Slovak University of Technology, Bratislava,
Faculty of Informatics and Information Technologies

Ing. Ján Nemčík

Dissertation Thesis Abstract

Quality of experience optimization in V2X communication

to obtain the Academic Title of philosophiae doctor
(abbreviated as „PhD.“)

in the doctorate degree study programme:

12826 D-AI Applied Informatics

in the field of study: Computer Science

Form of Study: full-time

zabezpečovaného v spolupráci s vysokou školou/vysokými školami Slovenská technická univerzita v Bratislave

Place and Date: 13.8.2025

Dissertation Thesis has been prepared at

Faculty of Informatics and Information Technologies, Slovak University of Technology, Bratislava, Slovakia

Submitter	Ing. Ján Nemčík Automotive Innovation Lab Ilkovičova 6276/2 842 16 Bratislava 4
Supervisor	prof. Ing. Ivan Kotuliak, PhD
Consultant	doc. Ing. Lukáš Šoltés, PhD

Dissertation Thesis Abstract was sent: 13.8.2025

Dissertation Thesis Defence will be held on 26.8.2025 **at** 13:15 **h at** Faculty of Informatics and Information Technologies, Slovak University of Technology,Ilkovičova 6276/2, 842 16 Bratislava 4, room 2.06.

prof. Ing. Ivan Kotuliak, PhD

Abstract

The growing number of connected vehicles is referred to as the Internet of Vehicles. Our research firstly classified different communication types and quality of service improvement on the edge servers in the IoV, mainly in V2X communication with surrounding services. In the second part of the thesis, we focused on different network parameters between the CAV and supporting infrastructure, primarily on the edge servers' communication latency and computation task processing period. As part of our analysis, we described existing solutions for vehicle-to-edge communication with the primary concern of increasing the quality of service. We studied several optimization methods resulting in better latency and effective resource management at the edge servers in this context. At the end of the analyzed part, we evaluated and proposed possible improvements in resource orchestration and management, providing acceptable latency at the edge servers, where the primary purpose is balanced and effective computational resource management. Based on the analysis of the studied optimization approaches and the possible modifications to be used in our work, we proposed their inclusion in the final solution. The proposed orchestration platform for the IoV environment supporting the V2X communication is tested in a simulated environment, and the results are compared to existing solutions in this domain.

Contents

1. Introduction

We are on the brink of a revolutionary era in Connected Autonomous Vehicles (CAVs), it is expected to bring a lot of significant changes in user experience, road safety, environmental impact, and the development of advanced applications [1]. The realization of this ambitious vision relies on the establishment of a strong and advanced infrastructure for the Internet of Vehicles (IoV). This infrastructure should possess the capability to efficiently handle a vast amount of requests and data concurrently [2]. The achievement of this vision relies on the capacity of the underlying infrastructure to fulfill these requirements, aiming for ultra-reliable, high-speed, and low-latency information exchange [3].

The emergence of edge, fog, and cloud computing has brought about a significant change in the way tasks are handled. This new paradigm enables vehicles to transfer their workload to highly capable servers located nearby [4]. The popularity of this approach has increased thanks to the introduction of 5G technology, which is well-known for its low latency and high-speed data transfer capabilities [5]. The upcoming 6G network technology is expected to have a significant impact on the field of Internet of Vehicles (IoV) [6]. It is anticipated to introduce novel materials, innovative technologies, and advanced algorithms, thereby revolutionizing the IoV landscape.

Our research aims to thoroughly examine and enhance the processes involved in task offloading from vehicles to the cloud infrastructure, specifically referred to as Vehicle-to-Cloud (V2C) or Vehicle-to-Infrastructure (V2I). Our objective is to enhance the Quality of Service (QoS) in the IoV domain by enabling real-time computation and processing capabilities while minimizing any negative impact on the user's experience. We specifically emphasize improving Vehicle-to-Everything (V2X) communication. In order to achieve this objective, we thoroughly analyze the current methodologies in this field, with the goal of introducing new and improved strategies for effectively managing and coordinating computational resources at the edge. Our research is focused on developing a dependable and efficient infrastructure to support the upcoming 6G network and its significant impact on the IoV. This infrastructure aims to provide reliable connectivity, minimize delays, and optimize resource usage. The establishment of this foundation is crucial for meeting the complex demands of 6G-V2X networks, which will usher in a new epoch of vehicle communication and connectivity.

