

Article

Microservice-Based Vehicular Network for Seamless and Ultra-Reliable Communications of Connected Vehicles

Mira M. Zarie ^{1,2}, Abdelhamied A. Ateya ^{1,3,*} , Mohammed S. Sayed ^{1,4} , Mohammed ElAffendi ³ 
and Mohammad Mahmoud Abdellatif ^{1,2} 

¹ Department of Electronics and Communications Engineering, Zagazig University, Zagazig 44519, Egypt; mira.mohsen@bue.edu.eg (M.M.Z.); mohammed.sayed@ejust.edu.eg (M.S.S.); mohammad.abdellatif@bue.edu.eg (M.M.A.)

² Department of Electrical Engineering, Faculty of Engineering, The British University in Egypt, Cairo 11837, Egypt

³ EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia; affendi@psu.edu.sa

⁴ Department of Electronics and Communication Engineering, Egypt-Japan University of Science and Technology, Alexandria 21934, Egypt

* Correspondence: aateya@psu.edu.sa

Abstract: The fifth-generation (5G) cellular infrastructure is expected to bring about the widespread use of connected vehicles. This technological progress marks the beginning of a new era in vehicular networks, which includes a range of different types and services of self-driving cars and the smooth sharing of information between vehicles. Connected vehicles have also been announced as a main use case of the sixth-generation (6G) cellular, with ultimate requirements beyond the 5G (B5G) and 6G eras. These networks require full coverage, extremely high reliability and availability, very low latency, and significant system adaptability. The significant specifications set for vehicular networks pose considerable design and development challenges. The goals of establishing a latency of 1 millisecond, effectively handling large amounts of data traffic, and facilitating high-speed mobility are of utmost importance. To address these difficulties and meet the demands of upcoming networks, e.g., 6G, it is necessary to improve the performance of vehicle networks by incorporating innovative technology into existing network structures. This work presents significant enhancements to vehicular networks to fulfill the demanding specifications by utilizing state-of-the-art technologies, including distributed edge computing, e.g., mobile edge computing (MEC) and fog computing, software-defined networking (SDN), and microservice. The work provides a novel vehicular network structure based on micro-services architecture that meets the requirements of 6G networks. The required offloading scheme is introduced, and a handover algorithm is presented to provide seamless communication over the network. Moreover, a migration scheme for migrating data between edge servers was developed. The work was evaluated in terms of latency, availability, and reliability. The results outperformed existing traditional approaches, demonstrating the potential of our approach to meet the demanding requirements of next-generation vehicular networks.

Keywords: distributed edge computing; software-defined networks; VANETs; 5G; fog computing; microservices



Citation: Zarie, M.M.; Ateya, A.A.; Sayed, M.S.; ElAffendi, M.; Abdellatif, M.M. Microservice-Based Vehicular Network for Seamless and Ultra-Reliable Communications of Connected Vehicles. *Future Internet* **2024**, *16*, 257. <https://doi.org/10.3390/fi16070257>

Academic Editors: Alessandro Raschella and Michael Mackay

Received: 24 June 2024

Revised: 10 July 2024

Accepted: 15 July 2024

Published: 19 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the evolution of wireless devices, the Internet and communication systems have evolved rapidly. Thus, next-generation networks (NGN), including sixth-generation (6G) networks, are expected to support novel applications and meet new requirements [1]. The vehicular network is a highly promising network that is expected to provide reliable communication infrastructure for vehicles. The introduction of the fifth-generation (5G) technology brings innovative solutions for vehicular communications, with significant

consequences for the transportation sector [2]. Connected vehicles, which serve as the foundation for intelligent transportation systems, are one of the most prominent applications of 5G technology and are expected to be a main part of the 6G networks. The upcoming 6G networks are expected to support ultra-high mobility; thus, enabling vehicular communication with the announced requirements of such networks is a big demand. This new paradigm, i.e., vehicular networks, includes a wide range of services to enable autonomous vehicles and facilitate data interchange between vehicles, infrastructure, and the cloud. Such systems promise increased safety, improved traffic management, and a better driving experience [3].

Forced by the recent advances in vehicles and embedded smart tools including wireless sensors, onboard computers, global positioning system (GPS), antenna, radar, and data storage solutions, vehicular networks have attracted academics and industry. Nowadays, the evolution of vehicles is to convert the traditional vehicles that supply basic services to smart vehicles that provide advanced services, achieving higher road safety and an easier life. VANT is a mobile ad hoc network (MANET) where vehicles act as mobile nodes or routers. This aims to acquire two types of communication: immediate communication between vehicles (V2V) and vehicle-to-everything (V2X), which provides communication between vehicles, roadside fixed infrastructure, and other network parts [4]. In VANT, each vehicle contains an onboard unit (OBU) that contains the radio transceiver to enable communication between vehicles and roadside units (RSUs).

According to the International Telecommunication Union (ITU), vehicular networks in the 6G (IMT2030) era must provide full coverage, ultra-high reliability, availability, ultra-low delay, and significant system flexibility. Specifically, delivering end-to-end latency as low as 1 millisecond, managing huge data flow, and supporting high-to-ultra-high mobility are significant problems that must be overcome [5,6]. Ultra-low latency communication (uRLLC) is required for numerous VANET scenarios and applications, including autonomous/safe driving. It is critical to think beyond the traditional structures of automotive networks to achieve these requirements [6]. Modern innovative technologies, including distributed edge computing, microservices, and software-defined networking (SDN), can greatly improve network performance. SDN provides a programmable network management architecture that improves flexibility and resource efficiency [7].

Distributed edge computing enables more efficient data processing and dissemination, improving the overall user experience in VANETs. It can speed up local data analytics and decision-making processes, allowing vehicles to respond instantly to changing conditions and events. In two ways, distributed edge computing can be deployed for vehicular networks: mobile edge computing (MEC) and fog computing [8]. Furthermore, MEC and fog enable offloading computationally intensive activities from vehicles, which frequently have limited processing capacity and energy resources. This offloading feature saves vehicle resources and ensures that programs that require a lot of computational power may still run smoothly and consistently [9].

MEC has arisen, providing cloud functionality at the edges of wireless networks. MEC significantly improves vehicular network performance by moving computational resources closer to the data source. This advancement allows vehicles to offload computationally heavy tasks to MEC servers with low latency. In the context of vehicular networks, MEC entails placing small-scale data centers or servers in important areas, such as base stations or RSUs, to execute functions that would normally need connection with a distant central cloud. This proximity dramatically reduces end-to-end latency, which is critical for real-time applications, including self-driving cars, collision avoidance, and high-definition mapping [10]. However, the complicated network environment and vehicles' intrinsic mobility pose issues typical offloading solutions frequently fail to meet.

Fog computing extends cloud computing to the network edge, distributing data processing instead of centralization. Fog computing in vehicular networks uses roadside equipment, traffic signals, and automobiles to process data closer to where it is created. This distributed strategy minimizes latency and boosts network resiliency [8]. Fog com-

puting aids traffic management, smart parking, emergency response, and environmental monitoring in automotive networks. It provides instant insights and reactions for time-sensitive applications by processing data locally. Fog computing's distributed nature makes vehicular networks more scalable, allowing them to handle the large data flow from linked vehicles and smart infrastructure. It lessens the load on centralized data centers, improving network efficiency and sustainability [11].

SDN is a networking framework that isolates the control and data planes, allowing for programmable administration. SDN allows VANETs to be more adaptable and dynamic, which is necessary for controlling vehicular networks' mobility. It allows network administrators to apply policies, optimize traffic flow, and improve security through a centralized controller monitoring the entire network. Furthermore, SDN centralizes control to optimize routing patterns and manage network resources, decreasing congestion and improving network performance. SDN's programmability rapidly adapts to changing network conditions, which is essential for dependable communication in dynamic vehicle contexts. This adaptability helps advanced applications, including real-time traffic management, emergency vehicle prioritization, and adaptive streaming [12].

Microservice in VANET is a promising software design and implementation approach that offers many benefits. Microservices facilitate dividing intricate applications into more manageable, autonomous, and independently implemented and expanded parts. It provides higher flexibility, modularity, and scalability in designing and implementing VANET applications [13]. Microservices can also improve fault tolerance, reliability, performance, and efficiency. Additionally, microservices enable greater flexibility and modularity in designing and implementing VANET applications, allowing for easier customization to meet the specific needs of different use cases and environments. As VANET systems become more complex and diverse, microservices will become increasingly important for successfully deploying and operating VANET applications. Microservices offer a promising approach to developing VANET systems that can improve reliability, efficiency, and flexibility [14].

The handover and migration processes are critical operations in VANET that enable seamless communication between network infrastructure. The handover process is challenging due to high vehicle mobility, network topology changes, and maintaining service continuity during handovers. Effective handover and migration mechanisms are essential for reducing service disruption and improving the reliability and performance of VANET applications [15]. The migration process is also important for integrating different network technologies and architectures, enabling efficient resource utilization, and reducing service disruption during handovers [16]. Microservices can simplify the handover and migration process by decomposing applications into smaller, autonomous, separately scalable components.

This work addresses the challenges posed by the ultimate requirements of 6G networks to enable vehicular communications with ultra-high vehicle mobility. The work presents an architecture for vehicular networks using a microservices-based approach. This design addresses latency and reliability concerns, ensuring seamless communication and effective data management. The key contributions of this work include the following:

1. Developing a reliable vehicular network architecture based on the distributed edge computing paradigm. The proposed structure deploys edge computing in two linked levels: MEC and fog;
2. Proposing a novel microservice approach for vehicular applications that considers task prioritizing. The proposed algorithm allows for greater flexibility, modularity, and scalability in designing and implementing VANET applications;
3. Implementing an efficient offloading approach for the proposed VANET based on the proposed microservice algorithm. It is difficult to control the energy usage and system delay during the vehicle's movement and quick change in the channels and bandwidth in wireless communication systems. With the assistance of RSU, the MEC server helps vehicles offload the computing tasks to MEC linked to the cellular base

- station. The offloading scheme effectively manages computational tasks between vehicles and edge servers, reducing latency and improving system energy efficiency;
4. Implementing a seamless handover approach to maintain seamless communication as vehicles move across different network coverage zones, ensuring uninterrupted service;
 5. Developing an efficient migration scheme for efficient data transfer between edge servers, ensuring data availability and consistency across the network. The proposed migration approach uses the proposed microservice algorithm to facilitate task migration between edge servers;
 6. Assessing the performance of the introduced microservice-based VANET regarding latency, availability, and reliability. Heterogeneous real-based simulation scenarios were considered for evaluating the proposed network and approaches.

2. Related Works

With the recent improvements in throughput and latency, cellular networks still face many failures and cannot provide the required reliable infrastructure for vehicular applications. This made the mobile network operators work hard to build a network that can be recovered automatically. Many proposed studies and solutions have been developed to enable VANET and its applications. This section summarizes the most related studies to our proposed VANET.

In [17], the authors provided a case study of independent vehicles that are foreseeable and trendy ways for vehicle expansion for the coming era. This type of vehicle requires uRLLC and the 5G cellular network among independent vehicles and its network infrastructure. The uRLLC supports the slicing solution for the network services and function for the vehicle network that is also offered by the 5G network for independent vehicles. This was all proven by a case study simulation, which proved that the slicing solution helps to develop the latency and reliability of the network system and ensures that uRLLC works well for vehicular network applications. Fakhar Abbas et al. [18] investigated the links among the VANETs affected by the high vehicle speed that causes quick breaks in these links. The authors proposed a new cluster-based routing scheme, i.e., an ad hoc multipath approach, that ensures the selection of maximum reliability links. Moreover, the authors optimized the proposed model using an ant colony optimizer (ACO). The ACO works to identify the best route path for the vehicles between the vehicle's links for VANETs represented in four quality of service (QoS) metrics, including reliability, end-to-end latency, throughput, and energy.

Ana Gómez-Andrades et al. [19] developed an automatic system diagnosis depending on self-organized maps (SOM) and Ward's hierarchical method. This helps in making the vehicular network more reliable and less costly. This system was tested using real LTE data to resolve its execution and match it with reference mechanisms. Andrei Vladyko et al. [20] presented a VANET system based on MEC that addresses the issues of high traffic density in automotive networks. The system provides an effective offloading method for vehicle traffic, considerably lowering the latency of data connection between vehicles and stationary RSUs. To assess the effectiveness of the developed VANET, the system was subjected to thorough simulation in a reliable setting. Furthermore, the system was experimentally tested, and the examinations verified that the system could effectively handle a large volume of traffic, minimize delays in communication, and offer a dependable solution for offloading in-vehicle networks.

The SDN-enabled vehicular network (SDN/VANET) provides full, real-time network management and monitoring. Yet, relying on a central control unit presents considerable obstacles, such as bottlenecks and controller failure. The distribution of the control scheme is a realistic option that poses important considerations regarding the deployment and the number of needed controllers. To address these problems, Azzedine Boukerche et al. [21] introduced an adaptive controller management technique for SDN/VANET. This technique used vehicle mobility density and delays to inform decision making. The authors used a split-and-merge clustering strategy that dynamically modifies the distribution of control

units based on real-time network conditions. The proposed clustering solution's performance was tested with genuine mobility traces. The results showed that the suggested approach greatly decreases energy consumption, lowers latency, and balances network strain. These findings demonstrated the effectiveness of the developed adaptive controller management technique for improving the overall efficiency and performance of next-generation vehicular networks.

Furthermore, Noura Aljeri et al. [22] suggested an effective proactive controller deployment and assignment technique for 5G SDN/VANET. This technique uses predicted vehicle traffic flow and latency data to dynamically optimize controller location and assignment. The proposed solution took a proactive approach, anticipating changes in traffic patterns and network circumstances and adjusting controller placement and allocation accordingly. This is critical for sustaining low latency and balanced load distribution in the fast-changing environment of vehicle networks. The authors ran a performance study to validate the technique across various mobility scenarios indicative of real-world vehicular networks. They evaluated the developed system's performance regarding delay and controller load. The study compared the proactive clustering strategy to conventional and contemporary controller placement strategies. The developed solution beat previous strategies by preemptively adjusting to traffic flow estimates, reducing end-user latency, and preventing controller overload. This proactive strategy successfully manages vehicular networks' dynamic and heterogeneous character, resulting in consistent and reliable network performance.

Penglin Dai et al. [23] investigated a scenario for compute offloading in a MEC-assisted architecture, concentrating on three critical aspects: job upload coordination across many vehicles, task migration, and servers' heterogeneous processing capabilities. The authors created a cooperative computation offloading (CCO) problem and solved it using the probabilistic computation offloading (PCO) approach. This approach lets MEC servers schedule live tasks independently. The PCO approach changes the objective function into an augmented Lagrangian form before iteratively achieving the optimal solution utilizing a convex optimization framework. Furthermore, the authors created an extensive simulation, and the results confirmed the PCO's superiority across various scenarios, emphasizing its effectiveness in minimizing job completion delays and optimizing resource utilization in MEC-assisted vehicle networks.

Haixia Peng et al. [24] investigated multidimensional resource management in VANETs powered by MEC and unmanned aerial vehicles (UAVs). The authors aimed to optimize the spectrum, computation, and cache resources available at MEC-mounted base stations and UAVs. To achieve this, they formulated a resource optimization problem to be solved by a central controller. Given the lengthy time necessary to solve the optimization problem and the tight delay constraints of automotive applications, the authors employed reinforcement learning techniques. They created a solution using the deep deterministic policy gradient (DDPG) algorithm. The simulation findings showed that the technique outperformed random resource management, resulting in higher QoS satisfaction.

Ammar Muthanna et al. [25] presented an integrated architecture that addresses the issues of traffic density and network coverage in VANET. This design combines MEC and SDN technologies to improve network reliability and scalability, particularly in high-traffic environments. To boost network coverage, the authors used a device-to-device (D2D) clustering mechanism that connects orphan nodes, i.e., vehicles disconnected from the network. This clustering strategy ensures that even the most difficult network nodes remain connected, increasing coverage and reliability. The suggested architecture enables ultra-low latency applications since it provides a stable and adaptable framework. The architecture's performance improved significantly when tested under realistic settings and network scenarios. Specifically, the MEC/SDN-enabled vehicular network architecture reduced task blockage rates by up to 74% compared to the baseline implementation. These findings demonstrate the efficiency of the suggested method in controlling high traffic densities while maintaining consistent network coverage.

