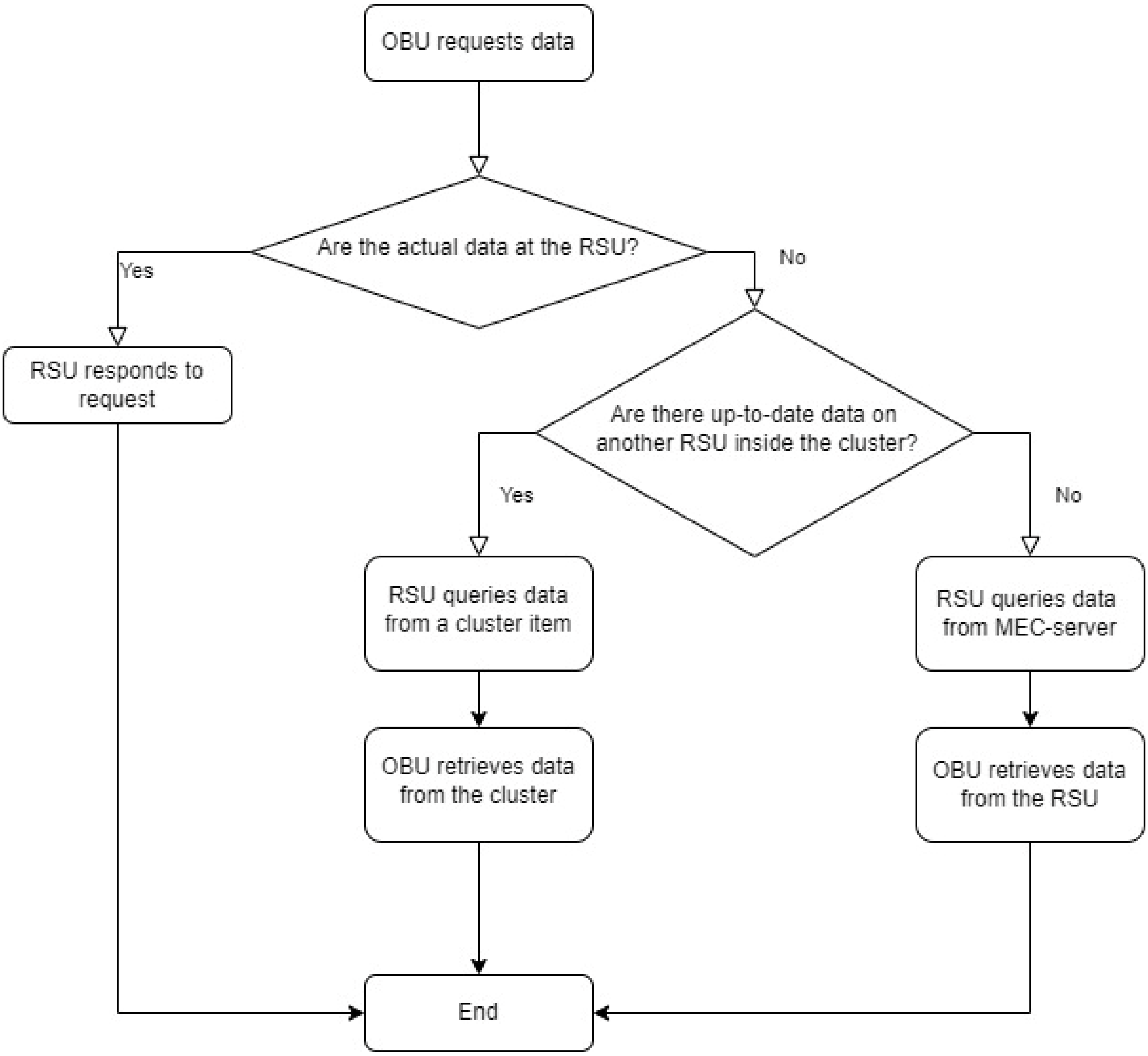

Figure 1. Scheme for solving the task.

4. Delay and Energy Consumption Analysis

4.1. Analysis of Delays

Consider the movement of vehicles in the urban transport network as a dynamic system. Let the urban network be a system of intersecting roads at right angles, as shown in Figure 2. The RSUs are located in the middle of each road section between two neighboring intersections (the symbol  is a graphic notation for RSUs), and we assumed that the coverage area of each RSU was the entire road section between the two intersections.

During the simulation, a random stream of vehicles is generated, which is processed by the system. If the RSU is free, the vehicle begins to interact with it. If there is no contact (RSU is busy or not functioning), the vehicle moves according to the previously planned route with constant speed on this section of the road and searches for a free RSU. After finding a free RSU, the vehicle begins interaction with it.

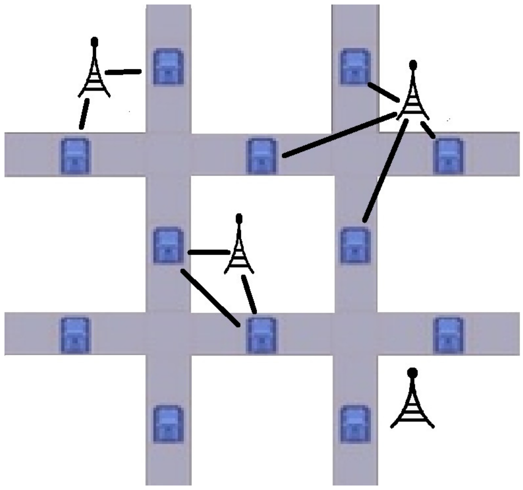

Figure 2. Urban network model.

