vehicles

Article

A Fuzzy-Based Approach for the Assessment of the Edge Layer Processing Capability in SDN-VANETs: A Comparation Study of Testbed and Simulation System Results

Ermioni Qafzezi [1,*], Kevin Bylykbashi [1], Shunya Higashi [2], Phudit Ampririt [2], Keita Matsuo [1] and Leonard Barolli [1]

1 Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan; kevin@bene.fit.ac.jp (K.B.); kt-matsuo@fit.ac.jp (K.M.); barolli@fit.ac.jp (L.B.)
2 Graduate School of Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan; 5972.sese@gmail.com (S.H.); bd21201@bene.fit.ac.jp (P.A.)
* Correspondence: qafzezi@bene.fit.ac.jp

Abstract: Vehicular Ad Hoc Networks (VANETs) have gained significant attention due to their potential to enhance road safety, traffic efficiency, and passenger comfort through vehicle-to-vehicle and vehicle-to-infrastructure communication. However, VANETs face resource management challenges due to the dynamic and resource constrained nature of vehicular environments. Integrating cloud-fog-edge computing and Software-Defined Networking (SDN) with VANETs can harness the computational capabilities and resources available at different tiers to efficiently process and manage vehicular data. In this work, we used this paradigm and proposed an intelligent approach based on Fuzzy Logic (FL) to evaluate the processing and storage capability of vehicles for helping other vehicles in need of additional resources. The effectiveness of the proposed system is evaluated through extensive simulations and a testbed. Performance analysis between the simulation results and the testbed offers a comprehensive understanding of the proposed system and its performance and feasibility.

Keywords: connected vehicles; fuzzy logic; resource management; edge computing; VANETs

check for updates

Citation: Qafzezi, E.; Bylykbashi, K.; Higashi, S.; Ampririt, P.; Matsuo, K.; Barolli, L. A Fuzzy-Based Approach for the Assessment of the Edge Layer Processing Capability in SDN-VANETs: A Comparation Study of Testbed and Simulation System Results. *Vehicles* **2023**, *5*, 1087–1103. https://doi.org/10.3390/vehicles5030059

Academic Editor: Filippo Cianetti

Received: 31 July 2023
Revised: 30 August 2023
Accepted: 31 August 2023
Published: 3 September 2023

1. Introduction

Large cities currently face a significant obstacle in the form of congestion and traffic jams. The European Commission's report highlights that road congestion in Europe can result in up to 1% of the GDP being impacted [1]. Some countries, such as the United Kingdom, experience more pronounced issues, with traffic congestion costing them 24.5 billion EUR, equivalent to 1.6% of their GDP, one of the highest rates in the region. Metropolitan areas in Japan with over one million residents also bear the burden of congestion costs, amounting to approximately 463.8 billion JPY, which represents over 61% of the total cost of vehicular transport [2]. Likewise, the United States annually incurs a substantial expense of around 190 billion USD due to traffic congestion [3]. This issue not only carries financial implications, but also leads to wasted time, environmental pollution, noise, and, most importantly, traffic accidents.

The statistics surrounding traffic accidents are even more alarming, with approximately 1.3 million people worldwide losing their lives each year in such incidents [4]. To address these concerns and improve safety and efficiency during travel, Vehicular Ad Hoc Networks (VANETs) have emerged [5]. VANETs assist drivers in avoiding collisions by providing real-time information about routes, thereby enhancing travel safety, efficiency, and convenience. While popular vehicular navigation systems such as Google Maps recommend alternative routes to avoid traffic congestion based on factors such as driving distance, time, and cost, their primary focus remains on serving individual drivers. In contrast, VANETs aim for more comprehensive goals by benefiting all road users through content awareness and inter-vehicle communication. Inter-vehicle communication allows for a broader understanding of the state of other vehicles in the network and the surrounding environment, including details about traffic lights, weather conditions, and public safety. Moreover, VANETs significantly reduce response time by promptly notifying vehicles about potential situations, even predicting accidents based on route capacity and expected traffic.

Although VANETs offers numerous possibilities, there are still various challenges that need to be addressed. Some of these challenges include the absence of a centralized management and coordination entity, meeting strict delay constraints and ensuring quality of service (QoS), addressing security and privacy concerns, accommodating various communication environments, such as urban, inter-urban, and highways, ensuring interoperability among heterogeneous wireless technologies, and effectively managing the abundant information and resources within VANETs. The increasing number of vehicles generating substantial data further complicates network management, while the emergence of new resource-intensive applications adds to the complexity [6,7]. To tackle these challenges, an intelligent architecture based on Fuzzy Logic (FL) and Software-Defined Networking (SDN) approaches is proposed. This architecture aims to efficiently manage Cloud-Fog-Edge (CFE) storage, computing, and networking resources in VANETs by leveraging FL for real-time resource management, while accounting for imprecision and uncertainty. However, in this work, we focus on the edge layer in an SDN-VANETs environment. The edge layer in a connected vehicles environment refers to the decentralized computing infrastructure that leverages the computing capability of vehicles themselves. In this context, vehicles act as edge computing units with the ability to share their computing resources, such as processing power, storage, and communication capabilities, with each other. The proposed system called Fuzzy-based System for Assessing Edge Layer Capability (FSA-ELC), is designed to evaluate the edge computing capabilities of neighboring vehicles in a vehicular network. By analyzing factors such as available storage, processing resources, and predicted contact duration of vehicles within its vicinity, the system aims to identify potential candidates to optimize data offloading, enhance vehicular services, and improve overall network performance. The proposed system's effectiveness is assessed using two main methods: extensive simulations and the implementation of a testbed. Through simulations, various scenarios and conditions are emulated in a controlled environment to observe the system's behavior. The testbed, on the other hand, involves physically implementing the system to evaluate its performance in real-world conditions. The main contributions are briefly summarized as follows:

- Our proposed approach considers a cloud-fog-edge layered architecture consisting of different capabilities and makes use of an integrated fuzzy-based system implemented in the SDN controllers.
- We implement and show the performance evaluation of a resource management system, named FSA-ELC, which assesses the edge layer's processing and storage capacity, and is composed of all the nearby vehicles within the communication range.
- Implementation of the proposed resource management system in a testbed.
- Comparison of simulation results with experimental results.

The subsequent sections of this article are structured as follows. In Section 2, we provide a comprehensive background overview of key concepts, namely the Internet of Things (IoT), VANETs, and cloud-fog-edge computing, focusing on their integration in the context of SDN-VANETs. We also provide an overview of related works in the field, exploring research and studies that contribute to the utilization of edge computing in vehicular networks for enhancing communication efficiency and addressing challenges. Section 3 presents the detailed explanation of the proposed fuzzy-based system and testbed design. In Section 4, the article highlights simulation results and testbed findings, providing a detailed comparison between the two to validate the effectiveness of the proposed system.

Finally, in Section 5, we draw insightful conclusions based on their study's findings and outline potential future work to further improve and advance the integration of these technologies in SDN-VANETs for even greater impact in smart transportation systems.

2. Background Overview

IoT and VANETs represent transformative paradigms that are reshaping the way we interact with the world around us. IoT involves connecting everyday objects and devices to the internet, enabling data exchange, automation, and intelligent decision making across various domains. VANETs, on the other hand, focus on creating an interconnected ecosystem where vehicles, infrastructure, and pedestrians communicate to enhance transportation efficiency, safety, and sustainability. Both IoT and VANETs heavily rely on other technologies to realize their full potential, such as network virtualization with SDN and cloud, fog, and edge computing.

2.1. Internet of Things

IoT refers to the vast network of interconnected devices and the technology facilitating communication between these devices and the cloud, as well as among the devices themselves. This interconnectedness has been made possible by the proliferation of affordable computer chips and high-bandwidth telecommunication, resulting in billions of devices now connected to the internet. Everyday objects are now equipped with sensors that collect data and enable intelligent responses to user interactions.