The complexity of VANET's environment and vehicles' intrinsic mobility pose issues typical offloading solutions frequently fail to meet. Bingxue Qiao et al. [26] suggested the heuristic task migration computation offloading (TMCO) scheme to address these challenges. Unlike old approaches, TMCO dynamically identifies the best spots for unloading jobs from moving vehicles, ensuring that activities are performed within tight timeframes. This approach considered both vehicle mobility and strict delay requirements. In this technique, a hash table was used to track the number of tasks given to each server, while a random function mimics the likelihood of job offloading. According to experimental results, the TMCO scheme outperformed standard full-offloading systems in latency by 10% on average. The reliability, availability, and security of exchanged data are among the numerous challenges of vehicular networks. Andrei Vladyko et al. [27] resolved the scientific challenge of creating a vehicle network structure that guarantees secure and correct data transmission in the V2X. They aimed to improve the safety of people using the roads by utilizing blockchain technology and MEC. The authors implemented an offloading model to manage traffic dispatching to the MEC efficiently. The system and the implemented subsystems and algorithms were evaluated in a dependable simulation environment across several scenarios. The simulations' results proved the system's efficiency.

Nirmin Monir et al. [28] developed a seamless handover mechanism for MEC-based V2X systems. The suggested approach successfully handles the handover as vehicles travel between adjacent RSUs on multilane bidirectional roadways. The system was implemented with MECs linked to the RSUs, considerably reducing installation time and speeding up the handover procedure. The MEC platform uses a MEC controller, which runs under the control scheme of an SDN controller in charge of network management. The implementation of the SDN paradigm ensures a smooth handover process. The MEC controller speeds up the handover process, increasing overall efficiency. The evaluation results confirmed the effectiveness and efficiency of the suggested plan.

Noura Aljeri et al. [29] offered a new methodology for estimating connectivity time in autonomous vehicular networks. The authors investigated the most accurate way to estimate the communication lifetime between automobiles and infrastructure units by using a variety of machine learning algorithms. They based their assessments on realistic mobility traces from the Cologne city dataset to ensure practical relevance and applicability. This dataset contains detailed and authentic vehicle movement patterns required for creating and testing the predictive algorithm in real-world scenarios. The suggested model combined a variety of machine learning methodologies, and each technique was tested for its accuracy and dependability in forecasting connectivity duration. The model considered various parameters influencing connectivity length, including vehicle speed, density, and traffic circumstances. Simulations and experiments demonstrated the efficiency of the proposed paradigm. The approach improved communication reliability in autonomous vehicle networks by precisely forecasting connectivity length, allowing for smoother and more efficient interactions between vehicles and infrastructure. Table 1 summarizes the features of the previously introduced related studies and compares these works with the proposed work to provide the novelty of the work.

Overall, 5G high-availability testing is an important step in ensuring that next-generation wireless networks satisfy the demanding needs of modern applications. This evaluation is based on robust modeling stages that include advanced mathematical frameworks [30]. Table 2 presents some of the most common methodologies that are particularly useful in addressing performance and availability challenges in designing 5G infrastructures. SRNs and MUGFs have been widely employed in the literature to simulate and examine the robustness of telecommunication networks, specifically those related to 5G technology. These approaches allow researchers to accurately measure the likelihood of network failures and understand how they affect the overall availability of the system. For instance, SRNs can represent different states of network components and the changes between these states caused by failures and repairs. MUGFs, on the other hand, enable the combination of many

performance measures into a unified analytical framework, making it easier to conduct a thorough assessment of network reliability [31,32].

G/G/m queuing models are commonly employed in analyzing 5G architectures for performance evaluation, specifically in assessing latency. These models aid in comprehending the behavior of network traffic as it moves through different nodes, each with its unique service characteristics. The versatility of G/G/m models in supporting various arrival and service time distributions makes them well suited for representing the diverse and ever-changing nature of 5G traffic patterns. Research has demonstrated that the utilization of G/G/m models can offer valuable insights into identifying performance limitations and aid in devising strategies to reduce latency and enhance overall network performance [33].

Table 1. Main features of the existing related works.

Ref.	Key Technology					Handover	Migration	Evaluation	Metrics	Application
	Fog	MEC	SDN	AI/ML	Microservices					
[17]	x	✓	x	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Latency • Reliability • Accuracy 	uRLLC
[18]	x	x	x	x	x	x	x	Simulation	<ul style="list-style-type: none"> • End-to-end latency • Reliability • Throughput • Energy consumption 	General
[19]	✓	✓	x	✓	x	x	x	Simulation/ Experimental	<ul style="list-style-type: none"> • False positive rate • False negative rate • Diagnosis error rate 	Automatic diagnosis
[20]	✓	✓	x	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Latency • Reliability • Availability • Network Security • Energy consumption 	General
[34]	x	x	✓	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Delay • Reliability 	General
[35]	x	x	✓	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Availability • Security 	5G-enabled vehicular apps
[21]	x	x	✓	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Energy consumption • Latency • Load on network entities 	General
[22]	x	x	✓	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Latency • Load on network entities 	5G-enabled vehicular apps
[23]	x	✓	x	x	x	x	✓	Simulation	<ul style="list-style-type: none"> • Latency 	General
[24]	x	✓	x	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Latency • Resources efficiency 	UAV-assisted vehicular apps
[25]	x	✓	✓	x	x	x	x	-	-	General
[26]	x	✓	x	x	x	✓	✓	Simulation	<ul style="list-style-type: none"> • Latency 	General
[27]	x	✓	x	x	x	x	x	Simulation	<ul style="list-style-type: none"> • Latency • Communication overhead • Availability 	V2X applications
[28]	x	✓	✓	x	x	✓	x	Simulation	<ul style="list-style-type: none"> • Latency • Reliability • Availability 	General
[29]	x	x	✓	✓	x	x	x	Simulation	<ul style="list-style-type: none"> • Latency 	General
Proposed	✓	✓	✓	x	✓	✓	✓	Simulation	<ul style="list-style-type: none"> • Latency • Availability • Reliability 	General

Table 2. Most common methodologies that are particularly useful in addressing performance and availability challenges in designing 5G infrastructures [30–33].

Methodology	Description	Benefits	Applications	Limitations
Petri nets	Petri nets are graphical and mathematical modeling tools used to describe and analyze the flow of information in systems with concurrent processes.	It provides a visual and formal representation of system dynamics, making identifying and resolving bottlenecks easier.	In 5G networks, Petri nets can model and analyze the synchronization and resource sharing between different network components.	It can become complex and difficult to manage as the system size increases.
Fault tree analysis (FTA)	FTA is a top-down, deductive failure analysis that focuses on identifying the root causes of system failures.	Helps identify critical components whose failure would have significant impacts, thus guiding redundancy and maintenance planning.	FTA can be used to model the probability of different failure scenarios in 5G infrastructure, helping to design more resilient systems.	Requires detailed knowledge of the system architecture and failure modes.
Stochastic reward nets (SRNs)	SRNs are extensions of Petri nets (a place/transition net) that incorporate stochastic timing and reward structures to model and evaluate complex systems.	SRNs allow for a detailed and flexible representation of system behavior, capturing both the probabilistic nature of events and their rewards (or penalties).	In the context of 5G networks, SRNs can be used to assess system reliability and availability by modeling different states of network components and their failure/recovery processes.	The complexity of SRNs can lead to significant computational overhead, especially for large-scale 5G networks.
Monte Carlo simulation	Monte Carlo simulation uses random sampling and statistical modeling to estimate the behavior of a system.	Offers flexibility and can model complex systems that are difficult to analyze using traditional methods.	Monte Carlo simulations can evaluate the performance and reliability of 5G networks under various scenarios and uncertainties.	Requires significant computational resources and careful interpretation of results.
Multidimensional universal generating functions (MUGFs)	MUGFs provide a powerful framework for analyzing the performance and reliability of multi-state systems.	MUGFs offer a compact and efficient way to represent and compute the performance distribution of complex systems, making them suitable for high-dimensional problems.	MUGFs can be applied to model the availability of 5G infrastructure by evaluating the combined impact of multiple performance measures, such as bandwidth, latency, and reliability, in a unified manner.	Developing accurate MUGF models requires a deep understanding of the system's operational characteristics and dependencies.
G/G/m queuing models	G/G/m queuing models generalize traditional queuing theory by allowing for arbitrary arrival and service time distributions with multiple servers (m).	G/G/m models provide a flexible approach to analyzing the performance of network nodes under realistic traffic conditions, enabling the identification of bottlenecks and optimization opportunities.	These models are particularly useful for the characterization of latency in 5G networks, where traffic patterns can be highly variable and unpredictable.	Solving G/G/m models analytically can be complex, often necessitating numerical methods or simulations.

3. Proposed VANET

VANET supplies the process of interchanging data between vehicles. This takes place through several processes through communication links known as V2V communications or through vehicle-to-infrastructure (V2I), e.g., evolved node B (eNB) of long-term evolution (LTE). The V2X type also works on communication with the assistance of any components related to the IoT. Furthermore, the wireless heterogeneous networking that is built depends on all the 802.11p, Bluetooth, and various systems of the cellular network generations as

well as the routing protocol that helps to build an efficient VANET. This section introduces the developed VANET and provides all network features and components.

3.1. System Structure

Figure 1 presents the proposed VANET network that deploys multiple technologies and interfaces to support heterogeneous vehicular applications. Heterogeneous wireless technologies can be used to build communication links between vehicles and RSUs. This kind of communication link is known for wireless access in the vehicular environment (WAVE). Passenger safety is verified with the help of WAVE communication. WAVE communication works to bring up-to-date information on vehicles and traffic flow. This offers a huge improvement in the competence of the management traffic system. The proposed VANET contains various entities, including OBU, RSU, and trusted authority (TA). Many applications that RSU accommodates aim to link with various network systems. OBU is available in every vehicle to gather helpful data about each vehicle, such as fuel, speed, acceleration, and other data. All information is then sent to any nearby vehicles with the help of the wireless network. The wired network is used to link between TA and RSU, knowing that all RSU are connected. TA is one of the main components of VANET, as it helps keep the VANET system working. Figure 2 presents the main components of the lower layer of the considered VANET.

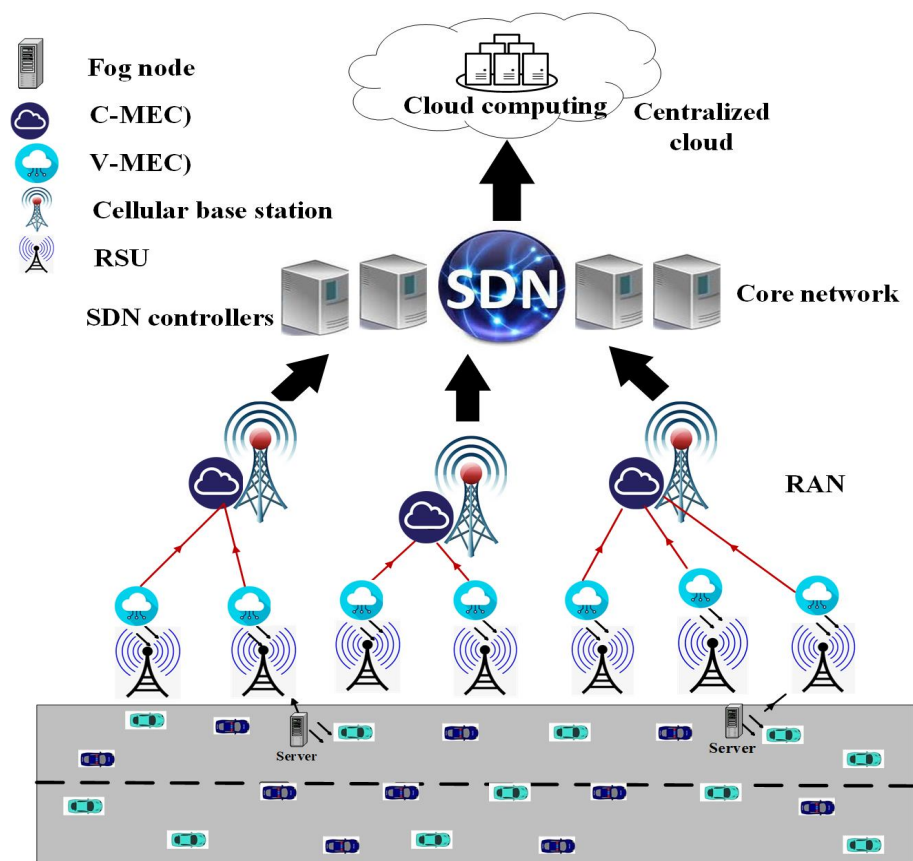


Figure 1. End-to-end structure of the proposed VANET.

The main features of these components are summarized as follows:

- RSU: RSU is situated in several locations along the roadsides. These locations include the parking regions or crossroads. This offers the vehicles domestic links for any crossing vehicles. The network devices inside the RSU depend on radio technology in IEEE 802.11P. This illustrates that these network devices are working to dedicate short-

range communication (DSRC). Besides this, RSU is useful to link different network systems with different network infrastructures;

- OBU: OBU is considered a global positioning system (GPS)-based tracking device available in each vehicle and offers vehicle data to transfer to RSU. Many electronic components are inside OBU, including IoT sensors, a resources command processor (RCP), an interface to users, and storage for read–write data to restore data. OBU’s fundamental task is linking OBU with RSU or other OBUs. The radio transceiver of the OBU uses the IEEE 802.11p for wireless connection [36]. This connection with OBUs and RSUs takes place in message form. Furthermore, OBU gets its power from the vehicle’s battery as well as the presence of several other components in each vehicle that work as an input for OBU, including the GPS, event data recorder (EDR), and forward and backward sensors;
- TA: TA’s main duty is to control the VANET system, including the vehicle users, RSU, and OBU. It is also responsible for guaranteeing the VANET security administration.

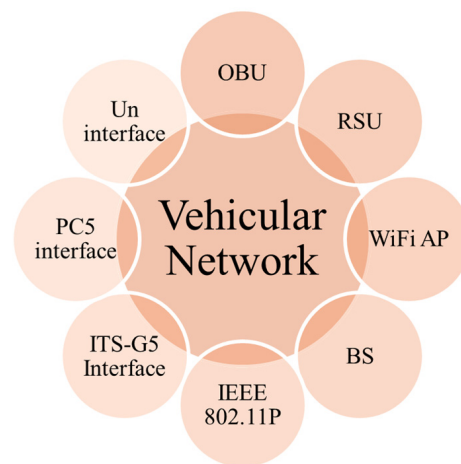


Figure 2. Main components of the lower level of the proposed VANET.

The proposed VANET deploys distributed edge computing at two tiers: fog computing (which represents the lower level) and MEC (which represents the higher level). Fog computing is considered an edge computing paradigm that helps the cloud computing network to enlarge its benefits and facilities among the network edges. It creates a link between the cloud and other smart devices that build all the links needed for the connection process. The difference between the cloud and the fog is that fog is assigned at the edges with less latency, while the cloud is more localized. Fog can send information, compute, save, stratify, and services for the end devices. VANET can meet its requirement by fog computing, for example, by decreasing the conjunction on the cloud and by fast reply to the important devices in addition to its capability to resolve the live data flow. The fog node can be represented in any device in the proposed VANET, including the infrastructure for roads and mobile vehicles. Each fog unit has all the facilities of network connections, stock, and computing units.

The proposed VANET also deploys MEC nodes into two levels. The first level represents the vehicular MECs (V-MECs), which are MEC servers connected to RSUs. The second level is made up of cellular MECs, which are MEC servers that are linked to eNBs (M-MECs). Using two MEC levels makes it easier to implement the proposed networking algorithms. The V-MEC uses our proposed handover and migration methods, which were introduced later, resulting in higher execution efficiency than standard systems. The introduction of distributed edge units makes it easier for V2X applications to meet the requisite QoS.