2. Internet of Vehicles

Nowadays, transportation systems are increasingly being intensified, hitting their limit globally as their usage keeps growing rapidly. However, many systems are inefficient, and the upkeep and upgrade costs are becoming marginal [7]. Currently, there are more than one billion vehicles used in the world, with a prediction to hit two billion by 2035, which could increase traffic congestion and a number of car accidents [8]. To cope with the evolving and emerging requirements and paradigms like the Internet of Things (IoT), cloud, edge, and fog computing have led to the development of intelligent devices to face and decrease the impact of such rapid growth. In addition, new opportunities for services and products are arising with the widely used 5G for data transfer and with the upcoming 6G, which is ubiquitous and still more affordable.

To seize the aforementioned favourable circumstances in the vehicular network with many connected vehicles, we refer to the IoV. Therefore, we can define IoV as a smart transportation area which consists of three components, namely vehicular mobile Internet, intra-vehicle network, and inter-vehicle network as shown in Figure 2.1[9].

Similar to other paradigms or applications, IoV also has specific requirements and characteristics it needs to achieve [10, 11]:

1. scalable and flexible architecture
2. unloading scheme and resource allocation
3. Software Defined Network (SDN) based V2X architecture with mobile edge computing
4. identity authentication and privacy protection
5. delay constraints

6. fault tolerance

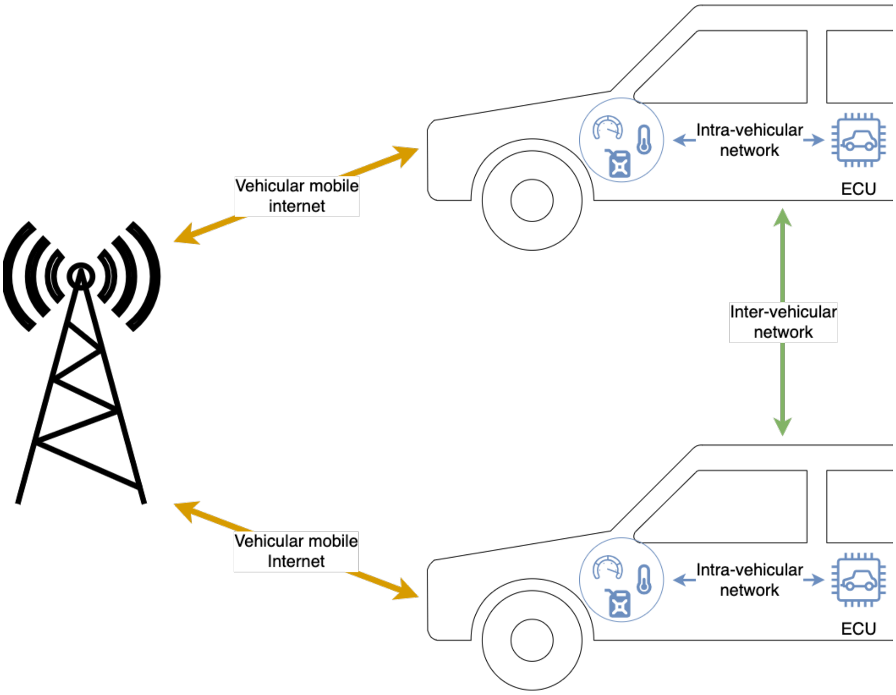

Figure 2.1: IoV network model

2.1 Architecture model of IoV

The architecture of the IoV is a complex framework designed to facilitate advanced communication and exchange of data among vehicles. The integration of diverse technologies, including wireless communication, cloud computing, and edge computing, facilitates the development of intelligent and interconnected transportation systems. Based on the established definition of the IoV, it becomes apparent that the architectural framework, although appearing to be uncomplicated, presents numerous obstacles that must be addressed. One of the primary considerations pertains to the comprehensive integration of various elements, encompassing vehicles, sensors, communication systems, roadside infrastructure, devices, and human involvement. The desired result entails enhancing driving safety, optimizing comfort, promoting smoother traffic flow, and maximizing fuel or battery efficiency. IoV plays an important role in enhancing vehicular safety and bridging the gap between traditional and intelligent automotive industries. It highlights the necessity of incorporating safety measures in response to road accidents, positioning IoV as a key innovator in intelligent transportation systems [12].