Numerous examples of IoT applications are in use today. Connected cars leverage internet connectivity for various purposes, such as monitoring driver performance, vehicle health, and optimizing fuel efficiency in rental car fleets. This is due to IoT technology, which has enabled cars to be equipped with an array of sensors, processors, and communication modules, transforming them into intelligent and interconnected devices.

Additionally, IoT applications have extended to smart cities, where governments use IoT technology for urban planning, infrastructure maintenance, and environmental monitoring, leading to more efficient and sustainable city management. The Internet of Things has transformed the way we interact with the world around us, enhancing various aspects of daily life, business operations, and urban development through the seamless integration of smart devices and the vast connectivity they offer.

2.2. VANETs

VANETs are a transformative concept that leverage the power of connectivity and data exchange to create an interconnected ecosystem of vehicles, infrastructure, and pedestrians. In VANETs, vehicles are equipped with advanced sensors, communication technologies, and onboard computing systems, enabling them to collect and share real-time data with each other and the surrounding environment. This seamless flow of information facilitates intelligent decision making, optimizing traffic flow, enhancing road safety, and improving overall transportation efficiency. Connected vehicles are at the heart of the VANETs, equipped with embedded communication modules that enable Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication [8]. Through V2V and V2I communication, connected vehicles can exchange critical information, such as speed, location, and road conditions, enabling features such as collision avoidance, adaptive cruise control, and real-time traffic management [9]. Additionally, connected vehicles can benefit from cloud-based services, leveraging the IoT to access data, perform complex analytics, and enable various personalized in-car services [10]. Connected vehicles offer numerous advantages to the IoT ecosystem, contributing to safer, more efficient, and intelligent transportation. Some key advantages of connected vehicles to IoT are as follows.

Enhanced Safety: Connected vehicles can communicate with each other and with roadside infrastructure, exchanging real-time data on traffic conditions, road hazards, and potential collisions. This communication enables advanced driver assistance systems

(ADAS) and cooperative collision avoidance, significantly reducing the risk of accidents and enhancing overall road safety.

Improved Traffic Management: By sharing data with smart traffic management systems, connected vehicles contribute to more efficient traffic flow. These vehicles can receive real-time traffic updates and suggested alternative routes, reducing congestion and alleviating bottlenecks on the road.

Remote Diagnostics and Maintenance: IoT-enabled sensors in connected vehicles can monitor various vehicle parameters, such as engine health, tire pressure, and battery status. These data can be transmitted to manufacturers and service centers for remote diagnostics and predictive maintenance, enabling timely repairs and reducing downtime.

Enhanced Fuel Efficiency: Connected vehicles can receive information about traffic patterns, road conditions, and fuel prices, allowing drivers to optimize their routes and driving behavior for better fuel efficiency. This leads to cost savings for vehicle owners and reduced greenhouse gas emissions.

Personalized Services: IoT connectivity enables personalized services for drivers and passengers. Connected vehicles can provide in-car infotainment, personalized navigation, and even access to smart home controls, making the driving experience more enjoyable and convenient.

Autonomous Driving: Connected vehicles play a vital role in the development of autonomous driving technology. The exchange of real-time data between vehicles and the surrounding environment is crucial for autonomous vehicles to make informed decisions and navigate safely without human intervention.

Fleet Management and Optimization: IoT-enabled connected vehicles are valuable for fleet operators as they can monitor the location, performance, and fuel efficiency of each vehicle in the fleet. These data allow for better fleet management, route optimization, and resource allocation, leading to cost reductions and improved operational efficiency.

Integration with Smart Cities: Connected vehicles can seamlessly integrate with smart city initiatives. They can provide data on traffic patterns, parking availability, and air quality, supporting urban planners in making data-driven decisions to create more sustainable and livable cities.

Real-time Updates and Software Upgrades: With IoT connectivity, vehicle manufacturers can remotely deliver software updates and new features to connected vehicles. This ensures that vehicles stay up to date with the latest technologies and improvements, enhancing their performance and safety over time.

2.3. Cloud, Fog, and Edge Computing integrated in VANETs

At the core of this concept is cloud computing, which provides a centralized repository for storing vast amounts of vehicle-related data, including telemetry, sensor readings, mapping information, and historical performance data [11–14]. Cloud-based services enable connected vehicles to access a wide array of data and applications from remote servers, facilitating services such as over-the-air software updates, navigation, multimedia streaming, and personalized user settings. Cloud resources also enable the integration of Artificial Intelligence (AI) and Machine Learning (ML) algorithms for advanced analytics, predictive maintenance, and behavior analysis, enabling vehicles to become more autonomous and efficient.

Fog computing is used to bridge the gap between cloud-based services and the vehicle itself. It involves the deployment of computing and storage resources at the network's edge, closer to the vehicles. This approach reduces latency and bandwidth consumption, enabling real-time data processing and decision making at the edge of the network [15–17]. Fog nodes placed at roadside infrastructure, such as traffic lights and smart intersections, can provide localized data processing and traffic management, ensuring quicker response times and improved safety for connected vehicles in urban environments.

Edge computing further enhances the real-time capabilities of connected vehicles by enabling processing and data analysis directly within the vehicles themselves. Edge nodes embedded within the vehicles process data from onboard sensors, cameras, and other connected devices, allowing instant response to critical situations and reducing dependency on cloud connectivity. Edge-based SDN enables dynamic reconfiguration of network policies within the vehicle, ensuring efficient utilization of resources and bandwidth allocation for diverse applications.

Figure 1 illustrates the logical architecture of cloud-fog-edge SDN-VANETs featuring content distribution.

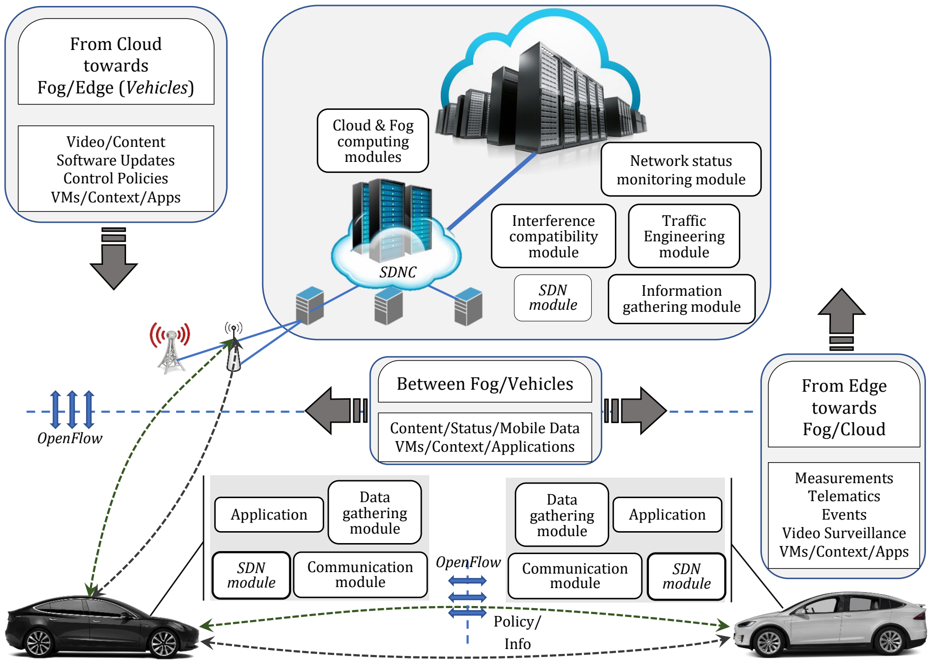

Figure 1. The logical framework of cloud-fog-edge SDN-VANET with content distribution.

2.4. SDN-VANETs Paradigm

In traditional networks, the network nodes (routers/switches) are equipped with the control plane and forwarding plane and have applications loaded onto it. The data packets are directed from each network node individually, based on their local logic, and application changes must be programmed systematically into each device individually, whereas in SDN, the applications and intelligence do not reside in network nodes but in the SDN controller, which makes the network programmable. SDN consists of the decoupling of control plane and forward plane, which pulls out the network intelligence from the individual nodes of the network and places it in the hands of a central authority, the SDN controller. The network acts like one big router controlling the network. SDN provides programmability and a scalable solution for network growth, as adding new devices or expanding the network can be done with ease through software-based configurations. This scalability is particularly valuable in cloud computing environments, IoT, and large-scale data centers. Moreover, SDN's intelligent traffic management capabilities optimize network traffic flow, leading to reduced congestion and better utilization of network resources.