SDN is considered one of the gathering techniques that stands out among programmable networking technologies. SDN is a main part of the proposed VANET that

can adapt to any alteration in the network topology. Besides its ability to realize the programmability portions related to the VANET control plane, it plays an important role in delivering novel services. SDN plays an important role in configuring network simplicity, computing, and new methods in network management.

Traditional VANET networks experience weak flexibility; however, deploying SDN for such networks achieves high flexibility. SDN can mirror between the data and the control plane and send the logical control starting from the node and ending to the central controller. Joining fog computing, MEC, and SDN helps cope with the challenges of the VANET system, including nonintelligence, inflexibility, and short connectivity. The SDN offers network behavior control while the fog computing transfers the position and service slots. The mirroring between the SDN and fog helps lower delays and improves network resources. The integration of SDN guarantees a seamless process for the established handover method by maintaining a persistent TCP connection.

3.2. VANET Features and Interfaces

The VANET wireless network contains nodes distributed along the RSU or vehicles with high mobility. The network's infrastructure restricts access to it; however, it is highly dynamic and dependable and provides many services. The proposed VANET has the following features:

- Scalability and flexibility: Edge computing enables dynamic resource allocation based on current demand, maintaining optimal performance. SDN offers a flexible networking framework that can readily scale to support increasingly linked vehicles and devices;
- Low latency and real-time processing: Distributed edge computing brings computer resources closer to vehicles, dramatically lowering latency by processing data locally rather than depending on remote cloud servers. Edge nodes offer real-time analysis and speedy decision making, which benefits critical applications such as collision avoidance, traffic management, and autonomous driving;
- Enhanced network management and control: SDN provides centralized network control via a controller that governs data flows across the network, simplifying network management and increasing overall efficiency. SDN's programmability enables the customization and optimization of network policies and behaviors, which adapt to changing traffic conditions and application requirements;
- Improved security and privacy: SDN increases security by isolating essential network control functions from possible attackers. Edge computing enables local processing of sensitive data, reducing the need to send it across the network and improving privacy;
- Load balancing: SDN allows for sophisticated load balancing across the network, reducing congestion and assuring effective use of network resources;
- Resource optimization: Edge nodes can be dynamically deployed and scaled to meet real-time demand, ensuring that computational resources are used efficiently;
- Enhanced QoS: SDN prioritizes vital vehicular data, ensuring that high-priority communications, including emergency alarms, are delivered quickly. SDN facilitates smooth control over network traffic and enables consistent and reliable performance for various applications, from entertainment to safety-critical systems;
- Reduced backhaul traffic: Processing data at the edge reduces the amount of data that needs to be transported back to central servers, resulting in energy savings and lower backhaul traffic;
- Adaptive network topology: With SDN's centralized control, the network topology may be dynamically adjusted to accommodate vehicle mobility while maintaining robust communication links;
- Efficient resources management: SDN's ability to dynamically assign network resources based on demand contributes to energy conservation by preventing resource overprovisioning;

- Efficient handover implementation: The system can effectively manage handovers between RSUs by combining edge computing and SDN, assuring continuous connectivity and service for moving vehicles;
- Advanced data analytics and AI integration: Distributed edge nodes may execute complex data analytics close to the data source, delivering actionable insights in real time for applications such as traffic prediction and route optimization. Integrating AI and machine learning algorithms at the edge might help improve decision-making processes, including predictive maintenance and intelligent traffic management;
- Interoperability: SDN enables interoperability across various network protocols and standards, resulting in seamless communication across various automotive systems.

The proposed VANET supports many forms of communications, including V2V, V2I, vehicle-to-pedestrians (V2P), V2X, and cellular vehicle-to-everything (C-V2X) communications. Enabling such forms of communication provides efficient communication over the network, provides all kinds of vehicular applications, helps in traffic safety, and offers very well-authorized data in real time. In V2V communications, the high transmission rate and the low latency are among the main advantages. Vehicles can propagate beneficial data like braking for emergencies, detecting accidents, or traffic situations between vehicles. V2I works mainly to send data among vehicles and the network's infrastructure. Then, the vehicles improve the link with RSUs to interchange data with different networks, such as the Internet. Moreover, the V2I demands a wider bandwidth than V2V, making V2I more protected against any risks or attacks.

C-V2X significantly improves vehicular communications, providing reliable, robust, and low-latency communication capabilities. C-V2X increases road safety, traffic efficiency, and a variety of ITS applications by utilizing both direct and network-based communication. C-V2X is a communication technology specifically developed to offer connection for ITS. C-V2X, a 3GPP standard, enables direct and network-based communication across cellular networks. Using C-V2X, vehicles can communicate directly with each other, infrastructure, and pedestrians through the PC5 interface. This direct communication interface supports the frequency band of 5.9 GHz ITS allocated mainly for vehicular communications. C-V2X enables vehicles for cellular communication using the Uu interface, which was developed for next-generation NodeB (gNB) [37]. This interface provides access to cloud services, traffic information, and broader network resources.

Integrating fog and MEC computing with SDN brings control facilities to the network's edge. Instead of delivering data to a central controller, fog computing processes services and data at the network's edge. Low latency, location awareness, and enhanced QoS are just some advantages. We considered the MEC platform introduced in [28] for the proposed VANET. We considered a distributed control scheme, in addition to the centralized controller, to provide global intelligence. Each RSU deploys an SDN RSU controller (RSUC), which has a direct interface to the centralized scheme. This distributed scheme supports distributed edge computing and assists the MEC controller deployed in the considered MEC platform. The SDN controller controls the RSUC, which is also an SDN-enabled device. On the other hand, the SDN controller does not take full control of the network; instead, it gives policy information to the RSUC.

The RSUC decides on the exact conduct based on the policy information received and their local understanding. A fog orchestration layer is placed at the SDN architecture, and the operation is located at the SDN controller coordinates between RSUCs. The RSUC sends data to the SDN controller while also maintaining local storage. In the event of an emergency, RSUC serves the requests rather than sending them to the controller, resulting in a significant reduction in latency. This has raised the same worry about latency control, emphasizing the significance of an edge-up design with hierarchically distributed control and an edge layer. This is rather than a cloud-down approach (Internet-based SDVN) in terms of the use of radio resources.

4. Proposed Microservice-Based Scheme

Microservices are becoming increasingly popular in many industries, including the automotive sector, due to their ability to improve software systems' flexibility, scalability, and maintainability. Microservices can potentially revolutionize the design and operation of VANETs in the designed system. VANETs are networks that enable communication between vehicles, road infrastructure, and other devices in the transportation ecosystem. These networks are characterized by their dynamic and rapidly changing nature, which makes them challenging to design and operate. Microservices are tiny, self-contained, and loosely interconnected software elements that can be deployed and scaled autonomously, making them an important part of any designed network system. We considered microservices for the proposed VANET since they offer several benefits, including fault isolation, ease of maintenance, and the ability to handle large amounts of data. By adopting microservices in the designed VANET, it is possible to create modular and scalable systems that can handle the challenges of vehicular networks. Microservices can implement various services in the proposed VANET, including traffic monitoring, collision warning systems, and remote vehicle diagnostics.

Another important benefit of implementing microservices in the designed VANET is their ability to improve the QoS, which is a critical aspect of VANETs, as it affects the performance of safety-critical applications such as collision warning systems. Microservices can help to improve QoS by enabling dynamic scaling, fault tolerance, and load balancing. For example, microservices can be used to automatically scale up or down the processing power of a given service based on the current demand. This approach ensures the service can handle the current load while minimizing resource usage. The use of microservices in the proposed VANETs can enable new and innovative services that were previously impossible. For example, microservices can be used to implement new applications that rely on real-time data, such as traffic congestion monitoring or parking space detection. These services can be deployed as independent microservices that can be easily integrated into existing systems, reducing the time and cost of development.

Each microservice can be deployed and scaled independently, allowing for greater resource utilization and improved performance. This is particularly important in the proposed VANET, where resources are limited, and efficient use of these resources is critical for the success of VANET applications. By adopting microservices, it is possible to create modular and scalable systems that can handle the dynamic and rapidly changing nature of any VANETs. This approach allows for greater flexibility, modularity, and scalability in designing and implementing VANET applications. Algorithm 1 provides the proposed microservice scheme for the considered VANET.

Algorithm 1: Microservices model for vehicular applications

```

1: Initialize service type (T), service priority, microservices (M), and categories of microservice
2:   For (i = 1: T)
3:     Output: Define the service type
4:     Output: Define the service priority
5:   End for
6:   For (x = 1: M)
7:     Output: Define the microservice type
8:     Output: Define the microservice priority
9:   End for
10: End

```

The proposed microservices algorithm depends mainly on the task priority. The algorithm starts by prioritizing the received tasks according to their main applications. It works by first indicating the data priority, defining the application, and assigning binary values to each task. If it is a high priority, it takes the data and divides the data service into microservices. After this, the microservices are also numbered based on their priority, with the decision of which data or service must be sent first based on their priority number. This

reduces the task load on the vehicle service. Based on task priority, it is now well known which task needs to be handled and which will be handled by the MEC service. The main steps of the proposed algorithm are introduced as follows:

- *Step 1:* The algorithm starts by initializing all the necessary parameters, including the service type, counter for the service priority (T), service priority, microservices, counter for the microservices priority (M), and categories of microservices;
- *Step 2:* We start a for-loop that starts from 1, which indicates a task with high priority, until T, which represents the counter for the service priority with less priority. Each step-in loop indicates the task “service type” and “service priority.” When the loop is done, we move to step 3;
- *Step 3:* After defining the task priority and type, we divide each task into microservices and categorize each based on microservice and microservice priority. In this case, the for loop starts from 1, which is the high priority, to M, which represents the counter for the microservices priority. Each step-in loop indicates the task “microservice type” or “microservice priority”.

5. Handover Scheme

VANET consists of vehicles traveling along roads or streets. Its main purpose is to improve vehicles' safety, comfort, reliability, and security on the road. However, one of the most significant challenges in designing VANET is the issue of handover (HO), which arises when a vehicle goes from one network region featured by an RSU to another region encompassed by a different RSU. Achieving seamless HO in VANET is complex because of the high vehicle mobility and the limited coverage of RSUs, which do not have overlapping signals. 5G technology can help provide enhanced characteristics that are well-suited for real-time video streaming and can be utilized in designing effective solutions for the HO problem. Any system must include a handover method to facilitate the transfer of active communication management between RSUs. For example, let us assume vehicle A is traveling from the coverage region of RSU₂ to RSU₁, whereas vehicle B is moving between the coverage of RSU₁ to RSU₂. After the handover procedure is finished, RSU₁ takes charge of communication management for vehicle A, while RSU₂ acquires responsibility for vehicle B. Two main categories of handover methods exist: inter-RSU handover and intra-RSU transfer [38].

Inter-RSU handover refers to the process in which a vehicle moves from one RSU to another, requiring a smooth transfer of communication duties to ensure uninterrupted connectivity. In contrast, intra-RSU handover refers to the process of a vehicle transitioning between various types of connections inside the same coverage area of an RSU, such as switching from V2V communication to vehicle-to-RSU (V2R) communication. Both types of handover approaches are crucial for guaranteeing efficient and dependable communication management in dynamic vehicle networks. Inter-RSU handover in VANET can be classified based on several criteria, one of which is the classification of handover into soft or hard HO. In hard handover, the link to the previous RSU is disconnected before connecting to the incoming RSU, while in soft handover, the link to the incoming RSU is established before the disconnection of the previous RSU [39].

The handover process in an SDN vehicular network is more intricate than in conventional cellular networks. Initially, the radio resources may be renegotiated with a new SDN controller. Secondly, flow tables must be updated to reflect the topology. Third, live migration and service redirection are required if a MEC is present, which adds even more complexity to the handover process. Lastly, the generation of the handover for multi-hop links and MEC services will occur concurrently. Each of these elements increases the difficulty of handover.

Trajectory prediction can be used to improve the handover process. This can assist in finishing flow table entry updates and service migration ahead of schedule. Learning-based methods are one way to tackle the problems of high mobility and handover. These techniques can be used to find possible trends that could lead to better load balancing and

handover. For instance, reinforcement learning has been applied using base station context data such as vehicle speed, user count, and handover history.

Similarly, online probabilistic neural networks have been proposed to use the mobility information of the vehicles to predict the next serving access point. However, the complexity of creating an appropriate objective that simultaneously maximizes the performance makes these machine-learning (ML) algorithms difficult to use, particularly when energy consumption needs to be considered. SDN prioritizes security and privacy because of the possibility of unauthorized access or false information causing catastrophic mishaps. Security mechanisms must perform real-time authentication to avoid traffic congestion that could hinder SDN operation [40].

Smooth vehicle HO is essential in a system designed with 5G-VANET to minimize user interruptions. Nevertheless, frequent handovers can result from vehicles' high mobility, which presents a major obstacle in VANET. These difficulties show the importance of finding practical ways to deal with the HO issue in systems built for 5G-VANETs. Seamless communication between vehicles must be maintained without interruption when they move from one RSU to another, which is challenging due to the latency caused by handover.

5G-enabled vehicular networks offer a promising solution to enable real-time services in high-density urban areas. Existing HO solutions do not effectively address the most significant problems in VANET, such as reduced quality of connection and QoS performance degradation. Frequent HOs can also lead to unnecessary and excessive signaling, resulting in a signaling storm that consumes VANET resources and energy, ultimately leading to HO failure. Several critical factors contribute to HO failure in VANET, including mobility, communication interference, coverage regions, and traffic congestion. Addressing these problems is essential for designing effective HO solutions to improve the VANET's QoS [28].

The high-speed movement of vehicles on highways poses significant mobility challenges in the proposed layout of 5G-VANET networks. Addressing these mobility problems is critical for ensuring seamless vehicle communication and reducing HO failures in the proposed design 5G-VANET.

Many of the current HO solutions address the ideal situation of vehicles traveling on roads and being HO from one RSU to another. However, signal strength information has been the primary focus of ping-pong solutions in the literature, which may be inaccurate in environments with fluctuating signal strength. Increasing the HO margin, HO performance indicator, and time-to-trigger are examples of HO parameters that can be improved to reduce pointless HOs in 5G-VANET networks. Addressing these HO problems is crucial for ensuring seamless communication and reducing the frequency of HO failures in high-speed highway environments. By developing effective solutions to these HO problems, 5G-VANET networks can improve the quality and reliability of communication between vehicles and RSUs, ultimately enhancing the safety and efficiency of highway transportation.

One approach is to use advanced encryption techniques to secure communication between vehicles and RSUs. Another approach is to use frequency-hopping techniques to avoid jamming attacks by changing the transmitted signal's frequency in a pseudo-random pattern. Additionally, directional antennas can be used to minimize interference and improve signal quality in areas with high signal interference. ML algorithms can also be deployed to detect and prevent jamming attacks by analyzing the patterns of jamming signals. In summary, jamming and interference signals pose significant challenges in designed 5G-VANET networks and can negatively impact network performance and driver safety. Addressing these challenges requires the deployment of effective countermeasures such as advanced encryption techniques, frequency-hopping techniques, directional antennas, and ML algorithms. By mitigating the impact of jamming and interference signals, 5G-VANET networks can ensure the safety and reliability of vehicle communication, ultimately enhancing the efficiency and effectiveness of transportation on highways. For the proposed microservice-based VANET, we considered the HO scheme introduced in [28].

6. Proposed Migration Scheme

Migration is a critical aspect of any designed VANET, which refers to the process of moving from one network to another. VANETs enable communication between vehicles, road infrastructure, and other devices in the transportation ecosystem. Due to the dynamic and rapidly changing nature of VANETs, migration is essential to network design and operation. By enabling vehicles to maintain connectivity and access services as they move between different network environments, migration plays a vital role in ensuring the reliable and efficient operation of VANETs. One of the primary challenges of migration in designed VANETs is ensuring seamless and uninterrupted connectivity during the transition from one network to another. This is particularly important for safety-critical applications such as collision warning systems, which rely on continuous connectivity to operate effectively. Therefore, designing and implementing migration mechanisms in the designed system to ensure seamless connectivity is crucial for the safety and reliability of vehicular networks.