Before optimizing the network, choosing the optimal location of the RSU, dynamically rearranging the routes of vehicles in accordance with the road situation, changing the speed of vehicles depending on the speed of traffic, etc., you must first determine whether the created system can work under the given conditions.

Let us make some additional simplifications to build a workable model. Let each vehicle v_i request a predetermined number of information blocks $k^{v_i} \leq m$ in the range from $k_1^{v_i}$ to $k_2^{v_i}$, denoted by $M_{v_i} = \left\{m_1^{v_i}, m_2^{v_i}, \ldots, m_{k^{v_i}}^{v_i}\right\}$. If, when moving along the route, the requested block of data is processed and the response from RSU is received, then we consider that the number of unprocessed blocks of information is reduced by one. This approach will further allow us to interpret the received data as information on the traffic situation at the nearest intersection in order to select the optimal vehicle speed or realign the current route to reduce the traffic intensity at a busy intersection.

When analyzing the network bandwidth, we assume that the cache size Z_s of each RSU is the maximum amount of C_m information processed. This allows monitoring of the memory overflow and selective deletion of data already stored on the RSU. At the same time, it is worth paying attention to the following important aspect: if n vehicles consistently turn to one data block, then, interpreting the information as a road situation at the intersection, the OBUs of these vehicles will send all vehicles to this intersection, forming congestion. Therefore, it is necessary to introduce some restrictions F on the number of calls l to each block of information before it is forcibly updated. We considered $l \leq F$. In this case, we additionally introduced the restriction T on the "lifetime" t_{m_i} of the m_i data, considering $t_{m_i} < T$.

As an assessment of the effectiveness of the system, we calculated the percentage of unprocessed requests generated at the initial stage by the vehicles.

This potentially led to a decrease in the efficiency of traffic organization. To reduce the number of unprocessed queries, we took a cluster approach to solving the problem in question.

The use of clustering in problem solving in VANET caused the authors to conduct simulations using a cluster caching scheme. Suppose that each RSU group can be a cluster with a memory sharing property. Otherwise, the cluster elements communicate with each other to serve any connected OBU in the coverage area of the cluster with a time delay less than that of the individual element, due to the increased amount of computing power and the amount of local cache. As an example, consider the RSU cluster $\{s_1, s_2, s_3\}$, as shown in Figure 3.

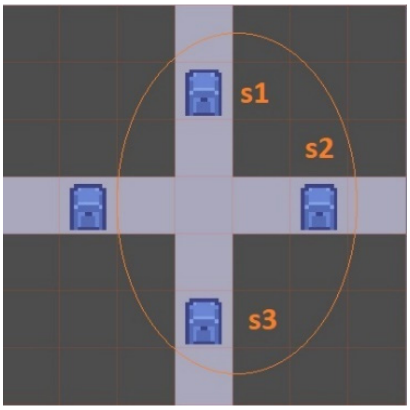

Figure 3. Clustering scheme for RSU.

Additionally, note that the internal communication of the elements of one cluster is controlled by base stations (BS), also placed in Figure 3. In this case, the connection of RSU-BS can be carried out both via Wi-Fi, and, for example, using fiber-optic connection.

When OBU v_i connected to RSU s_1 requests a data item m_j, RSU s_1 checks its local cache. If the requested data item is found, it is delivered after $t_i^{(0)}$ seconds. Otherwise, RSU s_1 checks the cache of other cluster elements, that is, s_2 and s_3, to deliver the requested data element in τ seconds. If the requested file is not found in any of the cluster elements, the requested data element is delivered to the OBU in $t_i^{(1)}$ seconds. Let us perform clustering by merging RSUs in two ways: by pairs (Figure 4a) and by fours (Figure 4b).

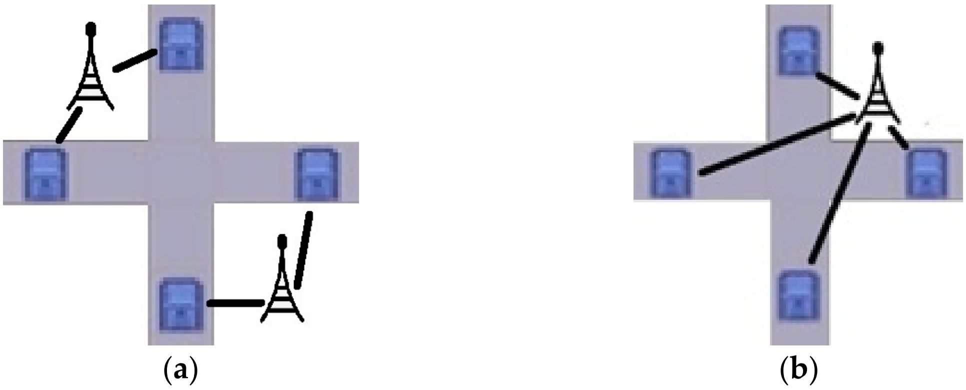

Figure 4. (**a**) Clustering scheme for RSU. Two road-side units combined in one cluster; (**b**) clustering scheme for RSU. Four road-side units combined in one cluster.