The integration of SDN technology in cloud-fog-edge VANETs facilitates intelligent network management. SDN allows for centralized control and orchestration of network resources, enabling seamless handovers between different connectivity options, such as cellular networks, Wi-Fi, and V2X (Vehicle-to-Everything) communications [18,19]. This dynamic allocation optimizes resource utilization and ensures efficient data processing and delivery. In the cloud-fog-edge VANETs environment, resources are spread across different layers, such as cloud data centers, fog nodes at the network edge, and on-board edge devices within vehicles. SDN orchestration ensures that resources are allocated dynamically based on the specific requirements of each application, service, or connected vehicle [20–24]. SDN-based controllers can optimize network paths and allocate bandwidth based on real-time data demands, prioritizing critical applications such as safety-related messages and emergency alerts.

2.5. Related Works

Recent advancements in VANETs have spurred research on innovative approaches to process data within the car and share the available computing resources with other nearby cars in need for free resources. These approaches leverage the distributed nature of VANETs to enhance data processing efficiency, reduce latency, and improve the overall performance of various applications.

Researchers have explored the integration of edge computing into vehicular networks to cater to QoS needs across a spectrum of applications with varying QoS constraints. Wu et al. in [25] approached the matter by investigating the uplink local delay between a vehicle and an edge node through a theoretical methodology grounded in stochastic geometry. Their analysis modeled the distribution of vehicles as an independent one-dimensional homogeneous Poisson point process, dissecting the primary contributors to transmission delay. Furthermore, Zhang et al. [26] adopted a network slicing strategy to establish distinct QoS requisites, thereby facilitating comprehensive QoS support for the manifold demands of vehicular networks. VANETs have a diverse range of applications, including the provision of reliable communication services in disaster-stricken areas. In [27], the authors provided a communications infrastructure in post-disaster scenarios via vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) collaboration framework. This involves integrating non-orthogonal multiple access into Air-Base-Station-supported Internet of Vehicles (IoV), enabling four spectrum reuse modes through successive interference cancellation (SIC). A total power consumption minimization problem is formulated, considering mode selection, power control, and channel state information latency while meeting reliability and rate requirements. Simulation results indicate that the proposed approach reduces power consumption significantly compared to existing methods. In [28], the authors introduced an enhanced vehicle rerouting strategy for VANETs aimed at alleviating urban traffic congestion during peak hours. Unlike prior methods, which focused on hop count, this strategy incorporates dynamic traffic information such as travel time for selecting vehicles to be rerouted. Additionally, the approach ensures collaborative rerouting by updating road capacities to prevent overutilization and future congestion. Through traffic simulations in various network scenarios, including real-world settings, the proposed strategy showcases superior performance compared to existing strategies, achieving a minimum 4.39% improvement in average travel time in the Kuala Lumpur network.

Other research works have explored the benefits of processing data at the edge of the network, such as inside vehicles or in nearby roadside units. A study by Zhang et al. [29] proposed a dynamic edge computing framework for traffic management in urban VANETs. The authors demonstrated that by processing traffic data at the edge, traffic signal coordination and congestion control could be significantly improved. Another paper [30] proposed a novel approach to jointly orchestrate networking and computing resources based on user requirements. The approach uses Intent-Based Networking (IBN) based on Software-Defined Networking and provides the ability to automatically handle and manage the networking requirements. The method takes into consideration the CPU of vehicles, memory capacities, and location constraints which decide where the service can be executed, and application requirements such as the bandwidth and latency for the service to function correctly. The proposed method shows satisfying results; response of the system is 95% faster, resource utilization is up to 76% and 71% higher acceptance ratio of computing and networking requests with various priorities. Another work that exploits the shared unoccupied on-board computing resources of smart vehicles is presented in [31]. The paper explored the concept of collaborative task offloading within decentralized vehicular networks and introduced an innovative collaborative edge computing solution. The main emphasis of the study was on effectively harnessing the computational resources available in other vehicles, while taking into account both task computational time and communication delays associated with data transmission. The task offloading framework adopts a dual-phase decision-making strategy. In the initial phase, the selection of edge nodes able to process tasks is carried out by fuzzy logic based on the vehicle computational capacity, vehicle mobility, and vehicle distribution in a collective manner. The output is an efficient network edge topology for task offloading. The subsequent phase handles task offloading to multi-hop neighbors. The edge node optimizes the request for each vehicle to find the best candidates for performing task processors for the tasks. The decision is taken by fuzzy logic system based on the offloading restrictions that encompass computational time and network resource limitations. The node calculates a capability value for each candidate, and the candidate with the highest capability value is chosen as the task processor. In cases where an edge node is unable to identify a suitable vehicle for task offloading, it reaches out to a neighboring edge node for assistance. Simulation results demonstrate that through enhanced collaboration among vehicles, the proposed scheme achieves a better task completion ratio and reduced task response time compared to other existing baseline methods.

3. Proposed System

Edge computing in connected vehicles involves processing and analyzing data directly within the vehicles themselves, bringing computational capabilities closer to the data source rather than relying solely on centralized cloud infrastructure. This paradigm shift offers several significant benefits that enhance the overall performance, efficiency, and intelligence of connected vehicles. At the core of edge computing in connected vehicles is the deployment of onboard computing resources, such as powerful processors, microcontrollers, and specialized hardware accelerators. Vehicles exploit their own processing, storage, and network capability to process data from various sensors and connected devices within the vehicle, including cameras, lidar, radar, GPS, and in-car sensors [32]. By analyzing data locally, edge computing enables real-time decision making, rapid response to changing road conditions, and efficient utilization of the available resources.

One of the primary advantages of edge computing in connected vehicles is the reduction in data latency. Time-sensitive applications, such as advanced driver assistance systems (ADAS), collision avoidance, and emergency braking, demand immediate responses to ensure safety and prevent accidents. With edge computing, data analysis occurs within milliseconds, reducing the time between data collection and actionable insights, making connected vehicles more responsive and safer on the road. Moreover, edge computing decreases the dependency on continuous internet connectivity for data exchange. Connected vehicles can operate effectively even in areas with limited or intermittent network coverage, since crucial data processing occurs locally. This ensures the uninterrupted functionality of essential services within the vehicle, such as navigation, entertainment, and safety features, irrespective of external network conditions.

Edge computing also optimizes the use of network bandwidth and cloud resources. By processing and filtering data locally, only relevant and essential information is sent to the cloud, reducing data transmission and avoiding bottleneck occupation and cloud storage costs. This bandwidth optimization is particularly crucial in scenarios with a large fleet of connected vehicles, where data traffic can be extensive, and cloud resources can be better utilized for high-level analytics and long-term data storage. Security and privacy are strengthened by edge computing in connected vehicles. Critical data can be processed within the vehicle's secure environment, reducing the risk of unauthorized access or data breaches during transit to external servers. Edge computing also fosters autonomy and decentralization in connected vehicles. Vehicles become capable of making intelligent decisions independently, without relying solely on cloud services. This increased autonomy is especially valuable in scenarios where real-time operations are essential, and cloud connectivity may be temporarily unavailable.

Moreover, computing process and data analysis through adjacent vehicles in connected vehicles involves leveraging the collective computing power and data resources of nearby vehicles to enhance the overall intelligence and efficiency of the connected vehicle ecosystem. This cooperative approach allows vehicles to share and exchange data, process information, and collaborate with one another, creating a dynamic network of vehicles that can make more informed decisions and respond better to changing road conditions. In this scenario, each connected vehicle acts as a node in a distributed network, capable of communicating with other nearby vehicles within its communication range through V2V communication links. Not only can they share the collected data with adjacent vehicles, but they can also share their processing, storage, and network resources with nearby vehicles that are in need of more additional resources (hereinafter referred to as *the vehicle*). Therefore, when *a vehicle* is in need for additional storage and computing resources, *the vehicle* can request to utilize those of neighboring vehicles, assuming they can establish and maintain a connection for a certain period. The proposed method, named FSA-ELC, evaluates the processing capabilities of each nearby vehicles within the communication range of *the vehicle*. FSA-ELC assesses the edge layer's processing and storage capacity, which comprises the total number of vehicles capable of communicating and sharing their storage and processing abilities with each other.