Another challenge of migration in VANETs is the potential loss of data during the transition from one network to another. This can occur due to network latency, packet loss, or other factors that may cause data loss during migration. Researchers have proposed several approaches to address this challenge to ensure data continuity during migration, such as caching, replication, and proactive handoff mechanisms. These approaches aim to reduce data loss and ensure critical data are available to the vehicle during migration. Moreover, migration in the proposed designed VANETs presents opportunities for innovation and new services. For example, the ability to migrate between different types of networks, such as cellular networks and dedicated short-range communication (DSRC) networks, enables the development of new services that rely on the strengths of each network type. This approach can improve the reliability, scalability, and efficiency of VANETs by leveraging the strengths of different network technologies.

In the designed VANET, the migration process is challenging because of the high vehicle mobility and the need to maintain service continuity during handovers. The migration process must consider several factors, including the vehicle's location, speed, direction, and communication requirements. Additionally, the migration process must be fast enough to ensure that the handover occurs without interruption in service. Furthermore, the migration process must be able to handle different types of services, such as infotainment, safety, and traffic management, which have varying requirements in terms of latency, bandwidth, and reliability. The migration process also critically integrates different network technologies and architectures, such as cellular networks, Wi-Fi, and ad hoc networks. This integration allows for the efficient utilization of network resources and the provision of high-quality services to vehicular users. The migration process in the proposed VANET must be designed to handle the dynamic nature of network environments, including changes in network topology, traffic load, and available bandwidth.

One of the most important benefits of an efficient migration process in a designed VANET is the reduction of service disruption during handovers. This is particularly important for safety-critical applications that require real-time communication such as collision avoidance and emergency response systems. The migration process must ensure that these applications continue to function seamlessly during handovers without any delay or loss of data. The migration process is a critical component in the designed VANET, enabling seamless transfer of services and data between different network infrastructures. The migration process must be fast, reliable, and efficient to ensure uninterrupted communication for vehicular users. It also plays a critical role in integrating different network technologies and architectures, reducing service disruption during handovers, and enabling safety-critical applications. Therefore, developing effective and efficient migration mechanisms is essential for the success of VANET and realizing its full potential for improving road safety, traffic efficiency, and passenger comfort. Algorithm 2 introduces the proposed migration scheme for the developed VANET. The algorithm is built based on our previously developed handover scheme proposed in [28].

Algorithm 2: Handover and migration algorithm

```

1:   Initialize SNR1, SNR2, SNR3, Signal power of RSU
2:   Initialize positions xi (i = 1, 2, 3, . . . , n)
3:   If (P > SNR1)
4:       Output: Dedicate parameter
5:       Output: Dedicate P RSU
6:   Else
7:       If (P < SNR1) || (P > SNR2)
8:           Output: Start inspection state phase
9:       Else
10:          If (P < SNR2) || (P > SNR3)
11:              Send a migration request to target MEC
12:              Initialize migration process
13:              Send task specification target MEC
14:              Receive migration response and time stamps
15:          Else
16:              Start the execution stage
17:              Migrate computing tasks
18:              Load to target MEC
19:              Move serving vehicle parameters to RSU
20:          End If
21:      End If
22:  End If
23:  End migration

```

The power indicates the vehicle's position with the RSU. The signal power for the RSU is denoted as P , and the signal-to-noise ratio (SNR) is calculated by dividing the signal RSU power value (P) by the noise power of the channel. The main steps of the proposed migration algorithm are summarized below:

- *Step 1:* First, the algorithm initializes all the parameters needed, including the SNR and P ;
- *Step 2:* Then, it compares the P with the SNR₁ (i.e., the first level of thresholds). If the $P > \text{SNR}_1$, two actions will occur: The first is "dedicate the parameter", and the second is "dedicate P RSU". If not, it goes to step 3;
- *Step 3:* We have two possibilities if P is not greater than the SNR₁. These possibilities are ($P < \text{SNR}_1$ or $P > \text{SNR}_2$). In this case, if either of the possibilities is true, the action is to "Start the inspection state phase". If not, then it goes to step 4;
- *Step 4:* If P is not smaller than SNR₁ or greater than SNR₂, then we have another two possibilities. These two possibilities are ($P < \text{SNR}_2$ or $P > \text{SNR}_3$). In this case, if either of the possibilities is true, the action is 4 steps. The first is "send migration request to target MEC", and second is "Initialize migration process". The third is "Send task specification to the target MEC", and fourth is "Receive migration response and time stamps". If not, it goes to step 5;
- *Step 5:* If P is not smaller than SNR₂ or greater than SNR₃, we have any other special condition. In this case, the action will be four steps. The first is "Start execution stage", the second is "Migrate computing task", the third is "Load target MEC", and the fourth is "Move serving vehicle parameters to RSU."

The proposed algorithm covers all possibilities during the handover and migration processes. The migration process takes place at the start of the handover algorithm.

7. Offloading Scheme

For the proposed VANET, we modified our proposed offloading scheme in [41]. The considered offloading model deploys three main offloading levels in addition to the local execution, and Figure 3 presents the main offloading levels in the proposed VANET. The local execution is represented as zero-level offloading, which illustrates an important

mission of the end device’s ability to process these tasks without the offloading procedure. This can occur based on obtainable resources offered by the end devices and QoS assigned for each task. The first level of offloading is exemplified in the fog computing server. The mission that cannot be carried out with the help of end devices is offloaded with the help of a suitable communication connection to the RSU’s MEC. Depending on the recent obtainable resources at the fog server, the fog server admits or declines the offloading. If the recently obtainable resources of the fog server are not sufficient at that time to manage the offloading workloads, the workload is moved to the second level. The fog server offloads all workloads that cannot be managed to the dedicated RSU’s MEC. The MEC unit can admit or decline the offloaded tasks depending on recent obtainable resources besides the QoS needed for these tasks.

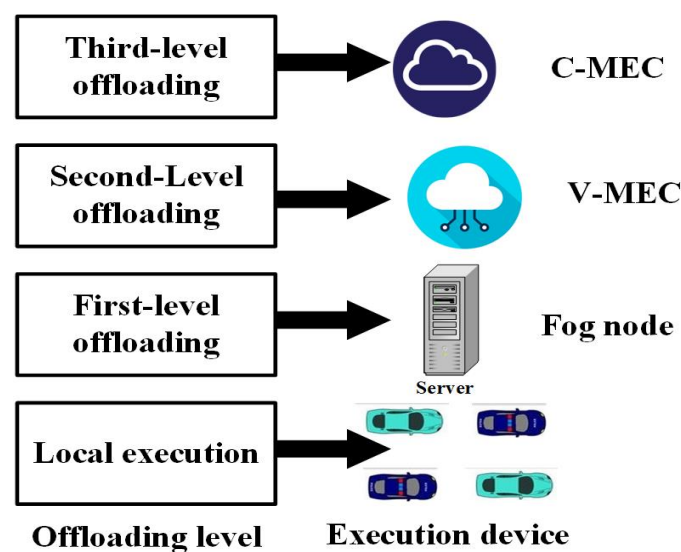


Figure 3. Considered offloading levels of the proposed microservice-based VANET.

The major offloading decisions are determined through the end devices based on their recent obtainable facilities, e.g., energy, storage, and processing, and the QoS time. Each device’s programmed profile determines the size of the workload and the necessary number of processor cycles. The end device’s decision engine implements hardware for the proposed offloading mechanism. It collects data from the program profile and data on the resources currently available from the resource’s schedule. Additionally, the decision engine gets the maximum permitted time to process the computing work, which indicates the needed QoS time for the recent workload. The decision engine uses these variables to select the offloading. The decision engine initially determines how long it will take to complete the job locally at the vehicle and then decides how to offload in terms of energy and time.

The energy choice is made by comparing the vehicle’s remaining energy after task execution with its threshold energy. The job is passed straight to the fog server if, following local execution, the vehicle’s remaining energy is greater than that, provided that offloading is permitted. The execution time and QoS latency are compared to determine the time decision of offloading. The job is offloaded to the fog unit if the decision engine chooses to offload. When the fog server’s decision engine receives an offloading request from a vehicle, it determines whether to execute the job internally or to move it to the next level. The decision engine of the fog server determines the execution time needed to complete the requested job before calculating the total time needed to complete it. The decision engine of the fog then determines the binary offloading energy and time decision values. After completing the desired job, leftover energy is compared to the energy threshold to determine the energy decision. The necessary QoS latency and the overall time needed to complete the requested job are compared to determine the time choice.

8. Simulation and Results

8.1. Simulation Setup

Considering various vehicular applications and scenarios, the proposed microservice-based framework was evaluated over NS-3 environments. The mobility of considered vehicles was achieved using the simulation of urban mobility (SUMO) platform. We used the ms-van3t platform available at [42]. We modified our previously proposed system model introduced in [43] and considered it for the proposed VANET. The vehicles' mobility was simulated to match highway conditions. Utilizing OpenFlow version 1.3, all essential SDN modules for the SDN controller and OpenFlow switches were integrated. The simulation topology was developed in accordance with the specifications outlined in Table 3. The simulation consisted of a five-kilometer-long, bidirectional, two-lane road with an RN87-like curvature. The traffic density (TD) parameters summarized in Table 3 were used to distribute vehicles along this road. Table 3 identifies the specifications of the used MEC servers and RSUs. It also displays the attributes of the fog nodes and the SDN network employed in the simulation. The road considered was covered by five RSUs, with a circular coverage of 0.5 km radius for each RSU. The system deployed five MEC units connected to RSUs, with the specifications introduced in Table 3. Each RUS coverage was served by five fog nodes distributed over the coverage area. Communication was carried out over the NR C-V2X using the platform introduced in [42].

Table 3. Parameters used during simulation.

Parameter	Value
Road	Length = 5 Km Direction: Bi-directional Lane: Two Curvature: RN87-like
RSU	Number = 5 Interface = NR C-V2X Radio transmission range = 500 m Max. transmission level = 20 dBm Packet transmission frequency = 10 Hz
Vehicles	Number (N) = $N \in \{200, 400, 600, 800\}$ TD = $TD_{vi} \in \{0.1, 0.2, 0.3\}$ veh/m Mobility (v) = $v_i \in \{30, 40, 50, 60, 70\}$ Km/h Interface = NR C-V2X
SDN	Service rate _{Controller} = 30,000 req/sec OpenFlow version = 1.3 Average request rate _{Switch} = $\epsilon [1500, 3000]$ Processing delay _{Controller} = 0.5 μ s Processing delay _{Switch} = 5 μ s
MEC	Number = 5 Placement = equidistant Processing $\epsilon [0.7, 2.5]$ GHz Max. workload = 10 Mbps Storage (RAM, HDD) = (4 \times 2048 Mb, 256 Gb)
Fog	Number = 5 per RSU Processing $\epsilon [0.7, 1.4]$ GHz Storage (RAM, HDD) = (2048 Mb, 10 Gb)
Tasks	Streaming bandwidth $\epsilon [10, 2048]$ (Kb/S) App Category: I, II, III, IV

Heterogeneous computing tasks of different applications were considered during the simulation. These tasks were of different applications, including sensor data, image processing, V2V communications, video processing, and traffic management. The data were extracted from the datasets introduced in [44–48]. Various datasets were considered to cover a variety of computing tasks ranging from simple to complex tasks. The tasks were categorized into four main categories in terms of computing complexity as follows:

- Category (I): Very simple tasks, including sensor data;
- Category (II): Simple tasks, including workloads equivalent to processing simple images and websites;
- Category (III): Medium-complexity tasks, including 3D images and simple videos;
- Category (IV): High-complex tasks, including high-quality and 3D videos.

Tasks were randomly allocated to the considered vehicles, and the next task was assigned once the task was completed.

To investigate the influence of the growing traffic on the efficiency of the developed microservice-based VANET, we examined four different values for the total number of deployed vehicles. In addition, we analyzed the impact of varying levels of TD by assessing three distinct traffic density scenarios. The three TD scenarios provide additional insights into the impact of alterations in vehicle distribution on the VANET. These scenarios involved varying levels of TD, ranging from low to high, with each level correlating to different distances between vehicles and frequencies of interaction. The vehicles were placed along the route based on the traffic densities stated in Table 3.

Three main performance metrics were considered for evaluating the proposed microservice-based VANET: reliability, availability, and latency. The reliability of the proposed network was assessed by measuring the successfully received packets. The packet delivery ratio (PDR) was calculated for the proposed system and compared with other existing solutions. The system availability was assessed by measuring the number of blocked tasks (BT) among the assigned tasks to vehicles. The results were compared with other existing systems to indicate the performance improvement of the developed microservice-based VANET. The latency performance of the proposed network was assessed by calculating the performance improvement in latency of handling assigned computing tasks. The following equation was used to calculate the percentage improvement of latency compared with the traditional MEC VANETs.

$$Latency\ improvement\ (\%) = \frac{Average\ latency_{Traditional} - Average\ latency_{System(i)}}{Average\ latency_{Traditional}} \quad \forall i \in \mathbb{R} \tag{1}$$

8.2. Results

Figure 4 presents the average value of the obtained PDR for the three systems with the parameters provided in Table 4. The PDR was measured at different traffic densities and distances from the RSUs. The proposed microservice-based VANET (system 3) achieved higher PDR at the three considered traffic densities. With the increase in the TD, the PDR of the existing systems (system (1) and system (2)) was reduced compared with the proposed microservice model. This is mainly due to the deployment of microservices and the proposed migration scheme. Moreover, the PDR of the three systems was measured at different vehicle mobilities. Figure 5 presents the average PDR of the three systems at different vehicle velocities with the specifications presented in Table 5.

Table 4. Parameters for PDR measurements (Figure 4).

Parameter	Vehicle Velocity (v)	Number of Vehicles (N)	System (1)	System (2)	System (3)
Value	50 Km/h	400	Traditional VANET	System developed in [28]	Proposed microservice-based VANET

Table 5. Parameters for PDR measurements (Figure 5).

Parameter	Traffic Density (TD)	Number of Vehicles (N)	System (1)	System (2)	System (3)
Value	0.1 veh/m	400	Traditional VANET	System developed in [28]	Proposed microservice-based VANET

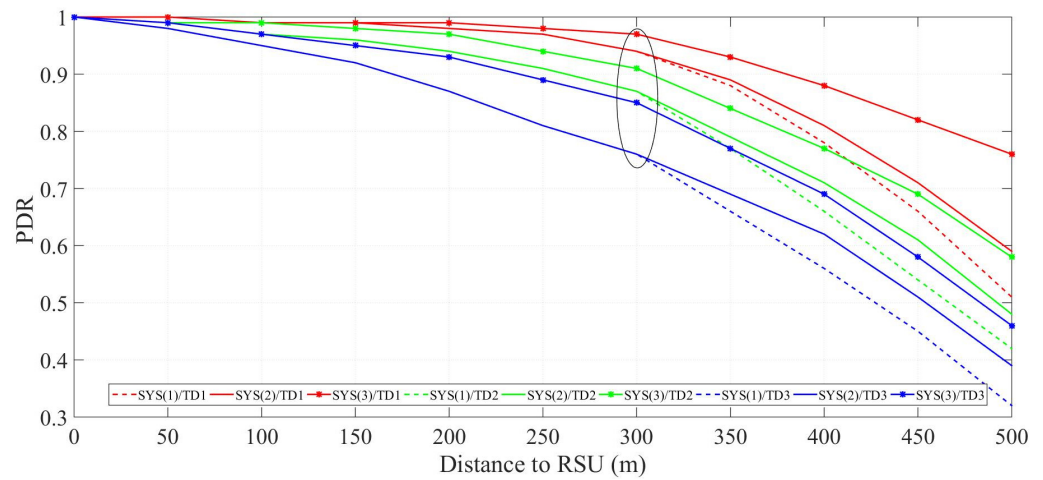


Figure 4. PDR at different TDs.