4.2. Energy Consumption

An important aspect of the solution of the problem under consideration is the value of energy costs to maintain the functioning of the OBU–RSU system. It is worth assuming that with the advent of clusters, the number of parallel requests between local edge devices may significantly increase. This is due to an increase in the number of computational tasks solved on the RSU. Data exchange in the OBU–RSU system, as well as RSU–MEC interactions, will be added to the interaction within the cluster to which the RSU belongs. At the same time, we should expect an increase in the number of calculations and the value of energy costs for maintenance of the info-communication network.

As before, we assumed that each OBU had M independent computational tasks to be performed. We considered a tuple $\{D_{ij}, c_{ij}, time_{ij}\}$ to solve each jth task by the ith OBU-RSU pair, where D_{ij} is calculated using Formula (4), c_{ij} is the total number of CPU cycles needed to solve a particular jth task and $time_{ij}$ is the allowed time to solve the jth task for the ith OBU-RSU pair.

We can distinguish three types of computational task executions: local, cluster and remote. In each of these cases, energy costs will be different.

Consider the local execution of the task. In this case, all interactions take place directly in an OBU–RSU pair; that is, all necessary data are already in the RSU cache. In this case, the consumed energy can be calculated using Formula (8):

$$E_{ij}^{L} = \sigma_i \cdot c_{ij}, \quad (8)$$

where σ_i is the coefficient, which denotes the energy consumed per processor cycle, and depends on the architecture of the RSU computing system. We took $\sigma_i = 10^{-11}(f_i)^2$, where f_i is the computation capability (number of processor cycles per second). In [30], information about the relationship between processor frequency and the number of cycles per unit of energy consumption is given; the value f_i is called the computing frequency.

Separately, we considered cluster and remote solutions of computational problems and obtained the corresponding formulas.

If we consider the interaction in the OBU–RSU system without cluster partitioning, the total energy expenditure for solving each jth task by the ith pair is calculated using Formula (9):

$$E_{ij}^{nC} = \sigma_i \cdot c_{ij} + \left(p_i^A \cdot T_{ij}^{down} + p_i^D \cdot T_{ij}^{off_c}\right) \cdot \alpha_{ij}, \quad (9)$$

where p_i^A is the receiving power of the RSU computing device, T_{ij}^{down} is the RSU receiving time, p_i^D is the MEC transmission power, $T_{ij}^{off_c}$ is the MEC transmission time and α_{ij} is a number taking the value "1" if the requested data are stored in the RSU cache and "0" if the data require updating or are absent.

If we consider the interaction in the OBU–RSU system using the partitioning of RSUs into clusters with n elements in each, in order to minimize data loss during OBU maintenance, the total energy consumption for each jth task by the ith pair could be calculated using Formula (10):

$$E_{ij}^{C} = \sigma_i \cdot c_{ij} + nE_0 + \left(p_i^A \cdot T_{ij}^{down} + p_i^D \cdot T_{ij}^{off_c}\right) \cdot \alpha_{ij}, \quad (10)$$

where E_0 is the energy consumed to perform one request for data availability inside the cluster, p_i^A is the receiving power of the RSU computing device, T_{ij}^{down} is the RSU receiving time, T_{ij}^{down} is the RSU receiving time, p_i^D is the MEC transmission power, $T_{ij}^{off_c}$ is the MEC transmission time, and α_{ij} is a number taking the value "1" if the requested data are stored in the RSU cache and "0" if the data require updating or are missing.

The total resource cost of each jth computational task of the ith OBU-RSU pair in terms of energy consumption can be defined by Equation (11):

$$E = \sum_{i,j} \left((1-\beta)\, E_{ij}^{nC} + \beta E_{ij}^{C} \right), \quad (11)$$

where $\beta = 0$ if the clusterless model is chosen, and $\beta = 1$ if clustering is used.

Due to the fact that in [31,32], it was shown that with the enlargement of clusters, reducing their number, the percentage of serviced OBUs increased, let us introduce the concept of average energy consumption per cluster $\frac{E}{n}$, where n is the number of cluster elements, that is, the value characterizing the average energy per cluster.

Combining the results of the analysis of the optimization problem in Section 4.1, we formulated the extreme problem using Equation (12).

$$\begin{cases} \underset{n}{\text{minimize}}\ {}^{E}/n \\ \underset{m}{\text{minimize}} \frac{\sum_i D(i)}{\sum_i V(i)} \\ D(i) \le \frac{L_S}{u_i} \cdot \alpha_s \end{cases} \quad (12)$$

5. Simulation Algorithm

Numerical modeling was performed using the Python programming language. The random flow of vehicles with a given distribution was generated using the cross-platform Godot Engine. The code of the authors' program of the proposed method is given in [33]. Using your own program code allows you to manage program functionality independently, have a transparent structure, utilize open-source component code and minimize scripting delays by minimizing the often-superfluous functions of known program modules. The developed program was highly specialized, and therefore allowed the most effective solution of the formulated problem. We considered the road network shown in Figure 5. We assumed in the simulation that the starting position, from which all vehicles start, was located in the center of the map (in the figure, this place is marked with a sign). This position allowed us to visually track the movement of vehicles and changes in the number of requests sent. RSUs are marked on the scheme by blue blocks (as in Figure 2,). The end point of the route of each vehicle was generated separately and was located on the border of the area shown in Figure 5.