In the following, we provide a detailed composition of the FSA-ELC system, explain the input and output parameters, and present the design and implementation of the FSA-ELC testbed. The input parameters of the FSA-ELC system are not interrelated, leading to an NP-hard problem. FL is employed to address these issues since it excels at handling such complex problems [33–38]. Additionally, our system requires real-time decision-making capabilities, and fuzzy systems have proven to deliver excellent results in decision making and control problems.

3.1. FSA-ELC System

The proposed approach for assessing the computing capability of the edge layer in the SDN-VANETs environment, named FSA-ELC, is built based on a FL system. This FL system employs input parameters that are uncorrelated with each other, ensuring a comprehensive and independent evaluation of the edge computing capability. Based on FSA-ELC output parameter result, the system can take action corresponding the circumstances. The input and output parameter, their corresponding term sets and the fuzzy rule base. The structure of the FSA-ELC system is given in Figure 2.

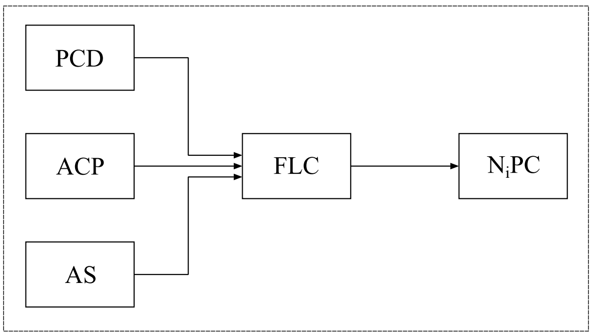

Figure 2. Structure of FSA-ELC system.

In the following, we provide a detailed explanation of the input and output parameters of the proposed system.

Available Storage (AS): In VANETs, vehicles are equipped with storage capability, which allows vehicles to store and retain data, information, and various types of content either for onboard use or for later retrieval and analysis. However, the amount of onboard storage can vary significantly between vehicle models and types. Some vehicles may have limited storage capacity. The available storage refers to the amount of storage space in a vehicle that is currently unused and available for storing data, applications, files, and other content. It represents the remaining capacity on the device's internal storage or external storage media. In VANETs, the vehicles can exchange data and access each other's available storage resources. For example, a vehicle might share map data, navigation updates, or media files with another vehicle, or retrieve specific data it requires for its own functionality. To do so, vehicles need to initiate virtual machines connections via V2V communications.

Available Computing Power (ACP): Computing resources in connected vehicles refer to the onboard processing capabilities, such as CPUs (Central Processing Units), GPUs (Graphics Processing Units), and specialized processors, which enable the vehicles to execute various computational tasks. These computing resources are essential for handling data processing, real-time decision making, running applications, and supporting advanced features in connected vehicles. In connected vehicles, the vehicles can share their available computing resources with *the vehicle* in need of additional computing resources via V2V communication. Distributing computing augments connected vehicles' processing capabilities, enable faster data analysis, and enhance overall performance. This collaborative approach promotes more efficient use of computing power, reduces individual vehicle workload, and fosters a cooperative ecosystem that benefits all vehicles in the network.

Predicted Contact Duration (PCD): An estimation of the duration or time interval during which two or more vehicles can maintain a stable and reliable communication link with each other. In the context of V2V communication, predicting contact duration becomes crucial to optimize data exchange, coordinate collaborative tasks, and enable efficient communication between nearby vehicles. Making an estimation about V2V communication time and the necessary time for terminating a certain task, vehicles can decide if this communication can be accomplished successfully; thus, optimizing network resources and reducing unnecessary communication attempts. Accurate prediction of contact duration supports advanced safety applications, and enables the exchange of real-time information critical for autonomous driving and traffic management.

To calculate the PCD between *the vehicle* and a neighbor vehicle i, we first calculate the relative speed between these two vehicles using the law of cosines, as given in Equation (1):

$$RSV_i = \sqrt{V^2 + V_i^2 - 2VV_i \cos\theta_i} \tag{1}$$

where V is the speed of *the vehicle*, V_i is the speed of neighbor i, and θ_i is the angle between their directions. Then, we use the law of cosines once again to calculate the PCD, as given in Equation (2):

$$(RSV_i \cdot PCD)^2 + D_0^2 - 2|RSV_i| \cdot PCD \cdot D_0 \cos(\gamma_i + \beta_i) = CR^2 \tag{2}$$

where D_0 denotes the initial distance between the two vehicles, CR is the communication range, γ_i is the angle between the direction of *the vehicle*, and D_0 is the imaginary line, whereas β_i is calculated with Equation (3), which is derived from the law of sines:

$$\beta_i = \begin{cases} \arcsin(\frac{V_i \sin\theta_i}{|RSV_i|}), & \text{for } V_i \leq \sqrt{V^2 + RSV_i^2} \\ 180 - \arcsin(\frac{V_i \sin\theta_i}{|RSV_i|}), & \text{for } V_i > \sqrt{V^2 + RSV_i^2}, \theta \geq 0 \\ -180 - \arcsin(\frac{V_i \sin\theta_i}{|RSV_i|}), & \text{for } V_i > \sqrt{V^2 + RSV_i^2}, \theta < 0 \end{cases} \tag{3}$$

We posit that when two vehicles are getting farther from each other from different directions, their directions form a positive angle, whereas when the vehicles are getting closer, θ is negative.

Neighbor i Processing Capability (NiPC): The output parameter of FSA-ELC, named NiPC, represents the ability of a potential neighbor to share its available resources with other vehicles in the connected vehicles environment. The scale of NiPC results ranges from 0 to 1. A Neighbor i with a processing capability score of 0 indicates that the vehicle is not willing or able to share its available resources with other vehicles. This vehicle may be heavily loaded with computational tasks, have limited processing power, or might not be able to engage in cooperative resource sharing for the necessary time. Conversely, NiPC score of 1 signifies that the vehicle is highly willing and capable of sharing its available resources with other vehicles. The Neighbor *i* Processing Capability is a dynamic parameter that can change as vehicles move and their relative positions shift. As vehicles navigate through the connected environment, they continually assess the processing capabilities of neighboring vehicles and adapt their cooperative computing strategies accordingly. Vehicles can use this scale to make dynamic decisions on which neighboring vehicles to collaborate with, optimize resource utilization, and achieve efficient cooperative data processing and decision making.

Table 1 displays the term sets for the input and output parameters of FSA-ELC. The input parameters are fuzzified using the membership functions illustrated in Figure 3. The determination of the number of terms for each parameter and the characteristics of the membership functions is based on experience gained from numerous simulations. It has been observed that using less than three linguistic terms for an input parameter may result in inefficient control and poor decision making, while employing more terms leads to redundancies and increased complexity. Similar considerations apply to the overlap of membership functions, as less overlap can yield suboptimal decisions, while more overlap may introduce redundancies. For real-time operation, we employ triangular and trapezoidal membership functions, which have been found to be the most suitable. Additionally, Table 2 present the Fuzzy Rule Base (FRB) of FSA-ELC, consisting of IF-THEN rules.

Table 1. Parameters and term sets for FSA-ELC.