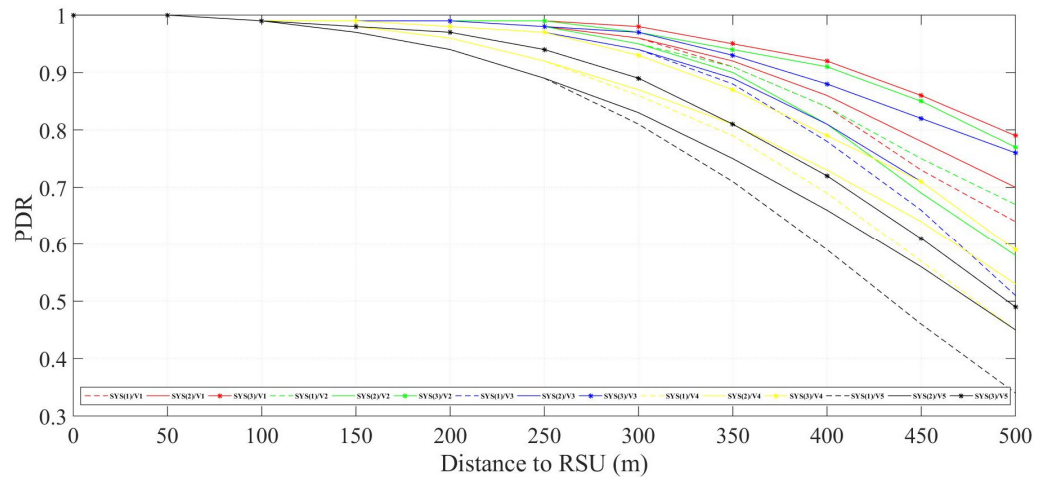


Figure 5. PDR at different vehicle mobility.

Figures 6–9 present the average performance of utilizing computing resources of vehicles, fog, and MEC nodes. This measure was considered to evaluate the performance of the proposed microservice and migration schemes. Also, this investigates the effectiveness of introducing the fog layer to VANETs. These figures measured the average resource utilization at different vehicle mobilities with the specifications introduced in Table 6. Each figure represents the results while simulating the systems with tasks from certain application categories. Figure 6 presents the average resource utilization of the four systems simulated using Category (I) tasks. The proposed model achieved lower utilization performance for this category of applications compared to other simulated systems. This is due to the plurality of available resources, e.g., fog and MEC, compared to the workloads required for processing. Since the assigned tasks are all sensor data that require low computing resources, this makes the resources of end devices, i.e., vehicles, enough to handle most of such tasks. However, with the increase in vehicle velocity, the performance of traditional systems degraded, and the performance of the proposed system increased.

Table 6. Simulation parameters for utilization efficiency measurements (Figures 6–9).

Parameter	Traffic Density (TD)	Number of Vehicles (N)	System (1)	System (2)	System (3)	System (4)
Value	0.1 veh/m	400	Traditional VANET	Traditional MEC-VANET	System developed in [28]	Proposed microservice-based VANET

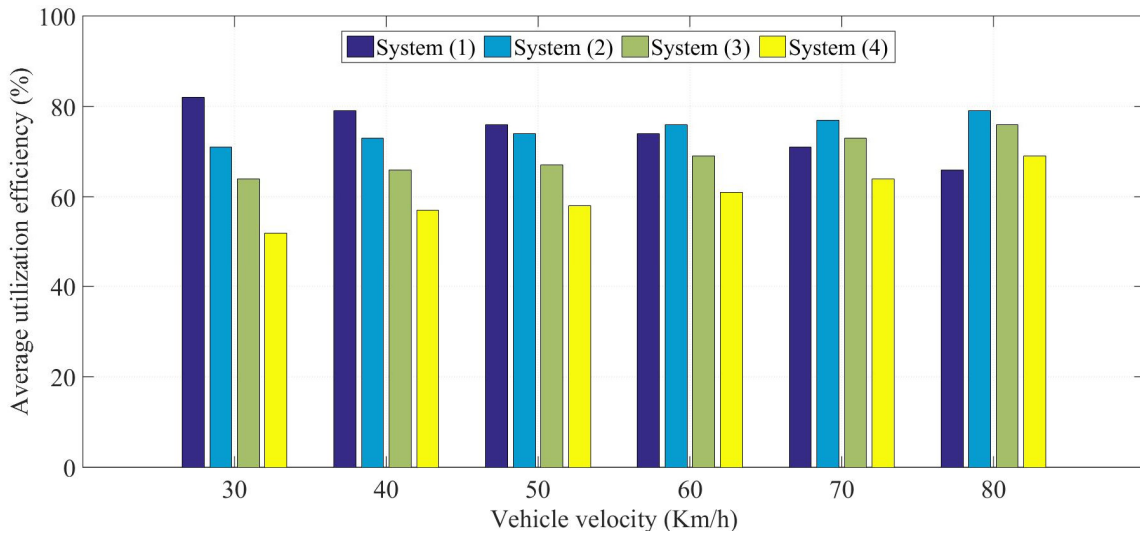


Figure 6. Average resource utilization efficiency at different vehicle mobility for Category (I) tasks.

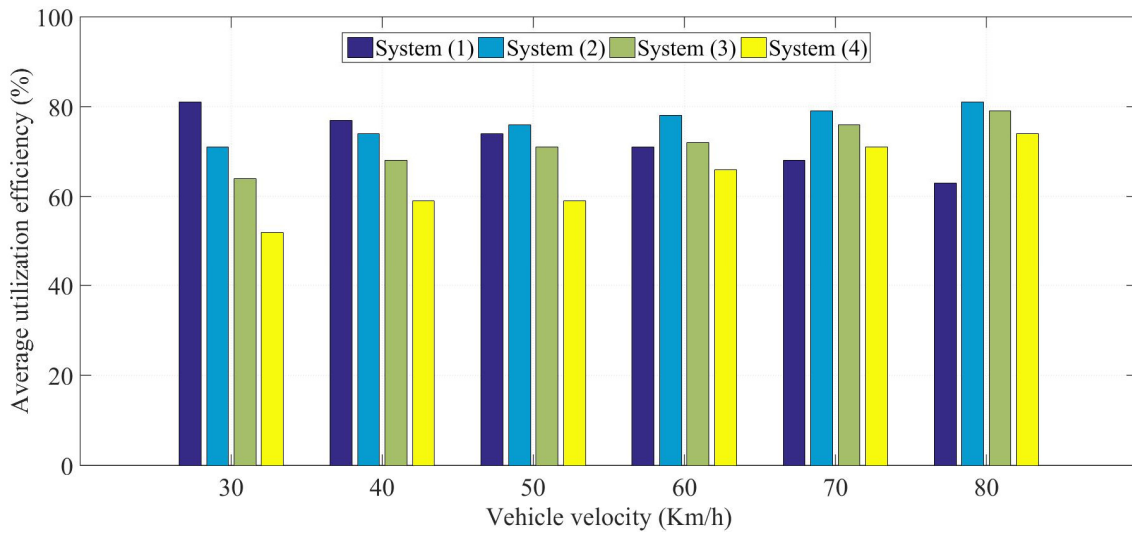


Figure 7. Average resource utilization efficiency at different vehicle mobility for Category (II) tasks.

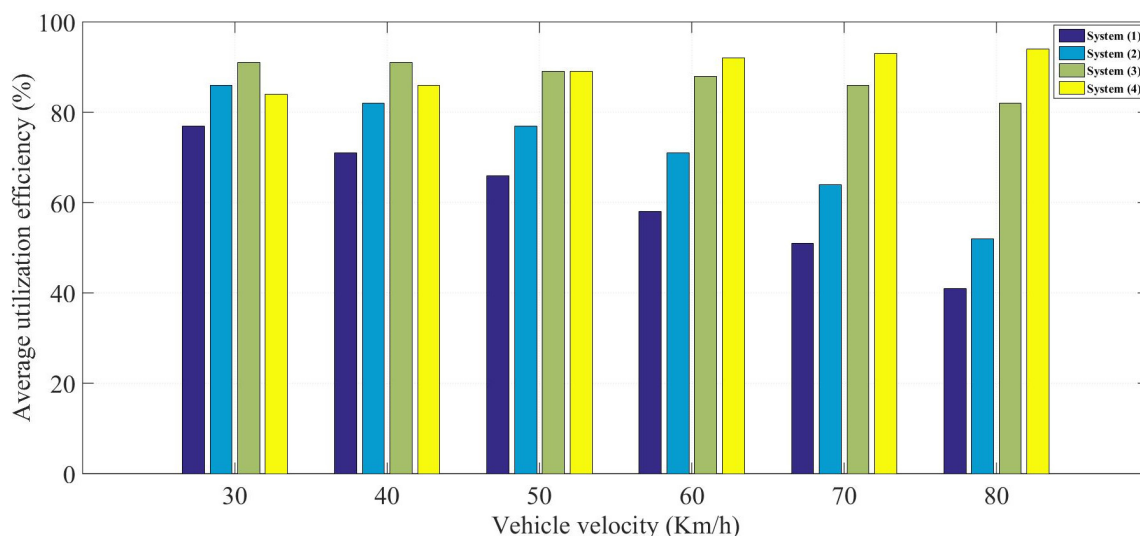


Figure 8. Average resource utilization efficiency at different vehicle mobility, for Category (III) tasks.

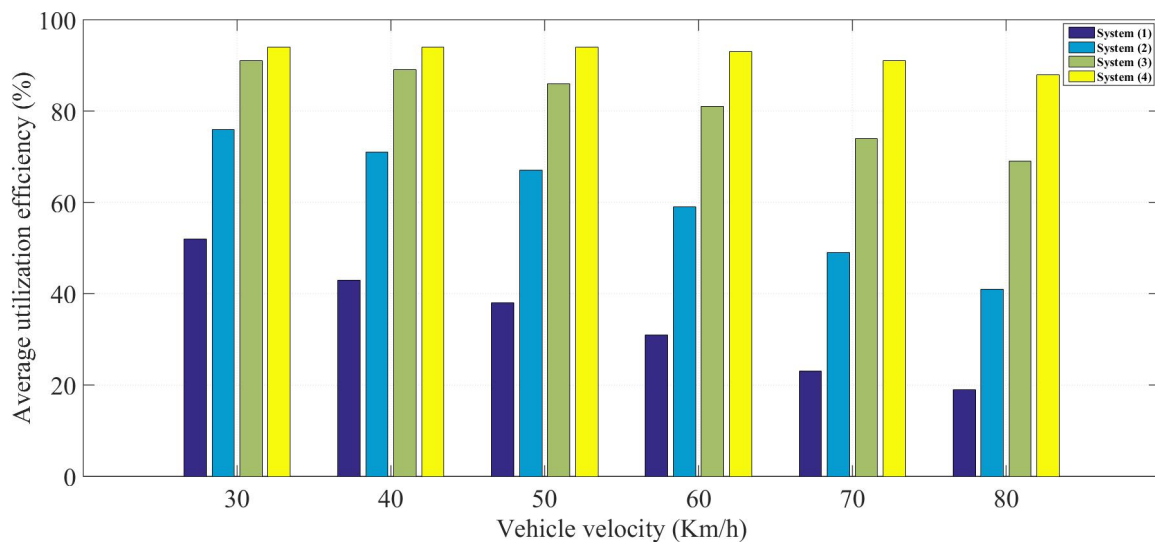


Figure 9. Average resource utilization efficiency at different vehicle mobility for Category (IV) tasks.

Figures 7 and 8 present the utilization efficiency for the four simulated systems when simulated using tasks from Category (II) and Category (III), respectively. With the increase in task complexity, moving from Category (I) to Category (II) and then to Category (III), the demands for higher computing resources and efficient use of such resources increase. The proposed microservice-based VANET achieved an average utilization of computing resources higher than other existing systems, mainly at high vehicle mobility. Also, this performance improvement increases in Figure 9, which provides the results of Category (IV). This is due to the complexity of tasks that need higher computing capabilities, achieved through introducing the fog layer and efficient use of resources, achieved using microservice and offloading approaches. Moreover, the proposed migration scheme increases efficiency in high vehicle mobilities.

Figures 10–13 present the resource utilization performance at different numbers of deployed vehicles, with the specifications introduced in Table 7. For tasks of Category (I), i.e., results introduced in Figure 10, the traditional VANET and traditional MEC-VANET provide efficient resources; however, the proposed system offers computing resources with capabilities higher than the system requires. This reduces the utilization efficiency in this case and for the Category (II) tasks introduced in Figure 11. For tasks of Categories (III and IV), the proposed system achieved higher utilization efficiency than other simulated systems, mainly for larger numbers of deployed vehicles. This can be interpreted in the same way as for Figures 8 and 9. Similarly, Figures 14–17 provide the obtained results of the resource utilization of the four simulated systems at different traffic densities for the specifications introduced in Table 8.

Table 7. Simulation parameters for utilization efficiency measurements (Figures 10–13).

Parameter	Traffic Density (TD)	Vehicle Velocity (v)	System (1)	System (2)	System (3)	System (4)
Value	0.1 veh/m	50 Km/h	Traditional VANET	Traditional MEC-VANET	System developed in [28]	Proposed microservice-based VANET

Table 8. Simulation parameters for utilization efficiency measurements (Figures 14–17).

Parameter	Number of Vehicles (N)	Vehicle Velocity (v)	System (1)	System (2)	System (3)	System (4)
Value	400	50 Km/h	Traditional VANET	Traditional MEC-VANET	System developed in [28]	Proposed microservice-based VANET

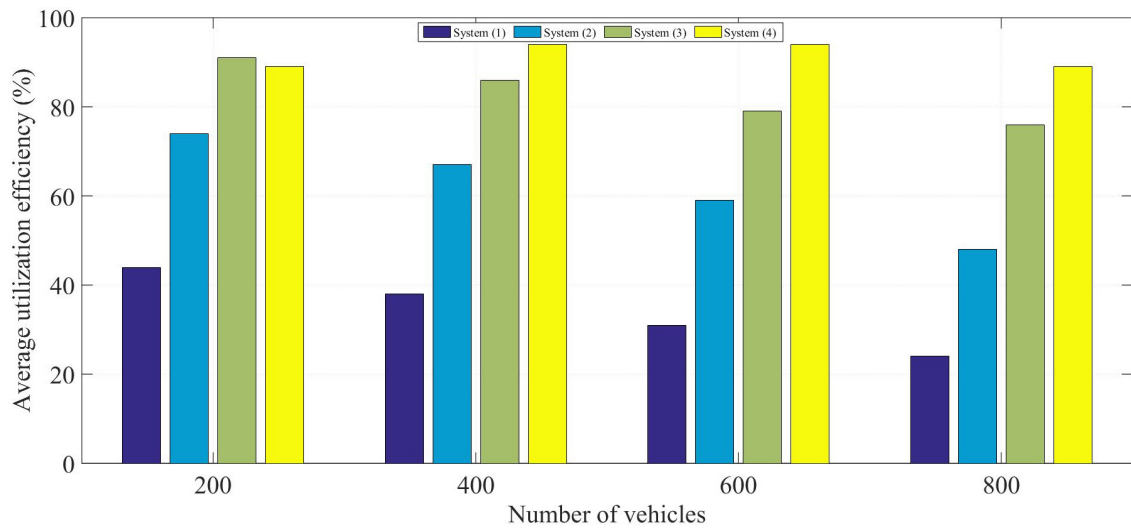


Figure 10. Average resource utilization efficiency at different numbers of vehicles for Category (I) tasks.