Despite the simplified overview of the IoV network model, several researchers have already analyzed and defined the IoV network layer model [8, 12, 13]. It consists of seven layers, each responsible for a particular part of the network components interconnection chain. The proposed seven-layer IoV architecture model is defined as follows [8]:

- user interface layer - provides direct interaction with the driver, e.g. infotainment
- data acquisition layer - collects data from various sources located on the roads
- data filtering and preprocessing layer - serves to analyze the collected information to filter irrelevant information and reduce network congestion by unrelated data
- communication layer - selects the best network to send the information
- control and management layer - responsible for managing different network service providers that are within the IoV environment
- processing layer - processes large amounts of information using various types of computing infrastructures
- security layer - communicates with all other layers and is responsible for all security functions

The layered architecture of the IoV represents a comprehensive framework that integrates advanced communication technologies, robust data management, and secure, interconnected systems to revolutionize the automotive and transportation industries. The IoV architecture encompasses various layers, including intra- and inter-vehicle communication, as well as integration with cloud and edge computing. Each of these layers assumes a crucial role in augmenting vehicle capability, safety, and efficiency. The cohesive interplay among these strata guarantees a smooth flow of data, instantaneous data processing, and the provision of a diverse array of services, hence facilitating the development of intelligent, secure, and highly efficient transportation systems.

2.2 Vehicle-to-Everything communication

As we already presented, in the IoV network, there is a lot of ongoing communication between different types of devices, environments, and even humans. All of this communication can be referred to as V2X communication. V2X technology creates a more comfortable, safer transportation environment and significantly improves traffic efficiency [14].

The V2X concept involves various communications technologies, including Vehicle-to-Vehicle (V2V), V2I, Vehicle-to-Pedestrian (V2P), Vehicle-to-Network (V2N), and V2C network connections [12, 14, 15, 16]. This technology connects numerous aspects of transportation, such as vehicles, pedestrians, roads and cloud. V2X contributes to the development and evolution of new transportation services and, therefore, essentially improves traffic efficiency, pollution reduction, resource-saving, and traffic management [14]. It enables ubiquitous access to information for drivers and passengers, raising the importance of robust security and efficient communication protocols [8].

The application of V2X in real-life covers many aspects, such as intelligent transportation systems, connected vehicles, and automated driving. Like any other application, also V2X applications have their requirements to meet. Different V2X applications have different needs in terms of latency, throughput, reliability, and safety in the V2X environment [14]. Automated driving requires extremely low latency and a secure network to operate. As a result, the security and reliable underlying infrastructure is the biggest priority for V2X.

The V2X sector is considered to be one of the most rapidly advancing areas within the industry. The subject matter encompasses various aspects such as automobiles, transportation, and communication, all of which are interconnected with the Internet. Extensive research has been conducted on the prerequisites for V2X applications. The applications in the V2X domain can be categorized as follows [17]:

- safety applications - utilized to enhance human safety by providing various forms of warnings, such as collision and speeding alerts,
- efficiency applications - aim to provide drivers with guidance to optimize traffic flow and enhance fuel or battery consumption,
- information services - offer a wide range of informational resources to drivers.

In conjunction with the advancement of information technology, there will be a continuous emergence of new demands for V2X in the context of automatic driving and intelligent transportation systems. The development of V2X technology has led to the establishment of four distinct application categories as defined by the 3GPP [18]. These categories include remote driving, advanced driving, vehicle platooning, and extended sensors. Nevertheless, it is anticipated that the V2X business will continue to witness an influx of further applications in line with the prevailing trajectory. Hence, the 3GPP has established certain criteria for various categories of V2X applications, with a particular emphasis on factors such as latency, dependability, and safety, which are of utmost importance.

The IoV network model facilitates a high degree of connectivity and interaction between vehicles, infrastructure, and the broader network. This model not only enhances vehicle safety and efficiency but also contributes significantly to the development of smart cities and intelligent transportation systems [9, 16, 19, 20, 21, 22]. Its ability to harness real-time data, support advanced driver-assistance systems, and integrate with emerging technologies like 5G and edge computing underscores its vital role.

2.3 Computing infrastructure

V2X couples many IoV network layers. In this work, we mainly focus on the two highest of them: the control and management layer and the processing layer. As we described in Section 2.1, there is a massive amount of data gathered in intra- and inter-vehicular networks that needs to be processed and evaluated to ensure a better user-, application-, and service experience within the V2X communication. Zhou H. et. al presented several challenges in data sourcing and transmission [23]. They stated that the integration of cloud computing has led to an increased focus on vehicular services closely related to vehicles, underlying the necessity to build proprietary vehicle cloud platforms. This means specific computational tasks need to be offloaded from the vehicles and devices to the more powerful computational units sitting at the edge or in the cloud. Therefore, this section introduces three paradigms used for computational task offloading in V2X communication: cloud, fog, and edge computing.