Parameters	Term Sets
ACP	Small (Sm), Medium (Me), Large (La)
PCD	Short (Sh), Medium (Md), Long (Lo)
AS	Small (S), Medium (M), Big (B)
NiPC	Extremely Low (El), Very Low (Vl), Low (Lw), Moderate (Md), High (Hg), Very High (Vh), Extremely High (Eh)

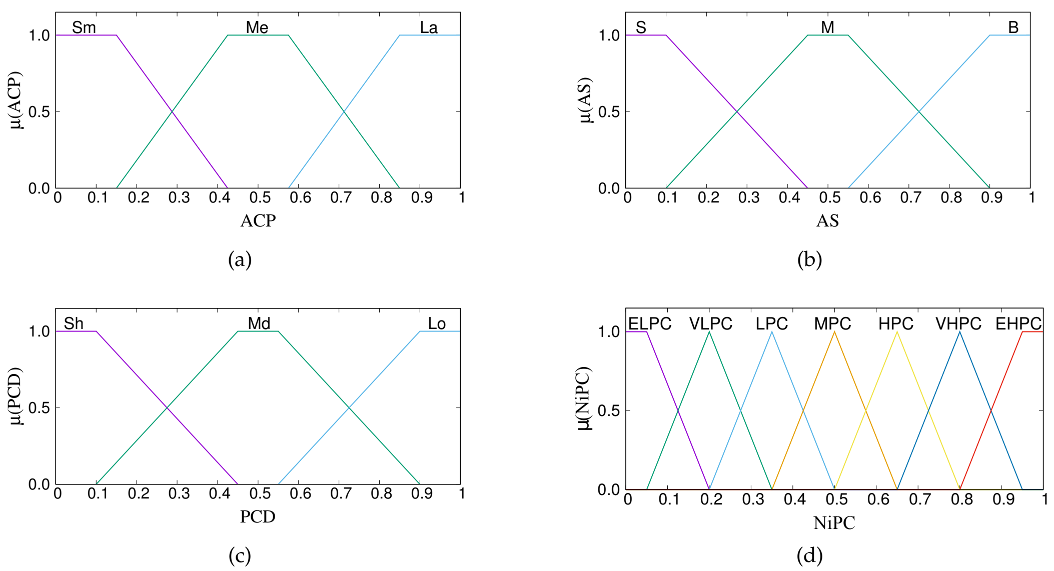

Figure 3. Membership functions of FSA-ELC. (**a**) Available Computing Power, (**b**) Available Storage, (**c**) Predicted Contact Duration, and (**d**) Neighbor *i* Processing Capability.

Table 2. FRB of FSA-ELC.

No	ACP	PCD	AS	NiPC	No	ACP	PCD	AS	NiPC	No	ACP	PCD	AS	NiPC
1	Sm	Sh	S	ELPC	10	Me	Sh	S	VLPC	19	La	Sh	S	LPC
2	Sm	Sh	M	ELPC	11	Me	Sh	M	LPC	20	La	Sh	M	HPC
3	Sm	Sh	B	ELPC	12	Me	Sh	B	LPC	21	La	Sh	B	HPC
4	Sm	Md	S	VLPC	13	Me	Md	S	LPC	22	La	Md	S	MPC
5	Sm	Md	M	VLPC	14	Me	Md	M	MPC	23	La	Md	M	VHPC
6	Sm	Md	B	LPC	15	Me	Md	B	MPC	24	La	Md	B	VHPC
7	Sm	Lo	S	LPC	16	Me	Lo	S	MPC	25	La	Lo	S	VHPC
8	Sm	Lo	M	LPC	17	Me	Lo	M	HPC	26	La	Lo	M	EHPC
9	Sm	Lo	B	MPC	18	Me	Lo	B	HPC	27	La	Lo	B	EHPC

3.2. FSA-ELC Testbed Design

To assess the simulation results of the FSA-ELC system, we conducted corresponding experiments using a small-scale testbed implemented with Raspberry Pis (RPi). The testbed comprises five RPis, representing vehicles moving through an urban area for approximately 25 min, encompassing several apartment blocks. Among these vehicles, one acts as the resource-needy *vehicle*, while the remaining four serve as potential neighbors capable of providing assistance if they possess sufficient processing capability. In that case, the system will decide to deploy the application at the edge layer; thus, bringing computational capabilities closer to the data source rather than relying solely on centralized cloud infrastructure. The testbed area measures 200 m $\times$ 200 m, and each vehicle has a communication range of 50 m. The vehicle movements are simulated using the sumo simulator, and the layout is illustrated in Figure 4. The components of FSA-ELC testbed are given in Figure 5, whereas the setup of the testbed is summarized in Table 3.

The mobility trace is used to obtain the locations of all the vehicles at each time step of the experiment since the deployment of a large-scale testbed in a real environment (e.g., the RPi moving around in the neighborhood) was impossible due to various factors (i.e., increased costs, lack of human resources, and time constraints).

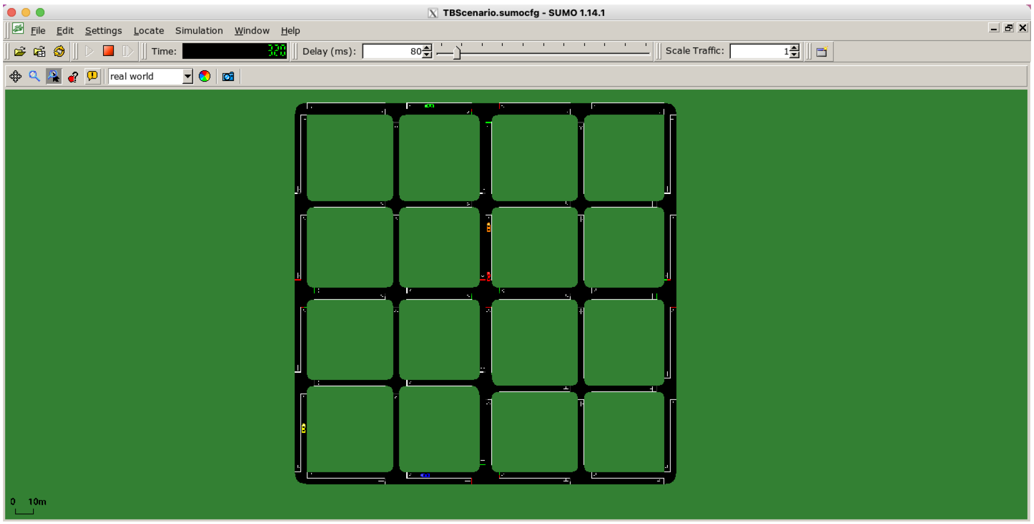

Figure 4. A screenshot of vehicles moving around the area in the sumo simulator.

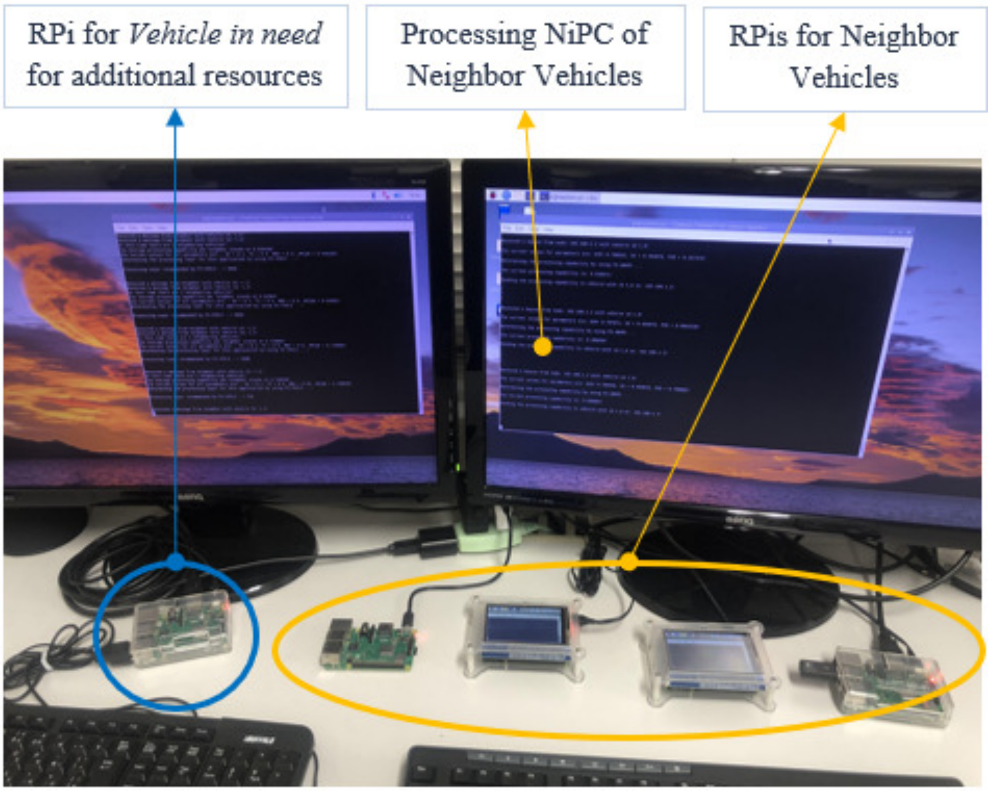

Figure 5. The components of the FSA-ELC testbed.