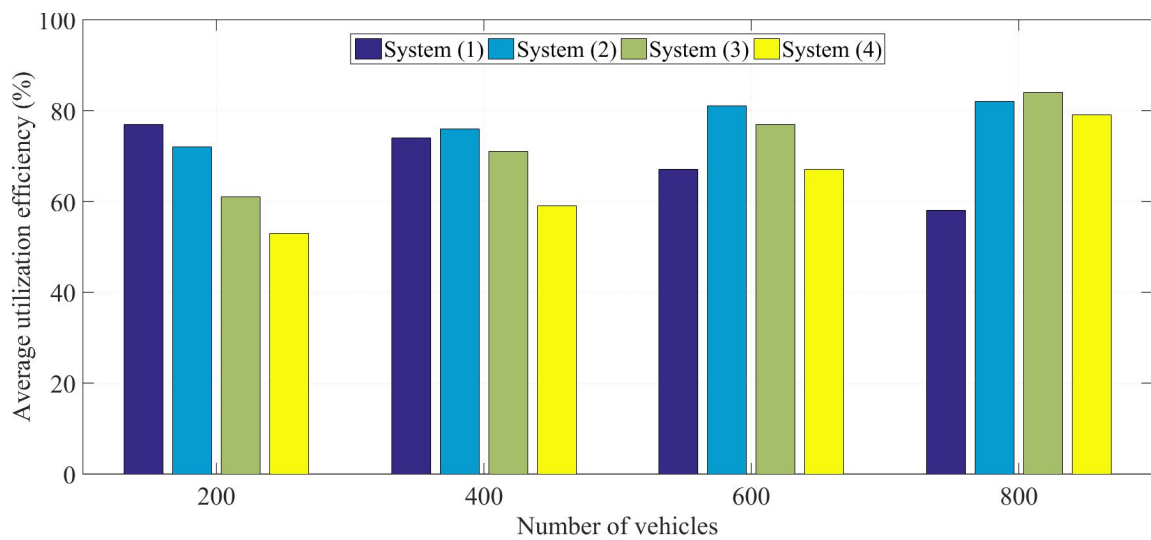


Figure 11. Average resource utilization efficiency at different numbers of vehicles for Category (II) tasks.

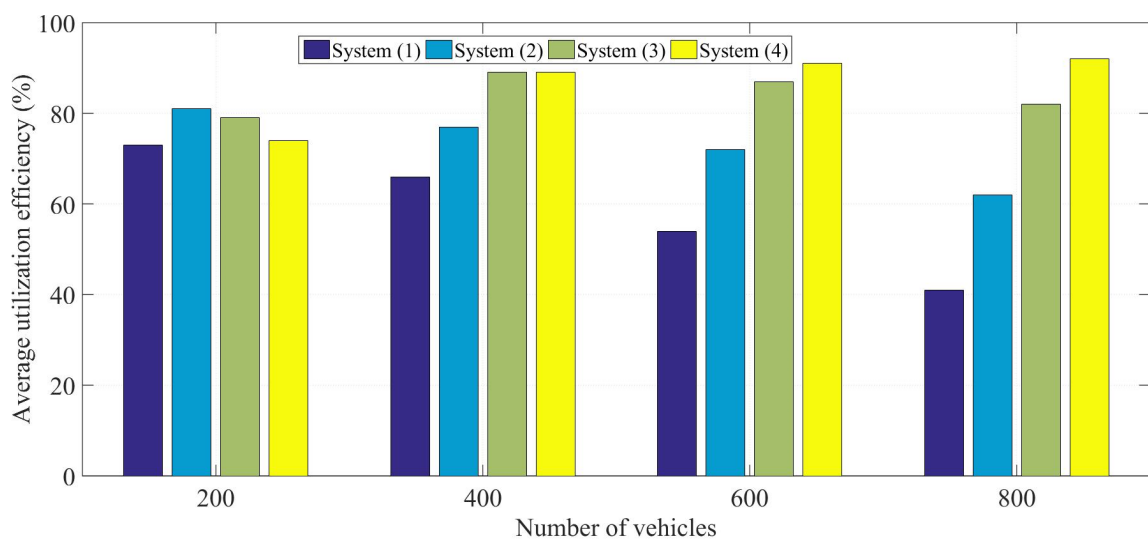


Figure 12. Average resource utilization efficiency at different numbers of vehicles for Category (III) tasks.

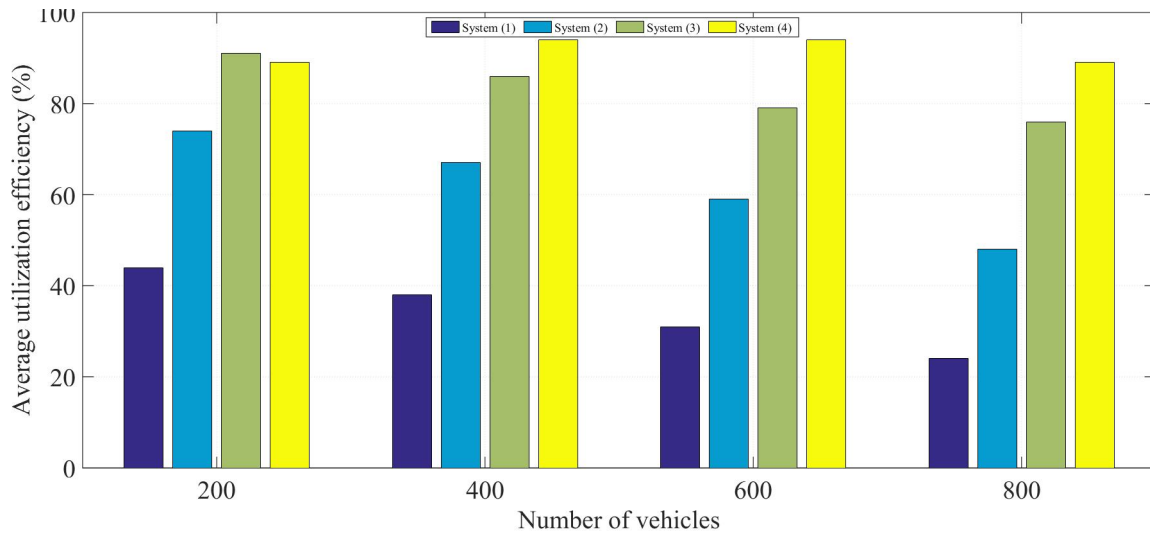


Figure 13. Average resource utilization efficiency at different numbers of vehicles for Category (IV) tasks.

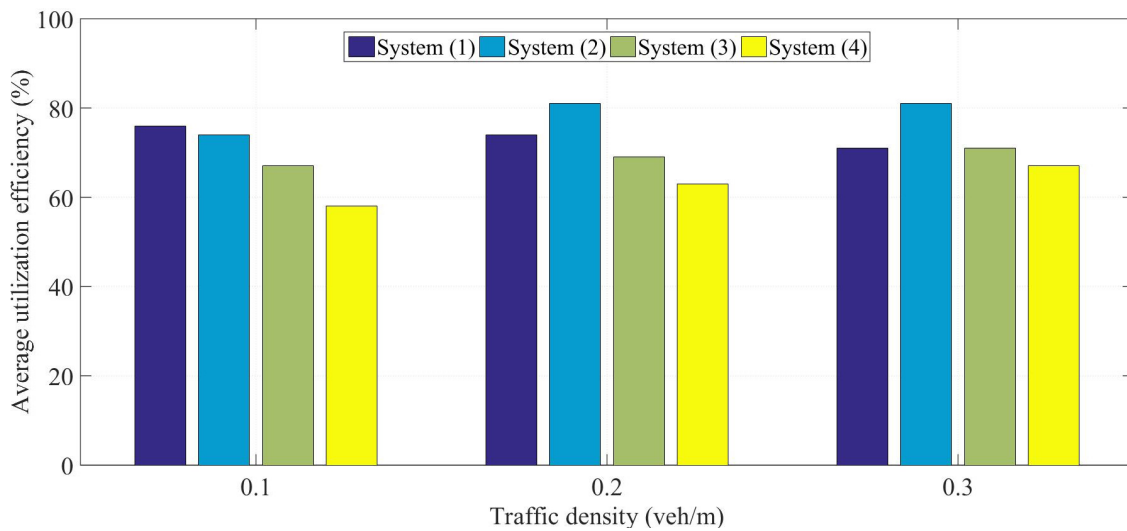


Figure 14. Average resource utilization efficiency at different TDs for Category (I) tasks.

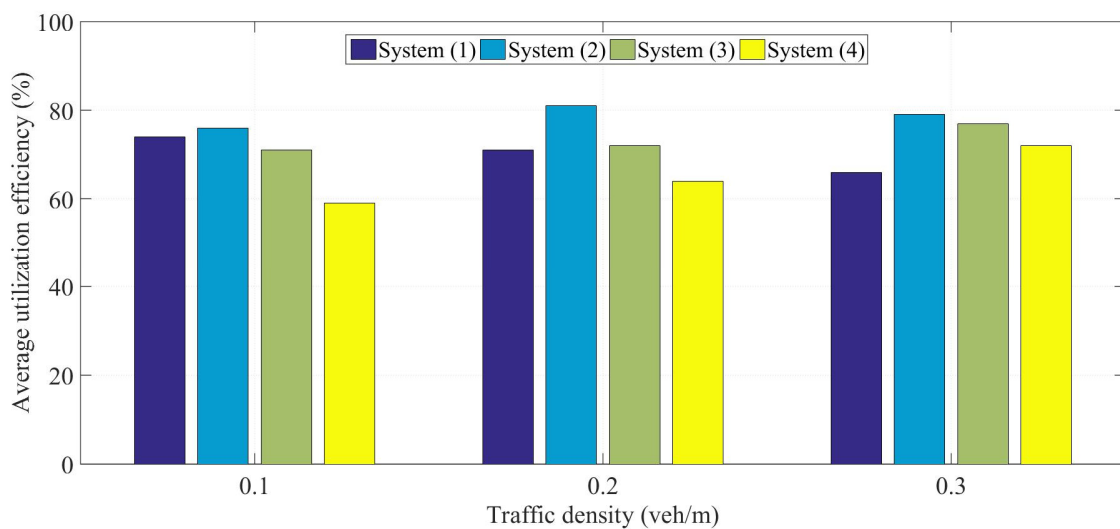


Figure 15. Average resource utilization efficiency at different TDs for Category (II) tasks.

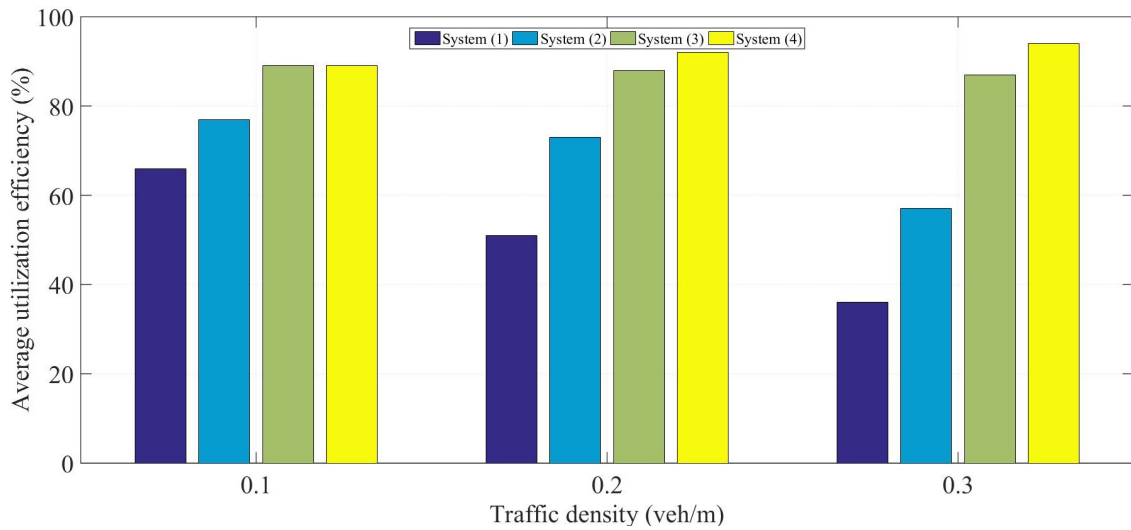


Figure 16. Average resource utilization efficiency at different TDs for Category (III) tasks.

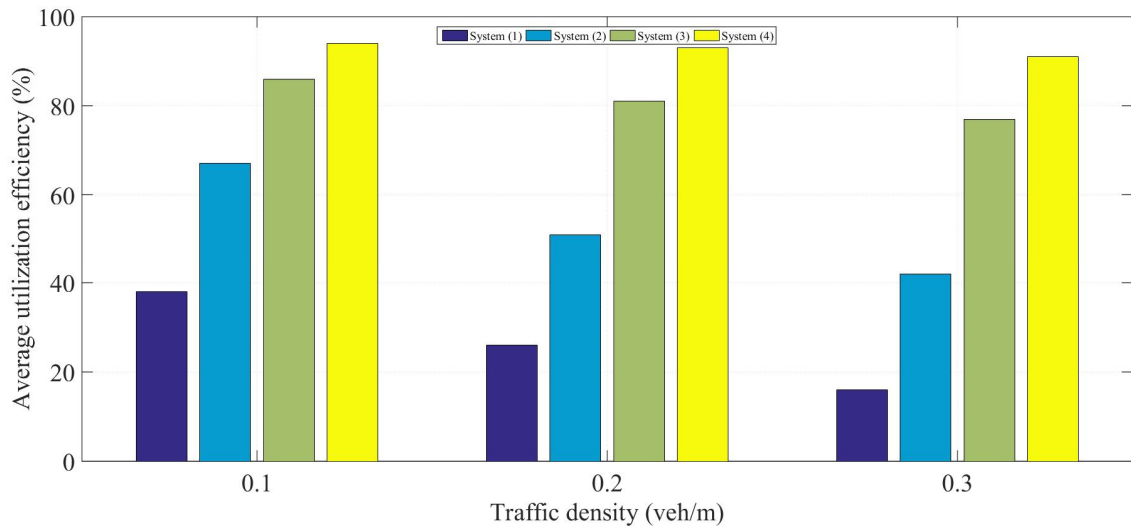


Figure 17. Average resource utilization efficiency at different TDs for Category (IV) tasks.

To evaluate the availability of the proposed microservice-based VANET, the average number of the blocked tasks, i.e., tasks that cannot be handled during simulation due to unavailability of resources or other reasons, was measured. The developed and existing systems used for comparison were simulated with Categories (III and IV) tasks. We considered these categories and neglected the first two categories since these tasks have constraints regarding QoS time. Tasks of the first two categories require low computing resources and have little constraints in terms of QoS time. Figures 18–23 present the average percentage of BT (%BT) compared to the total number of assigned tasks during the simulation. Figures 18 and 19 present the average %BT at different vehicle mobilities for the specifications introduced in Table 9. The proposed model outperformed the existing systems for the two categories of applications. This is due to the introduction of microservices and the increase of available resources introduced by fog nodes. Moreover, the system achieved higher performance at higher mobilities due to the proposed migration scheme with the microservice approach.

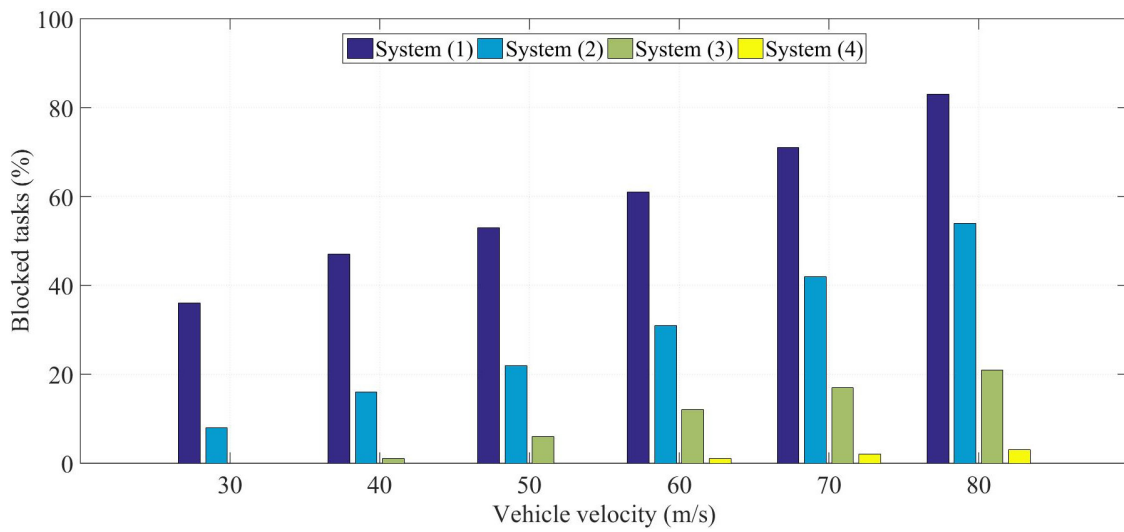


Figure 18. %BTs at different vehicle mobility for Category (III) tasks.