2.4 Quality of Experience in Vehicle-to-Everything communication

Presently, there is a growing number of vehicles that are connected to the IoV, making V2X communication increasingly crucial. Many modern automobiles already rely on intra- and inter-vehicle communication in various ways. Numerous solutions have been developed to enhance the QoS for this type of communication. The utilization of 5G and Mobile Edge Computing (MEC) in the IoV domain presents new opportunities for the development of applications and services that aim to improve the end user experience. However, the evolving trends in the IoV are placing greater demands on the network infrastructure and V2X communication. As illustrated, the share of connected cars is expected to undergo significant changes in the coming years, a finding that is also supported by a study conducted by Padmaja B. et al. [24].

The concept of Quality of Experience (QoE) holds significant importance within the field of telecommunications and digital services, as it encapsulates the holistic perception of a user's encounter with a particular system or service. In contrast to QoS, which is assessed based on objective network parameters such as latency, packet loss, and jitter, QoE encompasses a wider and more subjective assessment of the end-user's overall satisfaction [25]. The integration covers a diverse range of elements, including the user's anticipations, surrounding circumstances, and the inherently subjective character of the encounter. QoE builds on the recognition that the user's perception and satisfaction cannot be solely determined by the technical performance of a network or service. The adoption of a user-centric perspective redirects attention away from solely emphasizing technical efficiency towards the evaluation of perceived value and quality as subjectively experienced by the user.

The QoE within the context of the IoV encompasses a diverse array of factors due to the integration of intricate and diverse technologies, such as the IoT, vehicle communication systems, data analytics. As we already mentioned in Section 2.2, V2X applications will play a significant role in allowing safer, more reliable, and more efficient traffic flow. Nevertheless, even if the applications meet the expected requirements, the underlying network and infrastructure must meet even higher standards than the applications themselves. The grand vision is that V2X communications, supported by 5G/6G, will be an essential element of future CAVs. Furthermore, V2X communications will bring innovative benefits, such as unprecedented user experience, exceedingly improved road safety and air quality, various transportation applications, and a plethora of advanced applications [26]. The review presented by Garg S. et al. discusses various IoV models that aim to revolutionize transportation and automation industries [27]. It highlights the considerable impact of IoV technology on these sectors, particularly in terms of enhancing transportation utility and efficiency. Despite the positive outcomes, Damaj, I. W. et al. conducted a critical evaluation of recent advances in the field, identifying challenges and gaps in current technology [28]. They propose improvements to address these issues and advance the field of CAVs, where QoE is considered in terms of system, contextual, and human factors. The shift towards CAVs is driven by environmental and sustainability concerns, with the integration of connected and autonomous components indicating a trend towards more advanced and user-centric vehicle technologies.

Minovski D. et al. [29] highlight the necessity of developing novel approaches for assessing QoE within the framework of the IoT and autonomous vehicles. The conventional methods used to evaluate QoE, which primarily concentrate on multimedia services, must be revised to address the intricacies arising from the integration of IoT technologies in vehicular environments. In this study, Cheng et al. emphasised the significance of assessing novel methodologies that impact the QoE for end users [25]. The findings indicate that these innovative approaches are capable of accommodating the IoV, thereby leading to improvements in the core operations of service providers and the overall user experience. Hussain S. A. et al. [30] underscore the need for improving several measurement parameters that can have a substantial impact on the performance of the IoV. This emphasis is of utmost importance given that IoV technology has recently emerged as a noteworthy global innovation, exerting a profound influence on the transportation and automation sectors.

The substantial volume of data generated by V2X communication underscores the necessity of quickly collecting, processing, and evaluating it. Consequently, it becomes imperative to explore several branches of artificial intelligence in order to enhance these processes and gain a comprehensive understanding of the data. Hasan, M. K. et al. explored the use of machine learning and artificial intelligence techniques to improve the security and communication of the IoV [31]. They engaged a discussion regarding the potential benefits of machine learning in enhancing road safety, optimising traffic management, and facilitating efficient data processing. These advancements are expected to immediately enhance the

QoE [12]. While it is assumed that a significant amount of data needs to be collected, the utilization of various processing techniques would become essential [32]. Hashem I. A. T. et al. forecasted a substantial increase in the number of linked vehicles within IoV environments, reaching several hundred million by the year 2035 [33]. The significance of modern data analytics, namely utilising deep learning methodologies, was emphasised within the realm of the IoV.