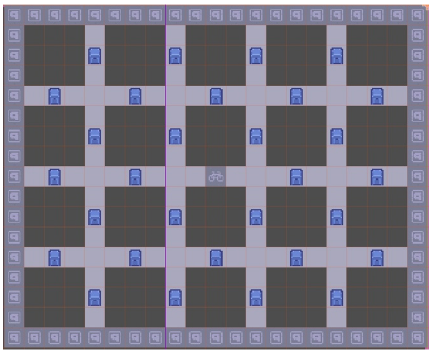

Figure 5. Simulated road network.

A brief diagram of the algorithm is shown below (Algorithm 1). At this stage, the authors do not aim to obtain an effective algorithm for solving the problem, taking into account computational or temporal complexity. Here, we only analyze the model; the search for efficient solutions is the direction of further research.

Algorithm 1: Cluster-based V2X algorithm with a shared cache

Input : $V,\ m,\ L_S,\ f,\ t_i^{(0)},\ t_i^{(1)},\ F,\ T,\ p_i^A,\ p_i^D,\ c_{ij}.$
Generate a random flow of vehicles
Make clusters
while V in road:
for i in range(V):
    Generating a set of requested data
    Compute D(i)
    for j in range(M):
        if no_clusters:
Interaction of the vehicle with RSU
        Compute V(i)
        Compute E(i)
Sum(E(i)) += SumE; Sum(V(i)) += SumV
        If not no_clusters:
Interaction of the vehicle with RSU
        Compute VC(i)
        Compute EC(i)
Sum(E(i)) += SumEC; Sum(VC(i)) += SumVC
print (SumE-SumEC, SumV-SumVC)

Consider a realistic urban scenario whose configuration is determined by the user, in which the system has a varying number of vehicles from low density (10 cars) to high traffic density (800 cars), from 4 to 64 RSUs depending on the city plan, a dynamically changing number of base stations depending on the number of clusters and 1 cloud server. The length of one block is 0.2 km. In the simulation scenario, for simplicity, we consider a single-lane stretch of road with multiple intersections, as shown in Figure 5. For V2I communication, the range of each RSU is 200 m, and a vehicle can send a content request as soon as it connects to one of the RSUs. It is assumed that each RSU covers the road and has some overlap in radio coverage with the RSU located in the middle of the next block.

We used the following parameters (Table 2) to perform numerical simulations. In our numerical study, we attempted to be as close to the IEEE 802.11 p/bd [34] as possible, in particular for the radio and antenna parameters. The "free" parameters have been chosen such that they reflect the most interesting simulation conditions. The choice of simulation parameters in this paper is conditional; researchers can independently perform calculations using the program developed by the authors and posted on GitHub [33].

Table 2. Parameters and values.

	Parameter	Value
Set of vehicles (pcs.)	V	800
Set of RSUs (pcs.)	S	64
Maximum cache memory size sth RSU (MB)	Z_s	2048
Diameter of the coverage area of the sth RSU (m)	L_s	200
Data block for exchange in the OBU-RSU system	$M = \{1, 2, \ldots, m\}$	20
Packet size (Byte)	C	255
Data rate (Mb/s)	α_s	3375 (27 Mbit/s)
Minimum speed of vehicles on the road (km/h)	$u_{\min}$	20
Maximum speed of vehicles on the road section (km/h)	$u_{\max}$	60
Average processing time per request (s)	$t_i^{(0)}$	0.1
Time required to update obsolete data (s)	$t_i^{(1)}$	1
Maximum number of accesses to the cell before updating (pcs)	F	3
Data update period in the cell (s)	T	10
Minimum number of requested data cells (pcs)	$k_1^{v_i}$	1
Maximum number of requested data cells (pcs)	$k_2^{v_i}$	12
Cluster task energy (watt/s)	E_0	$\in$[20, 80]
RSSI (mW)	p_i^A	$5{\cdot}10^{-8}$ (-73 dBm)
Transmit power (mW)	p_i^D	200 (23 dBm)
Frequency band (GHz)	f	5.9

6. Numerical Results and Discussion

We sequentially conducted numerical simulations in different variants of the initial conditions: without cluster division with one and two connected devices to one RSU, with cluster division of two elements in a block with one and two connected devices to one RSU, and with cluster division of four elements in a block with one and two connected devices to one RSU. In each option, simulations were run 10 times based on the parameters shown in Table 2. The numerical simulation results showed that when the vehicle flow was generated 11 or more times, the percentage of unprocessed requests coming from the OBUs changed only slightly. A 10-model simulation was sufficient to obtain the necessary accuracy in evaluating the efficiency of the approach under consideration.

The average percentage of unprocessed requests (the calculation was performed using the arithmetic mean formula) was 48.23%. The percentage of unprocessed requests was quite high; therefore, a significant portion of vehicles moved uncontrollably, not correcting their route depending on the road situation. This potentially led to a decrease in the efficiency of road traffic organization. To reduce the number of unprocessed requests, we applied the cluster approach to solve the problem in question.

The average percentage of unprocessed requests in the case of using the cluster approach was 32.16% and 22.54%, respectively, indicating a reduction in the total processing time of tasks on boundary devices compared to the traditional model considered earlier.

Similarly, numerical simulations for two sub-connections were performed, and the average number of losses was calculated. The average values of losses were, respectively, 18.01%, 5.18% and 1.49%. As a result, the use of a more modern type of communication allowed a significant gain for each of the considered systems of management organization of the RCS system. Figure 6 shows the results of comparative modeling of all considered models.