Table 3. Testbed setup.

Vehicles	5 RPi Model 3B+
Mobility trace generator	sumo
Area size	200 m × 200 m
Communication range	50 m
TS	[0.1, 0.5, 0.9]
DC	[0.1, 0.5, 0.9]
Experimental time	1500 s

The vehicle and its neighbors in the testbed communicate with one another, as given in Figure 6. *The vehicle* broadcasts every second a help beacon containing information about the vehicle id, speed, direction, timestep, and current and previous location. The vehicles within the communication range, also known as neighbors, receive the beacon and extract the information it contains so they can calculate the relative speed and the predicted contact duration. A RPi is considered a neighbor vehicle only if the distance, which is calculated using the coordinates obtained from the mobility trace, is shorter than their communication range. After calculating the predicted contact duration, each neighbor calculates its current available CPU and storage, which are the data they need to determine their processing capability. The processing capability is determined by running FSA-ELC and the result is sent back to the vehicle alongside its id in the form of a response message. Based on the result value, the system can then take action corresponding to the circumstances.

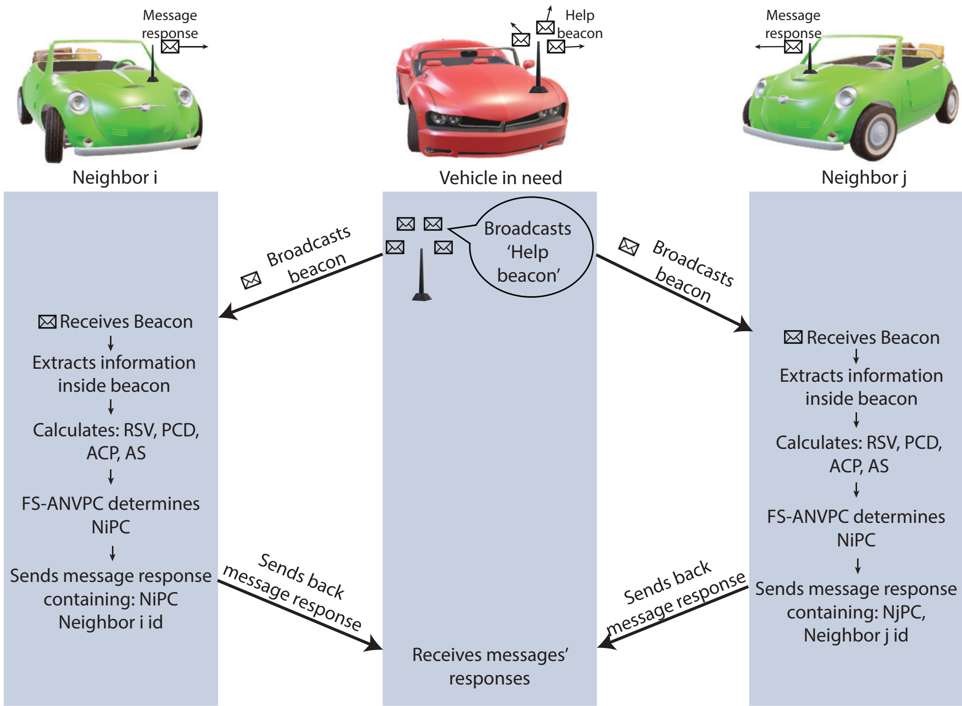

Figure 6. A scheme of the communication between *the vehicle* and its neighbors.

4. Evaluation Results

In this section, we discuss the simulation and the experimental results for FSA-ELC system. The experiments were conducted by the communication of five RPis moving randomly, with one of them representing *the vehicle* in need of resources.

4.1. Results of FSA-ELC

Three scenarios are considered for FSA-ELC, each with different available storage (AS) values: small, medium, and large. The results of these scenarios are depicted in Figure 7. In Figure 7a, when the available storage is small, no vehicles are considered as prospective helpers if the predicted contact duration is short, even if they have large ACP. However, we can see that when PCD increases, the neighbors can be considered as helpful depending on the available ACP. Figure 7b demonstrates that a neighbor with a medium AS can offer help in more scenarios, given that they have more than the minimum amount of computing resources available for other vehicles. If these neighbors have a medium amount of ACP, they should be in contact with the vehicle for a considerable amount of time to be considered as helpful. Similarly, in Figure 7c, when a neighbor is willing to provide a large amount of computing power, even vehicles with a short PCD are considered helpful. Different from the case with medium AS, here, we see that if PCD is long, the vehicles with small ACP are also included. In such cases, the neighbors can be used for processing applications that do not require large computing power but need vast storage sizes. These findings show the potential collaboration and resource-sharing dynamics among vehicles in the FSA-ELC framework.

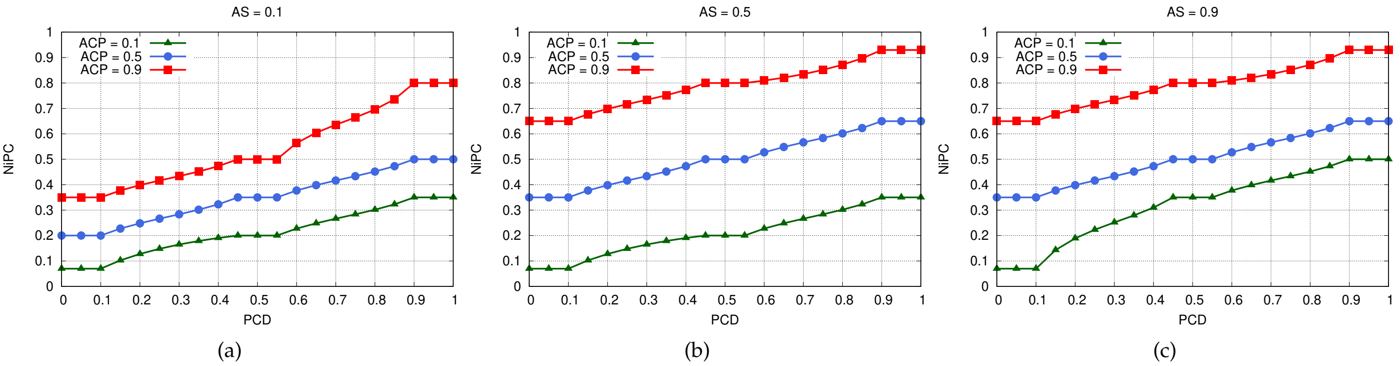

Figure 7. Simulation results for FSA-ELC. (**a**) AS = 0.1, (**b**) AS = 0.5, and (**c**) AS = 0.9.