As we discussed several challenges that appear to be crucial for the success of IoV, a major significance will be placed on the computing infrastructure to overhaul the amount of load produced by the devices connected in V2X networks. The aspects presented include the management of computational resources and their allocation. Salem A. H., Damaj I. W. and Mouftah H. T. introduced a novel concept in addition to Vehicular Edge Computing (VEC) in order to enhance the computational resources available for V2X requirements in smart cities [34]. The researchers focused their attention towards the advancement of computational capabilities for vehicles and the development of a QoE model specifically designed for connected vehicles, thus demonstrating the potential benefits of utilizing vehicle resources to improve the overall QoE. Wu G., Li Z. and Jiang H. discussed the improvement of user experience by proper resources allocation spectrum in VEC [35]. It highlights the importance of QoE as a metric for user satisfaction with provided services, especially in large-scale vehicle networks.

Based on our actual analysis of IoV infrastructure, we decided to further analyze task offloading techniques from connected vehicles to available computing nodes in close proximity utilizing V2X communication. There already exist several different approaches to optimizing task offloading from vehicle to edge and from edge to cloud. However, with rapidly evolving technologies and developing more robust and advanced applications for V2X, we anticipate a rapid growth in the data produced by IoV. To elaborate on and process all of that data, optimized task offloading will play a significant role in observing the low latency required by V2X and providing enough computational resources to handle such a load. The authors of optimized solutions acknowledged that First-In-First-Serve (FIFS) queuing in IoV communication is not sufficient to meet all V2X application requirements.

2.5 Analysis summary

The demand for improved user experience and increased road safety creates new challenges for V2X communication. This demand is accompanied by an increase in the number of connected cars. The V2X domain contains the potential for participation from a number of different research fields, including V2V, V2P, V2I, and V2C communication. At present, there are already an enormous number of connected cars, which results in a massive number of tasks that need to be processed and evaluated. It is necessary to have efficient management of computational resources in order to enable an IoV environment to support such a load and improve the QoE.

Based on the analysis of different task offloading proposals, ranging from offloading to neighbouring vehicles to vast cloud computing infrastructure, we identified several challenges and areas to be considered for task offloading in IoV. A very different approach to increasing QoE in the V2X environment is to increase security aspects in response to the current traffic situation. Such a concept was presented by the authors of [36], who formulated a solution for context-based application placement. It is essential to provide contextual application placement to increase customer satisfaction and road-level safety. Thus, context changes caused by, e.g., soft- or hardware failures will not impact safety on the road and will even handle emergency stops if needed.

A determination of an assignment between computing nodes and applications that are to be placed on those nodes is referred to as the application-placement problem [37]. Additionally, it is necessary to fulfill a number of application placement requirements, such as redundancy or hardware segregation. The generation and retrieval of data in IoV networks is becoming an increasingly difficult task to accomplish as new and innovative V2X applications emerge. In such a diverse environment, where networks are shared among various kinds of data traffic, problems such as increased packet loss, jitter, latency, and lower bandwidth become apparent. A novel approach to artificial intelligence-based content placement in IoV networks was presented by the authors in [38]. The authors took into consideration the following tasks that could be leveraged:

- content popularity prediction,
- content placement,
- content retrieval.

The task offloading methodology within the realm of the IoV presents a multitude of challenges, each of which requires careful consideration and strategic solutions. These challenges are in addition to the aspects that have been mentioned, which include the generation of data, transmission of data, and various diverse parameters that affect the placement of the application within the context of the IoV. These challenges include, but are not limited to, technical, operational, and infrastructure-related aspects, all of which are essential to the successful implementation and operation of IoV systems.

In spite of the fact that task offloading has a number of advantages, it also creates dependencies that may have a negative impact on the QoE in general. There is a possibility that the increase in network congestion could result in an increase in latency, while the performance of external servers might result in a delay in the computation of time [39]. Consequently, in order to achieve high quality of experience levels, it is necessary to guarantee the scalability of the underlying infrastructure and make adjustments where necessary.