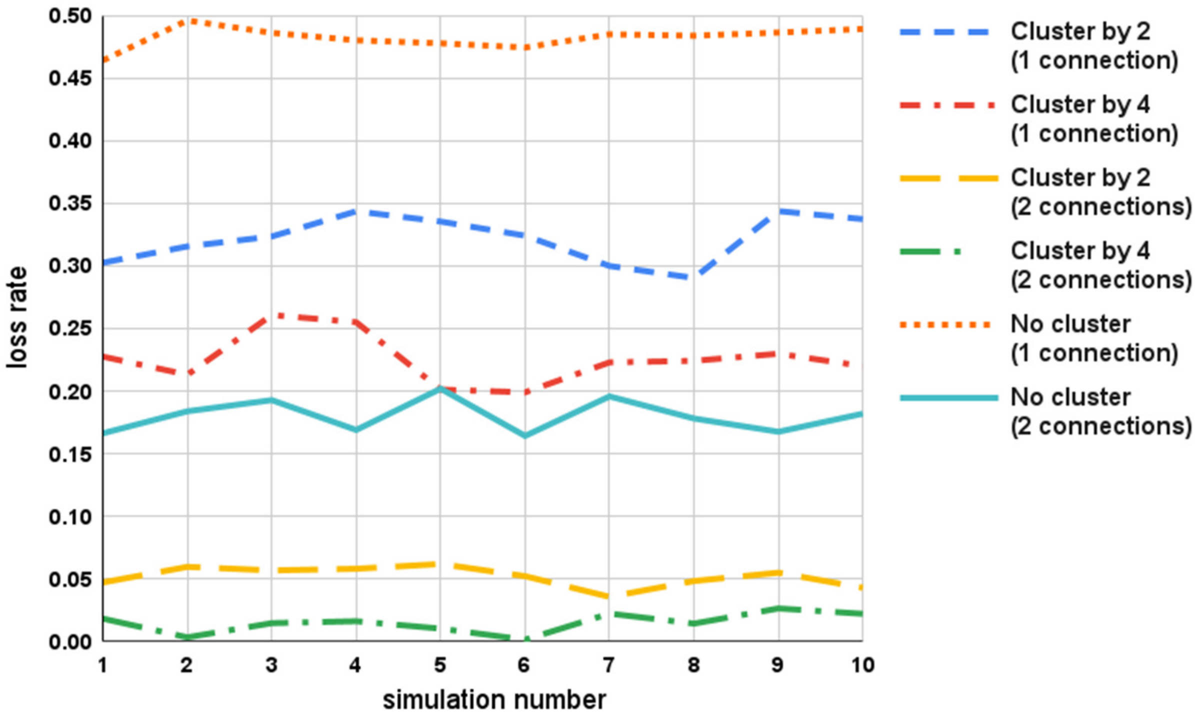

Figure 6. Average percentage of losses at 10 iterations.

The horizontal axis shows the sequence number of the iteration of numerical modelling of vehicle flow, and the vertical axis shows the percentage of unprocessed queries in the OBU–RSU system interaction. As can be seen from the presented graphs, combining RSUs into clusters allows a significant gain in the percentage of successfully processed queries. This result confirmed the conclusions drawn above. In this case, the more elements that are combined into one cluster, the higher the efficiency of system management.

It should be noted that the technical characteristics of modern means of communication do not allow the combining of all the RSUs into one overall structure, due to which it is necessary to utilize cluster division in accordance with the spatial location of the devices of edge computing. The presented graphs show that data transmission over two adjacent channels simultaneously allows us to significantly reduce the percentage of unprocessed requests generated by OBUs. In this case, the use of the cluster approach significantly reduced the value of losses in each of the cases under consideration.

Summing up the intermediate results, the most efficient in terms of data loss was the model with two-channel connection and partitioning of RSU into clusters of four elements, and the most inefficient was the classic single-channel scheme.

For qualitative comparison of the results, we calculated the average percentage of data block losses over 10 generations of a random flow of vehicles, and the sampling variance of these losses. A comparative analysis of the parameters under different conditions is shown in Table 3.

Table 3. The results of numerical simulation of vehicle flow in the road network.

Model	Average Percentage	Sampling Variance
No cluster, 1 connection	48.23%	1.29
Cluster by 2, 1 connection	32.16%	2.35
Cluster by 4, 1 connection	22.54%	4.66
No cluster, 2 connections	18.01%	2.10
Cluster by 2, 2 connections	5.18%	1.74
Cluster by 4, 2 connections	1.49%	1.08

It may be noted that using any of the considered clustering schemes allows us to reduce the percentage value of losses by an order of magnitude. In this regard, it is worth taking a comprehensive approach to analyze and build a model for its effective functioning.

Additionally, it should be noted that the value of sampling variance, which characterizes the dispersion of values relative to the average, is much smaller when using standard IEEE 802.11bd. This suggests that the results obtained using the abovementioned scheme are more resistant to external factors. The use of the scheme with two channels of simultaneous OBU and RSU interaction is more stable for the implementation of the interaction of elements of the dynamic system.

Although the use of new upgraded communication network standards and clustering reduces the amount of data loss when modeling the OBU–RSU system, the structural and functional complexity leads to increased power consumption.