4.2. Experimental Results of FSA-ELC

The experimental results of FSA-ELC are given in Figures 8 and 9. Let us compare them with the simulation results of FSA-ELC, given in Figure 7. The comparison between Figure 8 and the simulations reveals that both sets of results follow the same trend. However, in the testbed results, more oscillations can be observed, especially for small and medium ACP values. This variation is attributed to the fluctuations in the ACP of the Raspberry Pi (RPi) when running multiple applications simultaneously, emulating a scenario similar to a vehicle running its own applications alongside those requiring additional resources. For small AS (Figure 8a), vehicles with large ACP and long PCD are considered helpful. On the other hand, for medium and big AS (Figure 8b,c), vehicles with large ACP are deemed helpful, regardless of the PCD value. The range of output values (NiPC) varies from [0.069–0.680] for AS = 0.1 and slightly increases to [0.069–0.787] for AS = 0.5 and for AS = 0.9. These findings demonstrate the performance of FSA-ELC in both experimental and simulated environments, exhibiting its effectiveness in assessing computing capability in SDN-VANETs environments with various resource availability scenarios.

In Figure 9, we show the relation between the decisions FSA-ELC in respect to time. As we can see, the minimum (NiPC = 0.069) and maximum (NiPC = 0.787) values are the same as those in Figure 8 for the respective scenario. On the other hand, here, the variations are bigger due to the impact of both ACP and PCD, differently from the previous figure where only ACP was influencing the output value. Despite the variations, we can see that for the most part of the experimental time, the variation remains either over or below 0.5, especially in Figure 9b,c, indicating a stability in the decision of the system to determine potential neighbors. For medium and especially for big AS, the results indicate that in most of the cases the vehicles are determined as helpful (NiPC values are over 0.5), with only a few situations when NiPC is below 0.5. The same holds true for results in Figures 7 and 8.

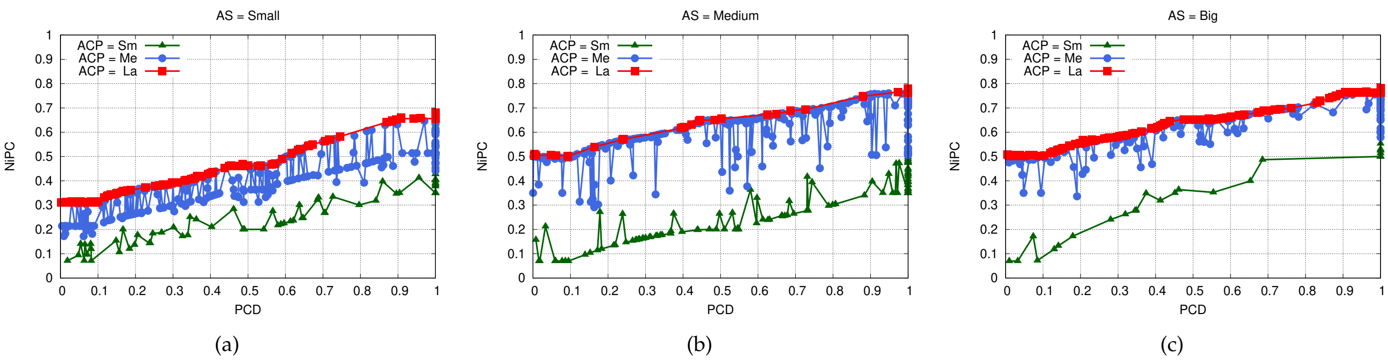

Figure 8. Testbed results for FSA-ELC. (**a**) AS = Small, (**b**) AS = Medium, and (**c**) AS = Big.

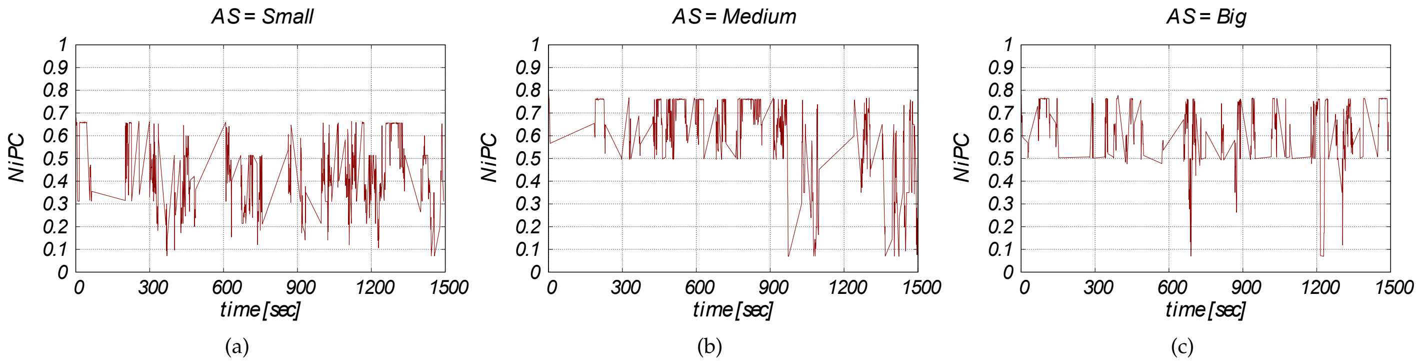

Figure 9. Experimental results for FSA-ELC in terms of time. (**a**) AS = Small, (**b**) AS = Medium, and (**c**) AS = Big.

5. Conclusions

This article introduces a fuzzy-based system and a testbed implementation aimed at assessing the available edge computing resources within a layered cloud-fog-edge architecture for SDN-VANETs. The proposed FSA-ELC system evaluates neighboring vehicles' capacity to assist those lacking sufficient resources for specific tasks, taking into account ACP, AS, and PCD values. Both simulation and experimental results demonstrate the influence of these parameters on the system's performance and feasibility. The findings from the simulations and experiments lead to the following conclusions:

- Neighboring vehicles with small ACP and small AS are unable to offer assistance to other vehicles in need.
- For medium and large AS values, vehicles with substantial ACP are considered helpful, irrespective of the PCD value.
- The highest NiPC value is attained when the neighboring vehicle has significant ACP, large AS, and an extended PCD.
- Notably, the simulation and experimental results exhibit a consistent trend, validating the system's effectiveness and reliability.

Despite the positive results, to take full advantage of the proposed system, it is noteworthy to determine the accuracy of the system as it can show which parameters lead mostly to false positives and what should be improved to reduce the false-negative outputs. In the following, we show some future aspects regarding improvements that we aim to make in the proposed system.

- In our testbed, we used the sumo simulator to generate the movement of vehicles. In the future, we aim to improve the testbed by implementing mobile RPis moving randomly in designated roads, equipped with GPS which enable real-time localization of vehicles (RPis).
- Second, in our testbed, we use only five RPis which represent the moving vehicles. We get an understanding of how the proposed system and the testbed performs, but we would like to implement a large size network with many more vehicles, making it more like a real-life scenario. A large size network will show how the proposed system responds to data congestion, interference, and many other problems that might arise which could have a profound effect on the network performance.
- Third, during the vehicle communication sessions in the experiment, no application is running, and only small data size packets are exchanged. In the future, we intend to evaluate the performance by running a real application in the vehicle. In this way, we can prove whether the application in the end is performed successfully via the decision taken by the proposed system. For example, the fuzzy-based system decides that edge layer is capable to execute the application, when in fact the application might not be successfully accomplished in the edge layer. Such false positives/negatives

determine the accuracy of the system; therefore, it is important to investigate which parameters lead to these false results, in order to improve the system.
- Lastly, we mean to compare our system with existing systems in terms of the achieved accuracy and applications that can be covered.