Nowadays, the protection of user data and privacy is of the utmost importance. The task offloading application within the IoV ecosystem presents security risks that necessitate changes to the infrastructure. IoV is a combination of a number of different technologies, services, and standards [40]; consequently, it is essential to implement secure gateways, anomaly detection systems, and dynamic application orchestration [41]. As a result of the vulnerability of IoV environments, cybercriminals have the ability to easily take control of vehicles and exploit vehicular data streams, which can result in damage to infrastructure and injuries to passengers. As a result, secure communication schemes need to be developed and implemented in order to mitigate the impact of this vulnerability [8].

3. Dissertation thesis objectives

Nowadays, autonomous vehicles are becoming every day's life routine. Since road safety plays an emerging role in the IoV domain, research in this field of information technologies does have a significant impact on crucial improvements.

With the increasing amount of connected cars, the demand for road safety and a better user experience comes with new challenges for V2X communication. There are several fields of research that can take part in the V2X domain, such as V2V, V2P, V2I, V2C communication. Currently, with many connected cars already, there is a huge load of tasks created to be processed and evaluated. In order to support such load, the effective computational resource management is necessary. Hence, we divided present research into two main parts:

1. Overview of IoV architecture, communication layers and their interconnection. Furthermore, we analysed several deployment models, where we described their advantages to additionally be able to identify suitable approach for our research

2. Analysis of existing task offloading algorithms developed for V2X communication needs. Because of different types of applications deployed in IoV networks and communicating with vehicles, there could be several ways to approach the problem as we described in Section 2.5.

Based on the analysis, there are three possible ways for optimization to take part. The task offloading decision algorithm could decide based on the computational resource effectiveness to ensure that computational nodes are equally loaded and are able to handle as many tasks as possible. The other approach would be to prioritize tasks based on their context criticality. The last approach is the resource optimization of neighboring parked vehicles, where the tasks could be processed.

According to our analysis of task offloading in IoV networks and identification of possible optimizations, we formulated the following objectives for this thesis:

- Proposal of task offloading methodology using new and existing optimization methods, aiming to combine effective computational resource management and context criticality of applications
- Verification and evaluation of implemented solution with aim to effective resource management and applications context criticality
- Verification and evaluation of proposed solution deployed in simulated V2X environment
- Comparison of proposed methodology to existing solutions

4. Problem proposal

In this Chapter we describe and deeper explain our objectives formulated in Chapter 3. For a better understanding of our efforts towards the task offloading, first we briefly describe what task offloading means and why it is important in such a heterogeneous environment as the IoV. In general task offloading aims to offload high intensive computational tasks from the end user to a remote site [42]. The offload process occurs under various constraints and consists of several hardware components, such as the end-user device producing the task, the network layer for data transmission, and cloud computational resources, as described in Section 2.3, for the task execution.

Existing algorithms offer different types of task offloading processes. Tasks could be offloaded to edge, cloud nodes, or partially to both respectively. For effective task offloading, there are also some network and resource allocation challenges [43]. As we mentioned, the IoV environment is heterogeneous, it creates dynamic inputs, such as network conditions or autonomous car behavior. The computational platform also has its dynamics, stating that cloud nodes are centralized and edge nodes are distributed, which affects the offloading as well.

Although, in our work we focused on resource allocation challenges. As authors presented in [43], the most important challenges are: partitioning decision, resource availability, task management and performance modeling. Due the diversity, it is difficult to propose an task offloading methodology optimizing all of the presented challenges. In our proposal, our goal is to create a joint approach combining context-based partitioning decision and task management based on resource availability of computational resources. In following sections we describe our understanding of these inputs for our methodology proposal.

5. Task offloading method design

In this chapter, we describe our task offloading algorithm based on the analyzed requirements and needs of IoV and the formulated problem in Chapter 4. In our case, we consider the contextual priority of the task, e.g., defined by vehicle speed and task maximal tolerable delay, which is crucial for some use cases in a V2X environment where late response is not considered any more [44, 45]. We also aim to maximally utilize the edge resources so that it can serve as many tasks as possible, the rest of the tasks will be offloaded to the cloud. We also optimize the reallocation and offloading rate of lower priority tasks with respect to the incoming tasks and their resource requirements in real-time manner. To solve these requirements we formulate custom Knapsack problem modification and propose and algorithm to solve it. The rest of this section is organized as follows: first, we introduce formulation of custom Colored Multiple Knapsack problem and the we explain the design of the proposed method, then we describe the architecture design of our solution and finally we propose our simulation environment.