Due to the fact that energy costs are highly dependent on the conditions under which the system functions (environmental impact, energy costs and associated costs), let us analyze energy consumption in relative terms. Let us assume that the cost of maintaining the system in a no-cluster scheme with one connection is 1. The remaining results are shown in Table 4.

Table 4. Average energy consumption in all schemes under consideration.

Model	Average Energy Consumption
No cluster, 1 connection	1
Cluster by 2, 1 connection	1.73
Cluster by 4, 1 connection	2.85
No cluster, 2 connection	1.79
Cluster by 2, 2 connections	2.66
Cluster by 4, 2 connections	3.97

The results are presented as graphs. The horizontal axis shows the percentage of losses and the vertical axis shows the relative value of energy costs. As can be seen in Figure 7, the use of a dual-channel connection to the RSU requires an increased load on the power system compared to the single-channel connection.

We then carried out an analysis of energy consumption, taking into account the cluster partitioning. On the horizontal axis, we plotted the number of elements in the cluster, and on the vertical axis, we plotted the relative value of energy costs. The trend line corresponded to the exponential growth in energy costs with the increase in elements in the cluster (Figure 8).

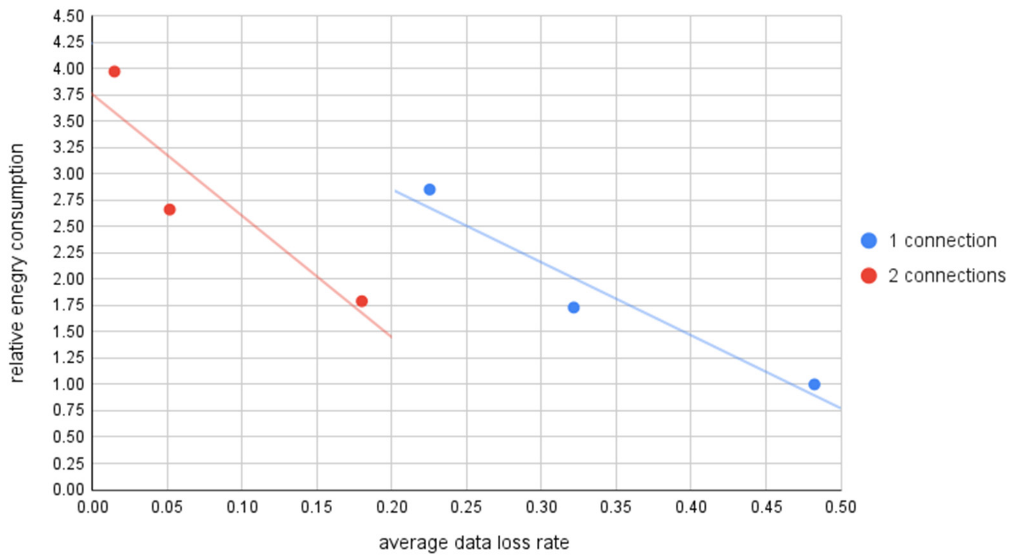

Figure 7. Comparison of average data loss and relative energy consumption.

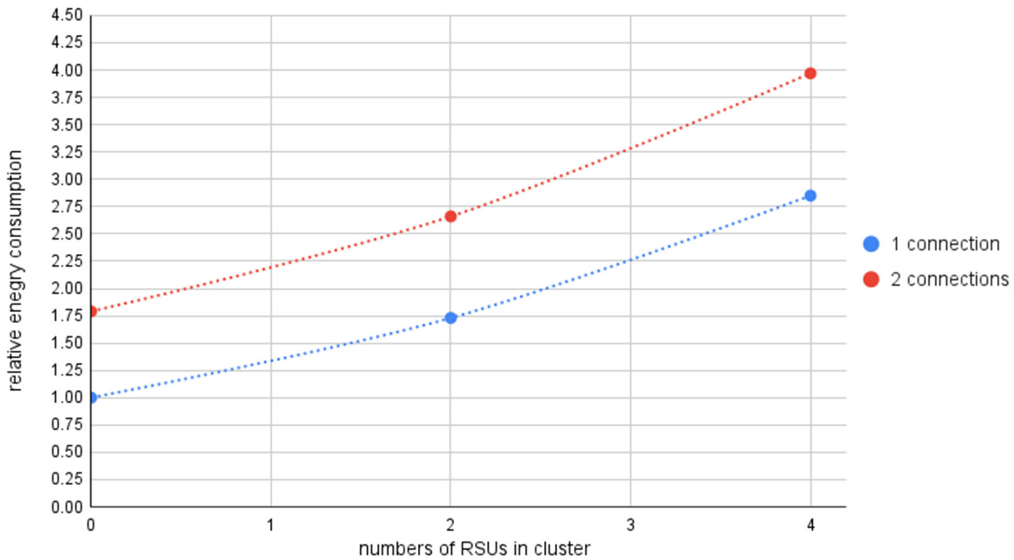

Figure 8. Comparison of the number of RSUs in a cluster relative to relative power consumption.

Based on the above, it is necessary to find a balance in the number of clusters, based on the tasks assigned to the system.