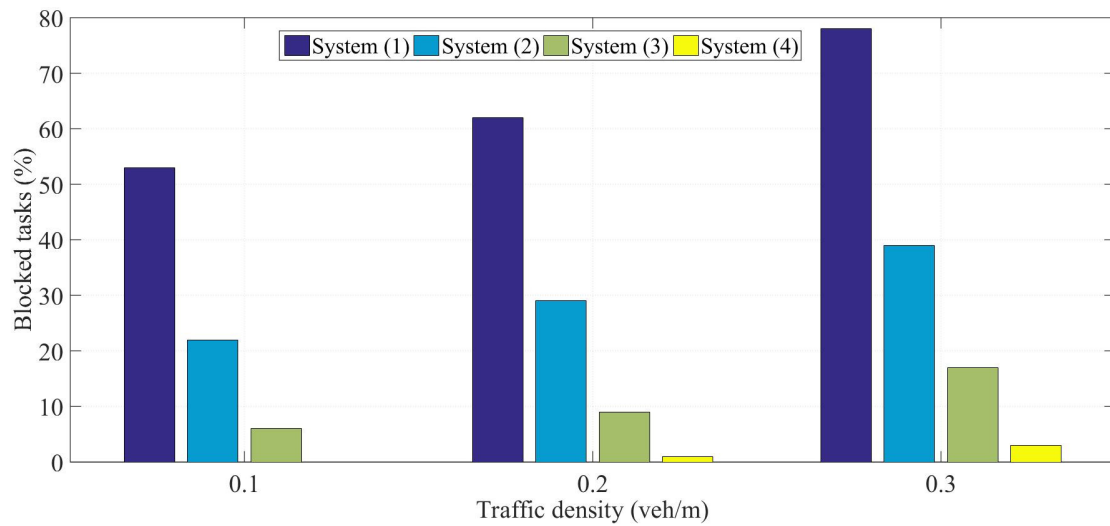


Figure 19. %BTs at different vehicle mobility for Category (IV) tasks.

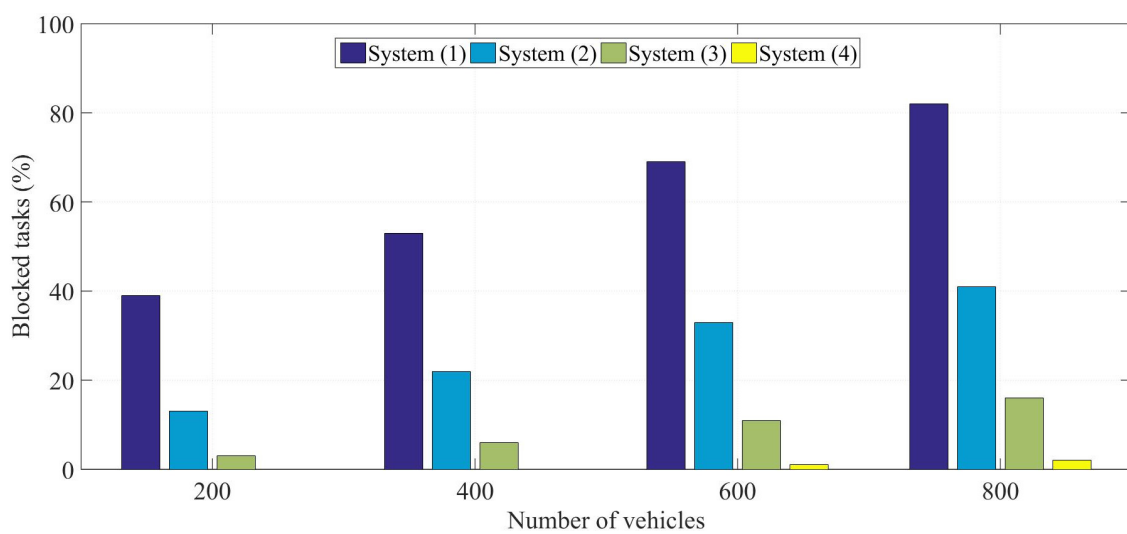


Figure 20. %BTs at different numbers of vehicles for Category (III) tasks.

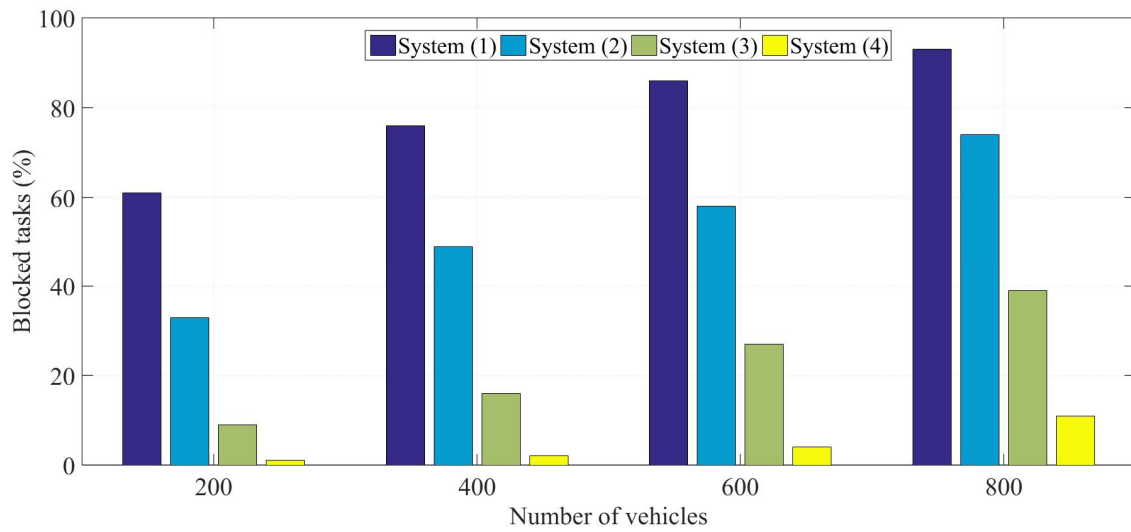


Figure 21. BTs at different numbers of vehicles for Category (IV) tasks.

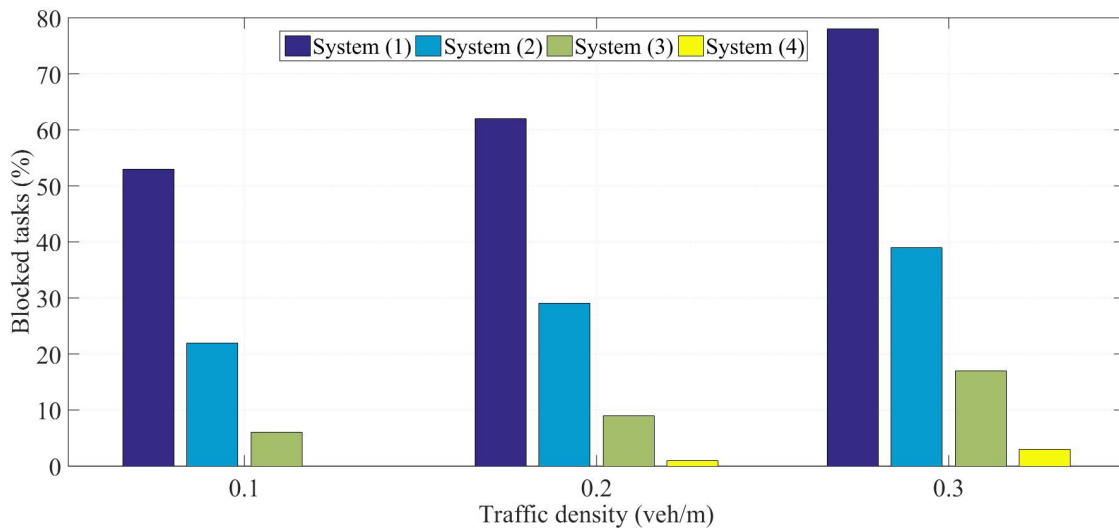


Figure 22. BTs at different TDs for Category (III) tasks.

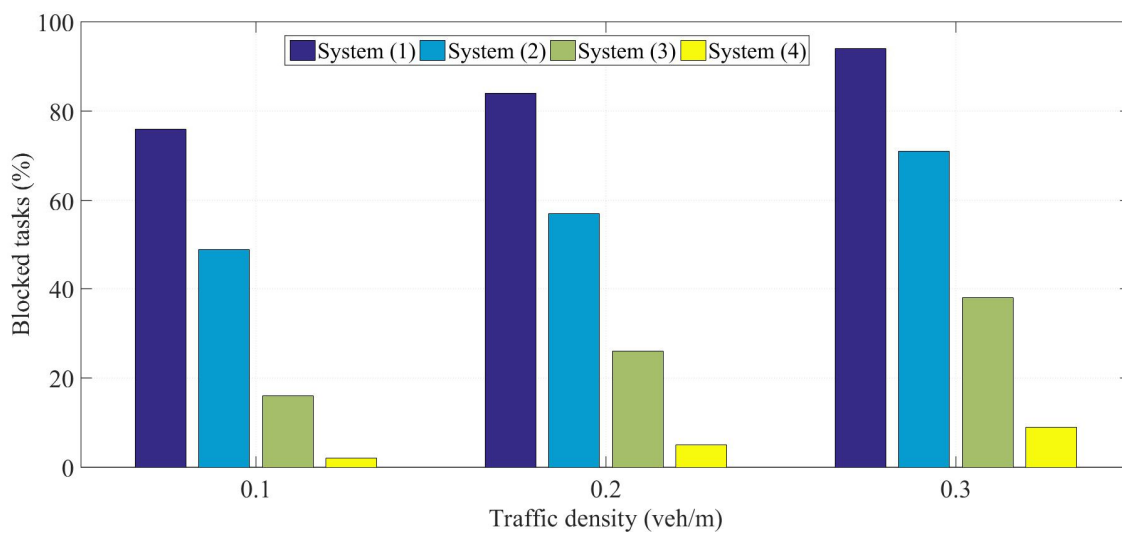


Figure 23. BTs at different TDs for Category (IV) tasks.

Table 9. Simulation parameters for the %BT measurements (Figures 18 and 19).

Parameter	Number of Vehicles (N)	Traffic Density (TD)	QoS Time (Category III) Tasks	QoS Time (Category IV) Tasks	System (1)	System (2)	System (3)	System (4)
Value	400	0.1 veh/m	10 ms	5 ms	Traditional VANET	Traditional MEC-VANET	System developed in [28]	Proposed microservice-based VANET

Figures 20 and 21 present the average %BT for different numbers of employed vehicles for the specifications introduced in Table 10. The proposed model outperformed the existing systems for the two categories of applications. This is mainly due to a higher number of deployed vehicles due to the introduction of microservice and the increase of available resources introduced by fog nodes. Similarly, Figures 22 and 23 present the average %BT at different traffic densities for the specifications introduced in Table 11.

Table 10. Simulation parameters for the %BT measurements (Figures 20 and 21).

Parameter	Vehicle Velocity (v)	Traffic Density (TD)	QoS Time (Category III) Tasks	QoS Time (Category IV) Tasks	System (1)	System (2)	System (3)	System (4)
Value	50 Km/h	0.1 veh/m	10 ms	5 ms	Traditional VANET	Traditional MEC-VANET	System developed in [28]	Proposed microservice-based VANET

Table 11. Simulation parameters for the %BT measurements (Figures 22 and 23).

Parameter	Vehicle Velocity (v)	Number of Vehicles (N)	QoS Time (Category III) Tasks	QoS Time (Category IV) Tasks	System (1)	System (2)	System (3)	System (4)
Value	50 Km/h	400	10 ms	5 ms	Traditional VANET	Traditional MEC-VANET	System developed in [28]	Proposed microservice-based VANET

The average latency for handling computing tasks was measured during the simulation. The performance improvement in terms of latency was calculated to the traditional MEC/VANET, using Equation (1). This performance measure was considered for the third and fourth categories of applications since the first and second application categories have no constraints in terms of latency. Figures 24 and 25 present the percentage improvement of the latency for the systems and specifications mentioned in Table 12. Figure 24 presents the performance improvement of latency for tasks of Category (III) of applications at different numbers of vehicles, and similarly, Figure 25 provides the results for tasks of Category (IV). The proposed system outperformed the existing systems and reduced the average latency required to handle computing tasks of Categories (III and IV).

Table 12. Simulation parameters for the percentage improvement of latency measurements (Figures 24 and 25).

Parameter	Vehicle Velocity (v)	Traffic Density (TD)	QoS Time (Category III) Tasks	QoS Time (Category IV) Tasks	System (2)	System (3)
Value	50 Km/h	0.1 veh/m	10 ms	5 ms	System developed in [28]	Proposed microservice-based VANET

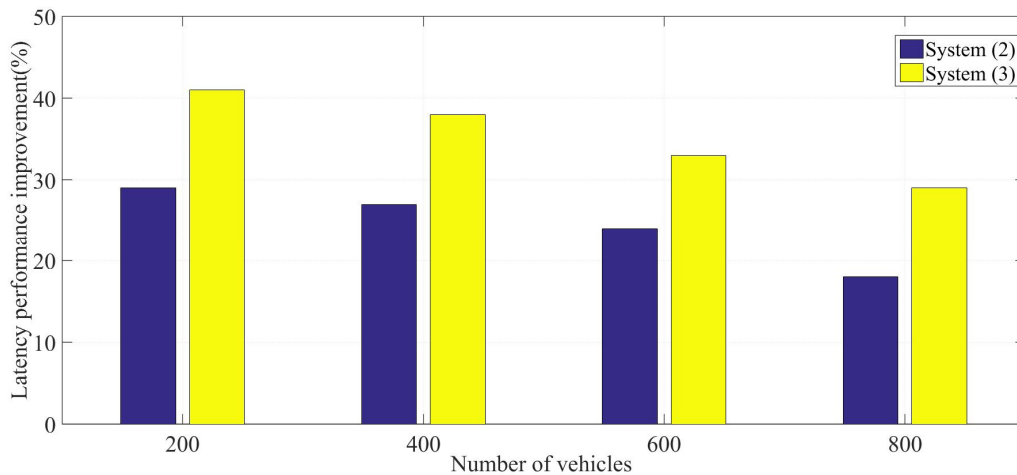


Figure 24. Latency performance improvement at different numbers of vehicles for Category (III) tasks.