Author Contributions: Conceptualization, E.Q. and K.B.; methodology, E.Q, K.B. and L.B.; software, K.B., E.Q., P.A. and K.M.; validation, S.H., K.M. and L.B.; writing—original draft preparation, E.Q. and K.B.; visualization, E.Q.; supervision, L.B. and K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Christidis, P.; Rivas, N.I. *Measuring Road Congestion*; Scientific Analysis or Review LF-NA-25550-EN-N; Publications Office of the European Union: Luxembourg, 2012. [CrossRef]
2. Mizutani, F.; Suzuki, Y.; Sakai, H. Estimation of Social Costs of Transport in Japan. *Urban Stud.* **2011**, *48*, 3537–3559. [CrossRef]
3. Schrank, B.; Albert, L.; Eisele, B.A. *Urban Mobility Report*; The Texas A&M Transportation Institute with Cooperation from INRIX: Bryan, TX, USA, 2021.
4. World Health Organization. *Global Status Report on Road Safety 2018: Summary*; World Health Organization: Geneva, Switzerland, 2018. (WHO/NMH/NVI/18.20). Licence: CC BY-NC-SA 3.0 IGO).
5. Jat, S.; Tomar, R.S.; Satya Prakash Sharma, M. Traffic Analysis for Accidents Reduction in VANET's. In Proceedings of the 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 11–12 December 2019; pp. 115–118. [CrossRef]
6. Bany Taha, M.; Alrabaee, S.; Choo, K.K.R. Efficient Resource Management of Micro-Services in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 6820–6835. [CrossRef]
7. Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big Data Analytics in Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 383–398. [CrossRef]
8. Lee, M.; Atkison, T. VANET applications: Past, present, and future. *Veh. Commun.* **2021**, *28*, 100310. [CrossRef]
9. Lottermann, C.; Botsov, M.; Fertl, P.; Müllner, R.; Araniti, G.; Campolo, C.; Condoluci, M.; Iera, A.; Molinaro, A. LTE for Vehicular Communications. In *Vehicular ad Hoc Networks: Standards, Solutions, and Research*; Springer International Publishing: Cham, Switzerland, 2015; pp. 457–501.
10. Hartenstein, H.; Laberteaux, K. Introduction. In *VANET Vehicular Applications and Inter-Networking Technologies*; John Wiley & Sons: Hoboken, NJ, USA, 2010; pp. 1–19. [CrossRef]
11. Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Kong, J.; Jue, J.P. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [CrossRef]
12. Paranjothi, A.; Tanik, U.; Wang, Y.; Khan, M.S. Hybrid-Vehfog: A robust approach for reliable dissemination of critical messages in connected vehicles. *Trans. Emerg. Telecommun. Technol.* **2019**, *30*, e3595. [CrossRef]
13. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog Computing and Its Role in the Internet of Things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, New York, NY, USA, 17 August 2012; MCC '12, pp. 13–16. [CrossRef]
14. Taherizadeh, S.; Stankovski, V.; Grobelnik, M. A Capillary Computing Architecture for Dynamic Internet of Things: Orchestration of Microservices from Edge Devices to Fog and Cloud Providers. *Sensors* **2018**, *18*, 2938. [CrossRef]
15. Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog Computing: A Platform for Internet of Things and Analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*; Springer International Publishing: Cham, Switzerland, 2014; pp. 169–186. [CrossRef]
16. David Linthicum. Edge Computing vs. Fog Computing: Definitions and Enterprise Uses. Cisco. Available online: https://www.cisco.com/c/en/us/solutions/enterprise-networks/edge-computing.html (accessed on 30 July 2019).
17. Samara, G.; Rasmi, M.; Sweerky, N.A.; Daoud, E.A.; Salem, A.A. Improving VANET's Performance by Incorporated Fog-Cloud Layer (FCL). In Proceedings of the 2021 22nd International Arab Conference on Information Technology (ACIT), Muscat, Oman, 21–23 December 2021; pp. 1–5. [CrossRef]
18. Nkenyereye, L.; Nkenyereye, L.; Islam, S.M.R.; Choi, Y.H.; Bilal, M.; Jang, J.W. Software-Defined Network-Based Vehicular Networks: A Position Paper on Their Modeling and Implementation. *Sensors* **2019**, *19*, 3788. [CrossRef] [PubMed]
19. Yang, F.; Wang, S.; Li, J.; Liu, Z.; Sun, Q. An overview of Internet of Vehicles. *China Commun.* **2014**, *11*, 1–15. [CrossRef]
20. Al-Heety, O.S.; Zakaria, Z.; Ismail, M.; Shakir, M.M.; Alani, S.; Alsariera, H. A Comprehensive Survey: Benefits, Services, Recent Works, Challenges, Security, and Use Cases for SDN-VANET. *IEEE Access* **2020**, *8*, 91028–91047. [CrossRef]

21. Ku, I.; Lu, Y.; Gerla, M.; Gomes, R.L.; Ongaro, F.; Cerqueira, E. Towards software-defined VANET: Architecture and services. In Proceedings of the 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), Piran, Slovenia, 2–4 June 2014; pp. 103–110.
22. Truong, N.B.; Lee, G.M.; Ghamri-Doudane, Y. Software defined networking-based vehicular Adhoc Network with Fog Computing. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 1202–1207.
23. King, D.; Rotsos, C.; Aguado, A.; Georgalas, N.; Lopez, V. The Software Defined Transport Network: Fundamentals, findings and futures. In Proceedings of the 2016 18th International Conference on Transparent Optical Networks (ICTON), Trento, Italy, 10–14 July 2016; pp. 1–4. [CrossRef]
24. Olimjonovich, M.S. Software Defined Networking: Management of network resources and data flow. In Proceedings of the 2016 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2–4 November 2016; pp. 1–3. [CrossRef]
25. Wu, Y.; Zheng, J. Modeling and Analysis of the Local Delay in an MEC-Based VANET for a Suburban Area. *IEEE Internet Things J.* **2022**, *9*, 7065–7079. [CrossRef]
26. Zhang, S.; Luo, H.; Li, J.; Shi, W.; Shen, X. Hierarchical Soft Slicing to Meet Multi-Dimensional QoS Demand in Cache-Enabled Vehicular Networks. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2150–2162. [CrossRef]
27. He, Y.; Wang, D.; Huang, F.; Zhang, R.; Gu, X.; Pan, J. A V2I and V2V Collaboration Framework to Support Emergency Communications in ABS-Aided Internet of Vehicles. *IEEE Trans. Green Commun. Netw.* **2023**. [CrossRef]
28. Ho, M.C.; Lim, J.M.Y.; Chong, C.Y.; Chua, K.K.; Siah, A.K.L. Collaborative Vehicle Rerouting System with Dynamic Vehicle Selection. *IEEE Trans. Intell. Transp. Syst.* **2023**. [CrossRef]
29. Ge, X.; Li, Z.; Li, S. 5G Software Defined Vehicular Networks. *IEEE Commun. Mag.* **2017**, *55*, 87–93. [CrossRef]
30. He, T.; Toosi, A.N.; Akbari, N.; Islam, M.T.; Cheema, M.A. An Intent-based Framework for Vehicular Edge Computing. In Proceedings of the 2023 IEEE International Conference on Pervasive Computing and Communications (PerCom), Atlanta, GA, USA, 13–17 March 2023; pp. 121–130. [CrossRef]
31. Buda, S.; Guleng, S.; Wu, C.; Zhang, J.; Yau, K.L.A.; Ji, Y. Collaborative Vehicular Edge Computing Towards Greener ITS. *IEEE Access* **2020**, *8*, 63935–63944. [CrossRef]
32. Bylykbashi, K.; Qafzezi, E.; Ampririt, P.; Ikeda, M.; Matsuo, K.; Barolli, L. Performance Evaluation of an Integrated Fuzzy-Based Driving-Support System for Real-Time Risk Management in VANETs. *Sensors* **2020**, *20*, 6537. [CrossRef] [PubMed]
33. Zadeh, L.A.; Kacprzyk, J. *Fuzzy Logic for the Management of Uncertainty*; John Wiley & Sons, Inc.: New York, NY, USA, 1992.
34. Kandel, A. *Fuzzy Expert Systems*; CRC Press, Inc.: Boca Raton, FL, USA, 1992.
35. McNeill, F.M.; Thro, E. *Fuzzy Logic: A Practical Approach*; Academic Press Professional, Inc.: San Diego, CA, USA, 1994.
36. Zimmermann, H.J. Fuzzy control. In *Fuzzy Set Theory and Its Applications*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 203–240.
37. Munakata, T.; Jani, Y. Fuzzy systems: An overview. *Commun. ACM* **1994**, *37*, 69–77. [CrossRef]
38. Klir, G.J.; Folger, T.A. *Fuzzy Sets, Uncertainty, and Information*; Prentice Hall: Upper Saddle River, NJ, USA, 1988.