In recent years, V2X communication technology in intelligent transportation systems (ITSs) has gained tremendous interest because it can potentially meet today's ITS needs, reduce the load on the transportation network infrastructure, improve road safety, ensure driving comfort, provide an alternative emergency communication system in case of natural disasters and provide a high degree of vehicle automation. To design vehicles with high levels of automation (eyes-off and mind-off), the vehicle must be able to process large amounts of real-time data related to the recognition of nearby objects, accidents, navigation tasks, etc. However, the computing resources of some existing vehicles are too limited to run such applications. At the same time, any disruption of system operation and stability can endanger people and machinery on the road, as well as create high costs for system recovery and support, which places the issues of reliability of transmitted data and ensuring high ITS performance in the foreground. Based on the analysis presented in this article, we can conclude that the use of clustering and more modern ways of organizing the connection of system elements will optimize and accelerate the operation of intelligent transport systems.

7. Conclusions

The proliferation of modern technologies in road transport, such as connected vehicles and autonomous vehicles, requires the solution of a number of problems to improve the efficiency of interactions between vehicles and elements of the stationary infrastructure. The problem is that with high traffic volume, the capacity of the information processing and transmission system does not always ensure traffic safety.

There are two main ways to solve this problem: first, by improving the technical characteristics of the equipment used (in particular, by building networks based on the new 5G mobile communication standard) and second, by improving the organizational and algorithmic support of VANET functioning.

This article presents the results of a numerical simulation confirming the high efficiency of the use of clustering in solving the problems under consideration, and the evaluation of the efficiency of energy consumption of the system.

In this case, the emergence of new technical solutions related to the implementation of the IEEE 802.11bd industrial standard, which implies dual-channel operation, can significantly improve the results. In particular, it was shown that the transition to the use of dual-channel data exchange devices gives a greater gain in performance compared to clustering on single-channel devices. At the same time, there is a significant increase in energy costs. Therefore, using a better circuit is worthwhile, adjusting its parameters to the specific task to be solved. New technical solutions provide new opportunities for the formation of information–algorithmic support of V2X systems and may lead to the need to adjust their functioning schemes.

The problem of synthesizing a loss-optimal system becomes less definite, because branching occurs in its solution. Depending on the numerical values of the parameters with which the simulated system is described, it may be more effective to use both two-channel devices and clustering.

Summarizing all of the above, we can note a number of advantages of the considered approach: (1) the considered scheme allows a significant reduction in the percentage of unprocessed requests coming from OBUs to RSUs when using clustering and more modern types of connections; (2) the use of dual-channel connection will not only greatly reduce data loss, but also reduce the relative power consumption compared to a single-channel connection. However, it is necessary to note the significant disadvantages of the method, which consist of the limitations of the considered mobility model: (1) the choice of rectangular metric; (2) multiply increasing the complexity of the model architecture during the transition to three-dimensional space; (3) the static structure of the model.

The direction of further research is to analyze the optimal placement of RSUs, taking into account the road conditions, dynamic restructuring of the route of the vehicle based on the information received from the RSU and the analysis of the optimal partitioning of RSUs into clusters to maximize the capacity of the transport network, taking into account the technical limitations imposed on the network equipment.

The implementation of the proposed approach, developed by the authors, is freely available on GitHub [33].

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

Funding: This work was supported by the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R239), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R239), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