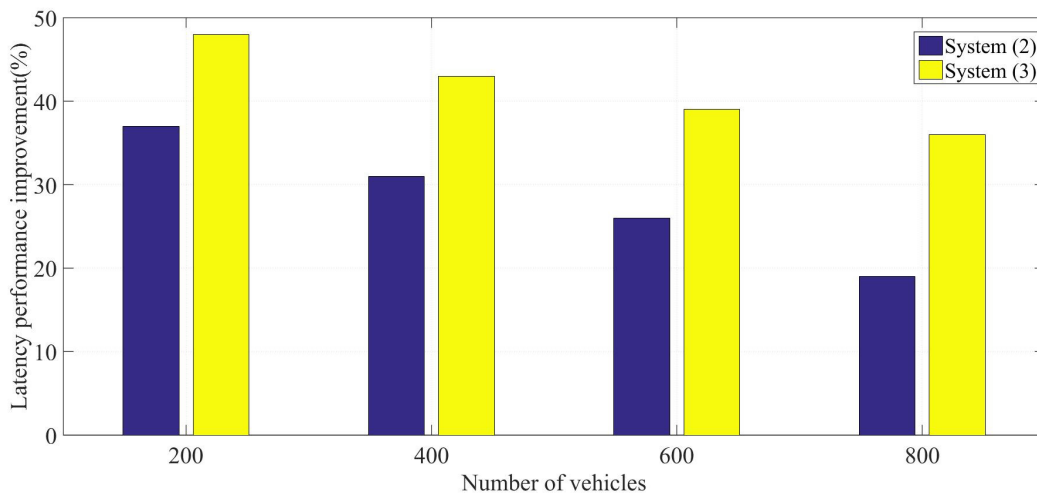


Figure 25. Latency performance improvement at different numbers of vehicles for Category (IV) tasks.

9. Conclusions

This work investigated the major challenges presented by VANETs’ demanding needs, including extremely low latency, exceptional reliability, and seamless communications at high mobilities. It introduced a novel structure for VANETs based on fog computing, MEC, and microservices architecture. Utilizing this two-layer edge computing approach improves the network’s capacity to handle computational activities and maintain minimal latency effectively. Introducing the microservice architecture with prioritizing tasks provided improved flexibility, modularity, and scalability in the design and implementation of automotive applications. An efficient offloading scheme was implemented to distribute the computing workload efficiently between vehicles and the two-level edge servers, resulting in a significant decrease in system delay, improved resource utilization efficiency, and higher network availability. In addition, a seamless migration scheme was developed and implemented with an efficient HO strategy to provide seamless communications with the intended QoS at high vehicle mobility. The proposed microservice-based VANET was simulated for heterogeneous real-based scenarios considering different TDs, different numbers of deployed vehicles, and different vehicle mobilities. The results indicated that the proposed system outperformed existing VANET systems regarding latency, availability, and reliability.

Author Contributions: Conceptualization, M.M.Z., A.A.A., M.M.A. and M.S.S.; methodology, M.M.Z., A.A.A., M.M.A., M.E. and M.S.S.; software, M.M.Z., A.A.A., M.M.A. and M.S.S.; formal analysis, M.M.Z., A.A.A., M.M.A. and M.S.S.; investigation, A.A.A., M.M.A. and M.S.S.; resources, A.A.A., M.M.A. and M.S.S.; writing—original draft preparation, M.M.Z., A.A.A., M.M.A. and M.S.S.; writing—review and editing, A.A.A., M.M.A., M.E. and M.S.S.; visualization, M.M.Z.; supervision M.S.S.; project administration, A.A.A.; funding, M.E. and A.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Prince Sultan University.

Data Availability Statement: The data are contained within the article and/or available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to acknowledge the support of Prince Sultan University for paying the article processing charges (APC) of this publication.

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

References

1. Chaccour, C.; Saad, W.; Debbah, M.; Han, Z.; Poor, H.V. Less Data, More Knowledge: Building next Generation Semantic Communication Networks. *IEEE Commun. Surv. Tutor.* **2024**, 1–11. [[CrossRef](#)]
2. Hakak, S.; Gadekallu, T.R.; Maddikunta, P.K.R.; Ramu, S.P.; M, P.; De Alwis, C.; Liyanage, M. Autonomous Vehicles in 5G and beyond: A Survey. *Veh. Commun.* **2023**, *39*, 100551. [[CrossRef](#)]
3. Liu, W.; Hua, M.; Deng, Z.; Meng, Z.; Huang, Y.; Hu, C.; Song, S.; Gao, L.; Liu, C.; Shuai, B.; et al. A Systematic Survey of Control Techniques and Applications in Connected and Automated Vehicles. *IEEE Internet Things J.* **2023**, *10*, 21892–21916. [[CrossRef](#)]
4. Mahi, M.J.N.; Chaki, S.; Ahmed, S.; Biswas, M.; Kaiser, M.S.; Islam, M.S.; Sookhak, M.; Barros, A.; Whaiduzzaman, M. A Review on VANET Research: Perspective of Recent Emerging Technologies. *IEEE Access* **2022**, *10*, 65760–65783. [[CrossRef](#)]
5. Guo, H.; Zhou, X.; Liu, J.; Zhang, Y. Vehicular Intelligence in 6G: Networking, Communications, and Computing. *Veh. Commun.* **2022**, *33*, 100399. [[CrossRef](#)]
6. Javed, F.; Khan, Z.A.; Rizwan, S.; Shahzadi, S.; Chaudhry, N.R.; Iqbal, M. A Novel Energy-Efficient Reservation System for Edge Computing in 6G Vehicular Ad Hoc Network. *Sensors* **2023**, *23*, 5817. [[CrossRef](#)] [[PubMed](#)]
7. Nurkahfi, G.N.; Triwinarko, A.; Prawara, B.; Armi, N.; Juhana, T.; Syambas, N.R.; Mulyana, E.; Dogheche, E.; Dayoub, I. On SDN to Support the IEEE 802.11 and C-V2X Based Vehicular Communications Use-Cases and Performance: A Comprehensive Survey. *IEEE Access.* [[CrossRef](#)]
8. Behravan, K.; Farzaneh, N.; Jahanshahi, M.; Hosseini Seno, S.A. A Comprehensive Survey on Using Fog Computing in Vehicular Networks. *Veh. Commun.* **2023**, *42*, 100604. [[CrossRef](#)]
9. Wakgra, F.G.; Kar, B.; Tadele, S.B.; Shen, S.-H.; Khan, A.U. Multi-Objective Offloading Optimization in MEC and Vehicular-Fog Systems: A Distributed-TD3 Approach. *IEEE Trans. Intell. Transp. Syst.* **2024**, 1–13. [[CrossRef](#)]
10. Waheed, A.; Shah, M.A.; Mohsin, S.M.; Khan, A.; Maple, C.; Aslam, S.; Shamsirband, S. A Comprehensive Review of Computing Paradigms, Enabling Computation Offloading and Task Execution in Vehicular Networks. *IEEE Access* **2022**, *10*, 3580–3600. [[CrossRef](#)]
11. Hamdi, A.M.A.; Hussain, F.K.; Hussain, O.K. Task Offloading in Vehicular Fog Computing: State-of-the-Art and Open Issues. *Future Gener. Comput. Syst.* **2022**, *133*, 201–212. [[CrossRef](#)]
12. Nkenyereye, L.; Naik, R.P.; Jang, J.W.; Chung, W.Y. Software-Defined Small Cell-Linked Vehicular Networks: Architecture and Evaluation. *Electronics* **2023**, *12*, 304. [[CrossRef](#)]
13. Taha, M.B.; Alrabaee, S.; Choo, K.K.R. Efficient Resource Management of Micro-Services in Vanets. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 6820–6835. [[CrossRef](#)]
14. Alam, F.; Toosi, A.N.; Cheema, M.A.; Cicconetti, C.; Serrano, P.; Iosup, A.; Tari, Z.; Sarvi, M. Serverless Vehicular Edge Computing for the Internet of Vehicles. *IEEE Internet Comput.* **2023**, *27*, 40–51. [[CrossRef](#)]
15. Alkaabi, S.R.; Gregory, M.A.; Li, S. Multi-Access Edge Computing Handover Strategies, Management, and Challenges: A Review. *IEEE Access* **2024**, *12*, 4660–4673. [[CrossRef](#)]
16. Zakarya, M.; Gillam, L.; Khan, A.A.; Rana, O.; Buyya, R. ApMove: A Service Migration Technique for Connected and Autonomous Vehicles. *IEEE Internet Things J.* **2024**, *1*. [[CrossRef](#)]
17. Ge, X. Ultra-Reliable Low-Latency Communications in Autonomous Vehicular Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5005–5016. [[CrossRef](#)]
18. Abbas, F.; Fan, P. Clustering-Based Reliable Low-Latency Routing Scheme Using ACO Method for Vehicular Networks. *Veh. Commun.* **2018**, *12*, 66–74. [[CrossRef](#)]
19. Gomez-Andrades, A.; Munoz, P.; Serrano, I.; Barco, R. Automatic Root Cause Analysis for LTE Networks Based on Unsupervised Techniques. *IEEE Trans. Veh. Technol.* **2016**, *65*, 2369–2386. [[CrossRef](#)]
20. Vladyko, A.; Khakimov, A.; Muthanna, A.; Ateya, A.A.; Koucheryavy, A. Distributed Edge Computing to Assist Ultra-Low-Latency VANET Applications. *Future Internet* **2019**, *11*, 128. [[CrossRef](#)]

21. Boukerche, A.; Aljeri, N. An Energy-Efficient Controller Management Scheme for Software-Defined Vehicular Networks. *IEEE Trans. Sustain. Comput.* **2022**, *7*, 61–74. [[CrossRef](#)]
22. Aljeri, N.; Boukerche, A. A Novel Proactive Controller Deployment Protocol for 5G-Enabled Software-Defined Vehicular Networks. *Comput. Commun.* **2022**, *182*, 88–97. [[CrossRef](#)]
23. Dai, P.; Hu, K.; Wu, X.; Xing, H.; Teng, F.; Yu, Z. A Probabilistic Approach for Cooperative Computation Offloading in MEC-Assisted Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 899–911. [[CrossRef](#)]
24. Peng, H.; Shen, X.S. DDPG-Based Resource Management for MEC/UAV-Assisted Vehicular Networks. In Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Victoria, BC, Canada, 18 November–16 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
25. Muthanna, A.; Shamilova, R.; Ateya, A.A.; Paramonov, A.; Hammoudeh, M. A Mobile Edge Computing/Software-defined Networking-enabled Architecture for Vehicular Networks. *Internet Technol. Lett.* **2020**, *3*, e109. [[CrossRef](#)]
26. Qiao, B.; Liu, C.; Liu, J.; Hu, Y.; Li, K.; Li, K. Task Migration Computation Offloading with Low Delay for Mobile Edge Computing in Vehicular Networks. *Concurr. Comput.* **2022**, *34*, e6494. [[CrossRef](#)]
27. Vladyko, A.; Elagin, V.; Spirikina, A.; Muthanna, A.; Ateya, A. Distributed Edge Computing with Blockchain Technology to Enable Ultra-Reliable Low-Latency V2X Communications. *Electronics* **2022**, *11*, 173. [[CrossRef](#)]
28. Monir, N.; Toraya, M.M.; Vladyko, A.; Muthanna, A.; Torad, M.A.; El-Samie, F.E.A.; Ateya, A.A. Seamless Handover Scheme for MEC/SDN-Based Vehicular Networks. *J. Sens. Actuator Netw.* **2022**, *11*, 9. [[CrossRef](#)]
29. Aljeri, N.; Boukerche, A. Performance Evaluation of Communication Lifetime Prediction Model for Autonomous Vehicular Networks. In *PE-WASUN '23, Proceedings of the Int'l ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, Montreal, QC, Canada, 30 October–3 November 2023*; ACM: New York, NY, USA, 2023.
30. Khan, Z.; Koubaa, A.; Benjdira, B.; Boulila, W. A Game Theory Approach for Smart Traffic Management. *Comput. Electr. Eng.* **2023**, *110*, 108825. [[CrossRef](#)]
31. Martins Maciel, P.R. *Performance, Reliability, and Availability Evaluation of Computational Systems, Volume I: Performance and Background*; Chapman & Hall/CRC: Philadelphia, PA, USA, 2023; ISBN 9781032295374.
32. De Simone, L.; Di Mauro, M.; Longo, M.; Natella, R.; Postiglione, F. Performability Assessment of Containerized Multi-Tenant IMS through Multidimensional UGF. In Proceedings of the 2022 18th International Conference on Network and Service Management (CNSM), Thessaloniki, Greece, 31 October–4 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 145–153.
33. Simone, L.D.; Mauro, M.D.; Natella, R.; Postiglione, F. Performance and Availability Challenges in Designing Resilient 5G Architectures. *IEEE Trans. Netw. Serv. Manag.* **2024**, *1*. [[CrossRef](#)]
34. Aljeri, N.; Boukerche, A. An Adaptive Traffic-Flow Based Controller Deployment Scheme for Software-Defined Vehicular Networks. In *MSWiM '20, Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Alicante, Spain, 16–20 November 2020*; ACM: New York, NY, USA, 2020.
35. Garg, S.; Kaur, K.; Kaddoum, G.; Ahmed, S.H.; Jayakody, D.N.K. SDN-Based Secure and Privacy-Preserving Scheme for Vehicular Networks: A 5G Perspective. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8421–8434. [[CrossRef](#)]
36. Fung, C.C.; Yogarayan, S.; Abdul Razak, S.F.; Azman, A. A Review Study of IEEE 802.11p on-Board Unit for V2X Deployment. In Proceedings of the 2023 11th International Conference on Information and Communication Technology (ICoICT), Melaka, Malaysia, 23–24 August 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 165–171.
37. Kumar, D. Rammohan Revolutionizing Intelligent Transportation Systems with Cellular Vehicle-to-Everything (C-V2X) Technology: Current Trends, Use Cases, Emerging Technologies, Standardization Bodies, Industry Analytics and Future Directions. *Veh. Commun.* **2023**, *43*, 100638. [[CrossRef](#)]
38. Fernandez, Z.; Martin, A.; Perez, J.; Garcia, M.; Velez, G.; Murciano, F.; Peters, S. Challenges and Solutions for Service Continuity in Inter-PLMN Handover for Vehicular Applications. *IEEE Access* **2023**, *11*, 8904–8919. [[CrossRef](#)]
39. K, S.; Chinnasamy, C. Efficient VANET Handover Scheme Using SSDN by Incorporating Media Independent Handover Framework. *Measur. Sens.* **2023**, *26*, 100684. [[CrossRef](#)]
40. Khan, N.; bin Salleh, R.; Khan, Z.; Koubaa, A.; Hamdan, M.; Abdelmoniem, A.M. Ensuring Reliable Network Operations and Maintenance: The Role of PMRF for Switch Maintenance and Upgrades in SDN. *J. King Saud Univ.—Comput. Inf. Sci.* **2023**, *35*, 101809. [[CrossRef](#)]
41. Ateya, A.A.; Muthanna, A.; Koucheryavy, A.; Maleh, Y.; El-Latif, A.A.A. Energy Efficient Offloading Scheme for MEC-Based Augmented Reality System. *Clust. Comput.* **2023**, *26*, 789–806. [[CrossRef](#)]
42. Ms-Van3t. Available online: <https://github.com/ms-van3t-devs/ms-van3t> (accessed on 15 April 2024).
43. Ateya, A.A.; Vybornova, A.; Samouylov, K.; Koucheryavy, A. System Model for Multi-Level Cloud Based Tactile Internet System. In *Wired/Wireless Internet Communications, Proceedings of the 15th IFIP WG 6.2 International Conference, WWIC 2017, St. Petersburg, Russia, 21–23 June 2017*; Springer International Publishing: Cham, Switzerland, 2017; pp. 77–86. ISBN 9783319613819.
44. Cruz, S.; Aguiar, A. Cooperative Localization in Vehicular Networks Dataset 2020. Available online: <https://iee-dataport.org/open-access/cooperative-localization-vehicular-networks-dataset> (accessed on 1 June 2024).
45. From Images to 3D Shapes (FI3S). Available online: <https://www.kaggle.com/datasets/lehomme/from-images-to-3d-shapesfi3s> (accessed on 20 April 2024).
46. TikTokDataset. Available online: <https://www.kaggle.com/datasets/yasaminjafarian/tiktokdataset> (accessed on 20 April 2024).

-
47. YouTube Faces with Facial Keypoints. Available online: <https://www.kaggle.com/datasets/selfishgene/youtube-faces-with-facial-keypoints> (accessed on 20 April 2024).
 48. Google Scraped Image Dataset. Available online: <https://www.kaggle.com/datasets/duttadebadri/image-classification> (accessed on 20 April 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.