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

References

1. Cisco White Paper. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022*; Cisco: San Jose, CA, USA, 2019; 33p.
2. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile Edge Computing: A Survey. *IEEE Internet Things J.* **2018**, *5*, 450–465. [CrossRef]
3. Brehon-Grataloup, L.; Kacimi, R.; Beylot, A.L. Mobile edge computing for V2X architectures and applications: A survey. *Comput. Netw.* **2022**, *206*, 108797. [CrossRef]
4. Vladyko, A.; Khakimov, A.; Muthanna, A.; Ateya, A.A.; Koucheryavy, A. Distributed Edge Computing to Assist Ultra-LowLatency VANET Applications. *Future Internet* **2019**, *11*, 128. [CrossRef]
5. Vladyko, 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. [CrossRef]
6. De Oliveira, L.F.P.; Manera, L.T.; Luz, P.D.G.D. Development of a Smart Traffic Light Control System with Real-Time Monitoring. *IEEE Internet Things J.* **2020**, *8*, 3384–3393. [CrossRef]
7. Dziyauddin, R.A.; Niyato, D.; Luong, N.C.; Izhar, M.A.M.; Hadhari, M.; Daud, S. Computation Offloading and Content Caching Delivery in Vehicular Edge Computing: A Survey. *Comput. Netw.* **2021**, *197*, 108228. [CrossRef]
8. You, C.; Huang, K.; Chae, H.; Kim, B.H. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1397–1411. [CrossRef]
9. Ren, J.; Yu, G.; He, Y.; Li, G.Y. Collaborative Cloud and Edge Computing for Latency Minimization. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5031–5044. [CrossRef]
10. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE ACM Trans. Netw.* **2016**, *24*, 2795–2808. [CrossRef]
11. Bernardini, C.; Silverston, T.; Festor, O. MPC: Popularity-based caching strategy for content centric networks. In Proceedings of the International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 3619–3623. [CrossRef]
12. Perabathini, B.; Baştuğ, E.; Kountouris, M.; Debbah, M.; Conte, A. Caching at the edge: A green perspective for 5G networks. In Proceedings of the International Conference on Communication Workshop (ICCW), London, UK, 8–12 June 2015; pp. 2830–2835. [CrossRef]
13. Mahmood, A.; Casetti, C.; Chiasserini, C.F.; Giaccone, P.; Harri, J. Mobility-aware edge caching for connected cars. In Proceedings of the 12th Annual Conference on Wireless On-Demand Network Systems and Services (WONS), Cortina d'Ampezzo, Italy, 20–22 January 2016; pp. 1–8.
14. Garetto, M.; Leonardi, E.; Traverso, S. Efficient analysis of caching strategies under dynamic content popularity. In Proceedings of the Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 2263–2271. [CrossRef]
15. Poularakis, K.; Iosifidis, G.; Sourlas, V.; Tassiulas, L. Exploiting Caching and Multicast for 5G Wireless Networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 2995–3007. [CrossRef]
16. Golrezaei, N.; Molisch, A.F.; Dimakis, A.G.; Caire, G. Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution. *IEEE Commun. Mag.* **2013**, *51*, 142–149. [CrossRef]
17. Gad, M.E.; Hosny, S.; Mokhtar, B.; El-Sherif, A.A. Hierarchical Proactive Caching for Vehicular Ad Hoc Networks. In Proceedings of the 2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 28–30 October 2019; pp. 150–153. [CrossRef]
18. Abani, N.; Braun, T.; Gerla, M. Proactive caching with mobility prediction under uncertainty in information-centric networks. In Proceedings of the 4th ACM Conference on Information-Centric Networking, Berlin, Germany, 26–28 September 2017; pp. 88–97. [CrossRef]
19. Musa, S.S.; Zennaro, M.; Libsie, M.; Pietrosemoli, E. Mobility-Aware Proactive Edge Caching Optimization Scheme in Information-Centric IoV Networks. *Sensors* **2022**, *22*, 1387. [CrossRef] [PubMed]
20. Huang, X.; Xu, K.; Chen, Q.; Zhang, J. Delay-Aware Caching in Internet-of-Vehicles Networks. *IEEE Internet Things J.* **2021**, *8*, 10911–10921. [CrossRef]
21. Yang, C.; Liu, Y.; Chen, X.; Zhong, W.; Xie, S. Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks. *IEEE Access* **2019**, *7*, 26652–26664. [CrossRef]
22. Cooper, C.; Franklin, D.; Ros, M.; Safaei, F.; Abolhasan, M. A Comparative Survey of VANET Clustering Techniques. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 657–681. [CrossRef]
23. Bali, R.S.; Kumar, N.; Rodrigues, J.J. Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. *Veh. Commun.* **2014**, *1*, 134–152. [CrossRef]
24. Khan, Z.; Koubaa, A.; Fang, S.; Lee, M.Y.; Muhammad, K. A Connectivity-Based Clustering Scheme for Intelligent Vehicles. *Appl. Sci.* **2021**, *11*, 2413. [CrossRef]

25. Paramonov, A.; Khayyat, M.; Chistova, N.; Muthanna, A.; Elgendy, I.A.; Koucheryavy, A.; El-Latif, A.A. An Efficient Method for Choosing Digital Cluster Size in Ultralow Latency Networks. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9188658. [CrossRef]
26. Luoto, P.; Bennis, M.; Pirinen, P.; Samarakoon, S.; Horneman, K.; Latva-aho, M. Vehicle Clustering for Improving Enhanced LTE-V2X Network Performance. In Proceedings of the European Conference on Networks and Communications (EuCNC), Oulu, Finland, 12–15 June 2017; pp. 1–5. [CrossRef]
27. AlNagar, Y.; Hosny, S.; El-Sherif, A.A. Proactive Caching for Vehicular Ad hoc Networks Using The City Model. In Proceedings of the Wireless Communications and Networking Conference Workshop (WCNCW), Marrakech, Morocco, 15–18 April 2019; pp. 1–7. [CrossRef]
28. Abuelenin, S.M.; Abul-Magd, A.Y. Empirical study of traffic velocity distribution and its effect on VANETs connectivity. In Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE), Vienna, Austria, 3–7 November 2014; pp. 391–395. [CrossRef]
29. Torgunakov, V.; Loginov, V.; Khorov, E. A Study of Channel Bonding in IEEE 802. *11bd Networks. IEEE Access* **2022**, *10*, 25514–25533. [CrossRef]
30. Zhu, Z.; Zhang, M.; Hao, W. Chapter 6—Artificial intelligence technology in the Internet of things. In *Intelligent Sensing and Communications for Internet of Everything*, 1st ed.; Zhu, Z., Chu, Z., Li, X., Eds.; Academic Press: London, UK, 2022; pp. 245–297. [CrossRef]
31. Plotnikov, P.; Vladyko, A. Minimizing delays in the interaction of edge devices using clustering in VANETs. *Proc. Telecommun. Univ.* **2022**, *8*, 6–13. [CrossRef]
32. Plotnikov, P.; Vladyko, A. Numerical Analysis of the Mathematical Model of a Cluster V2X-System. *Proc. Telecommun. Univ.* **2023**, *9*, 14–23. [CrossRef]
33. Plotnikov, P.; Tambovtsev, G.; Vladyko, A. VANET Ver. 1.0.0—Software Module for Modeling the Interaction of Edge Devices. 2022. Available online: https://github.com/quanuhs/VANET (accessed on 18 June 2023).
34. C2CCC White Paper. *Next Generation V2X—IEEE 802.11bd as Fully Backward Compatible Evolution of IEEE 802.11p*; CAR 2 CAR Communication Consortium: Braunschweig, Germany, 2023.