5.1 Problem formulation

Based on the identified challenges regarding task offloading in V2X scenarios, we propose a task offloading method with respect to the task's required maximal computation time, task priorities and node resource capacities. The computation time required by tasks is defined by V2X application necessities. The priority is defined by the context in which the vehicle is presently situated and could be denoted by speed, urban area or weather conditions. Since our problem is targeting task offloading optimization, we have to consider fitting the maximum amount of tasks into the edge computation layer so that the tasks could be delivered with lowest latency possible, which is one of the critical requirements in the V2X environment [46].

As we discussed in the previous section **??**, to solve this problem, we formulate Multiple Knapsack Problem with Color Constraints (MKCP) and explain the design of our Edge-to-Cloud Offloading Decision Algorithm (EAODA). To formulate MKCP, we consider the priority of the item as the value of the item, the proportional task CPU and memory requirements as the item's weight, and the knapsack affinity as the item color. We assume that knapsacks can accept items of multiple colors. The assignment is bivalent - a task is either allocated to a node or not. The mathematical model can be stated as follows:

$$\max \sum_{j=1}^{m} \sum_{i=1}^{n} w_i x_{ij} \tag{5.1}$$

subject to:

$$\sum_{j=1}^{m} x_{ij} \leq 1, j \in M = \{1, ..., m\} \tag{5.2}$$

$$\sum_{i=1}^{n} x_{ij} w_i \leq c_j, i \in N = \{1, ..., n\} \tag{5.3}$$

$$x_{ij} \in 0, 1, i \in N, j \in M \tag{5.4}$$

$$y_j^c \in 0, 1, \forall c \in C_j, j \in M \tag{5.5}$$

$$x_{ij} \leq y_j^{c(i)}, i \in N \tag{5.6}$$

where:

x_{ij} - is a binary variable expressing whether the object i has been inserted into the knapsack j

y_j^c - is a binary variable expressing whether the object of color c has been inserted into the knapsack j

w_i - is the ratio of i object to knapsack capacity

n - is the number of items

m - is the number of knapsacks

c_j - is the capacity of knapsack j

C_j - is the color set of knapsack j

5.2 Proposed method

Edge-to-Cloud Offloading Decision Algorithm has been developed to address the challenge of task offloading optimization at the edge in a V2X environment. Referring to the MKCP, the tasks are represented as items in the knapsack with value denoted as task priority, weight represented as the task's CPU and memory requirements, and color, which restricts the task's maximum completion time. The edge nodes indicate knapsacks to be filled with items and tasks, with constraints on CPU and memory capacity and colors guaranteeing the completion time for each allocated task. We assume that nodes can allocate tasks with lower completion expectations. Tasks are assumed to have the capability to store preliminary results; thus, the offloading of a task would not require the restart of the computation process.

The algorithm operates based on a progressive inclusion of elements, hence prohibiting the pre-arrangement of objects beforehand. The situation implies an optimization strategy that enables the exchange of items between multiple knapsacks. This allows us to increase the capacity of specific knapsacks, thereby making space to fit an extra item that would otherwise not be assigned.

For the purpose of the algorithm, we define a knapsack problem Resource Allocation Knapsack Problem (RAKP), an instance of the MKCP, as follows:

$$x_{tj} \begin{cases} 1 & \text{if task } t \text{ is assigned to node } j \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

The objective of the algorithm is to:

$$\max \sum_{j=1}^{m} \sum_{t=1}^{n} \sigma_t x_{tj} \tag{5.8}$$

where σ_t represents the objective function defined as:

$$\sigma_t = w_t + p_t \tag{5.9}$$

While trying to fulfill the objective defined in 5.8, following constraints must be satisfied:

$$x_{tj} = \{0, 1\}, \forall t \in \{1, ..., n\}, \forall j \in \{1, ..., m\} \tag{5.10}$$

$$\sum_{t=1}^{n} x_{tj} w_t \leq c_j, \forall j \in \{1, ..., m\} \tag{5.11}$$

$$y_j^c = \{0, 1\}, \forall c \in C_j, j \in \{1, ..., m\} \tag{5.12}$$

$$x_{tj} \leq y_j^{c(t)}, t \in \{1, ..., n\} \tag{5.13}$$

where c_j represents capacity of a node, the constraint 5.11 must be true for both, CPU and memory resources, y_j^c is binary variable indicating if node j can allocate tasks with color c, C_j are colors of a node and x_{tj} is defined in 5.7. Constraint 5.13 ensures that the task's color $c(t)$ is evaluated before allocation decision x_{tj}, tasks with not matching color cannot be allocated to a particular node. The priority p_t denotes task's priority $p \in P = \{1, ..., l\}$, where l represents the maximum priority. The weight of a task w_t is modeled as normalized ratio of resource utilization for a specific node:

$$w_t = \frac{\frac{\mathcal{C}_j - w_{ct}}{\mathcal{C}_j} + \frac{\mathcal{M}_j - w_{mt}}{\mathcal{M}_j}}{2} \tag{5.14}$$

where w_{ct} and w_{mt} represent CPU and memory requirement of tasks t, $\mathcal{C}_j$ and $\mathcal{M}_j$ denote the maximal allowed CPU and memory capacity for node j.

Tasks without defined CPU and memory requirements are not considered by RAKP and are assumed to be automatically processed by cloud nodes. There are several possibilities for handling not allocated tasks with specified CPU and memory requirements. In our scenario, these tasks are considered to be offloaded to the cloud and decision regarding these tasks is not in the scope of this thesis. We also consider the situation, that tasks have resource requirements corresponding to the maximal computation time requested by their color. These assumptions are described in 5.11 and 5.12, hence we are not allocating the tasks based on actual resource usage, which is crucial for solving the RAKP. Such a situation could possibly lead to node overcommitment and concurrency issues, resulting to CPU throttling or Out of Memory (OOM) errors.

EAODA presented in this thesis avoids a complete enumeration of all possible solutions by minimizing the possible number of reallocations. If the new task that requests the node to be allocated to cannot be assigned directly to any of the nodes since there is not enough available CPU and memory resources, EAODA tries to reallocate any of its existing tasks to any other node, relaxing it's original capacity enough to allocate the new task. If there is still not enough relaxed space on any node for the new task, then EAODA finds, based on the objective function, less important tasks than the new task. Following this heuristic, EAODA can quickly identify if the amount of relaxed resources by these tasks is sufficient. If so, the task is allocated to the node; otherwise, it is offloaded to the cloud. In the case of task allocation, for the lower objective released tasks, EAODA tries to find a suitable node for allocation. If the task can't be allocated to any remaining nodes, it is offloaded to the cloud to continue it's computation. With this setup, we can minimize the number of needed reallocations; hence, stopping and resuming the computation also costs time, but this parameter is not considered in this work. EAODA must be aware of the current state of task allocations at all times, and after each calculation, the allocation state must be adjusted accordingly.

As already mentioned, EAODA aims to distribute tasks among nodes based on task resource requirements and node resource capacity. In cases of node overcommitment, the resources could be saturated, thus leading to computation delays and not fulfilling the time computation constraints. Therefore, if there is no overall available capacity among all nodes for an incoming task, the task is automatically rejected and considered offloaded.

5.3 Formal verification

To ensure the correctness of EAODA, we need to provide formal verification of the algorithm. Let m be the number of given nodes, R_i be the set of tasks assigned to node i, R_{it} be task t allocated to node i, c_i be the capacity of the node i, K_i be the set of colors of the node i, and we are supposed to assign a given item with weight w and color k. For every given R_i, the capacity and color constraint defined in 5.16 must apply. To prove the correctness of this algorithm, we have to prove that it does not violate the capacity and color constraints, nor does it change the overall number of assigned items other than increasing it by one. We can interpret this statement as following theorem:

Theorem A:

$$\sum_{i=1}^{m} |R_i| \leq \sum_{i=1}^{m} |R'_i| \leq \sum_{i=1}^{m} |R_i| + 1 \wedge \forall i \in \{1, 2, 3, ..., m\} \tag{5.15}$$

$$\sum_{t=1}^{|R'_i|} R'^{k}_{it} \leq c_i, k \in K_i \tag{5.16}$$

where R' represents set after allocation.

The three phases of the algorithm, as stated in Section 5.2, are each represented by a lemma. In these lemmas, the variable s_l^k denotes the weight of items of color k that are transferred to the knapsack l. The variable x indicates the index of the knapsack where the new item is to be added, whereas the variable y represents the index or indices of the knapsacks where the items are to be relocated to:

Lemma 1: In the first phase, the greedy knapsack algorithm phase, the item may only be assigned to a knapsack, but only if

$$\sum_{t=1}^{|R_i|} R_{it}^{k} + w \leq c_i,\ k \in K_i \tag{5.17}$$

Lemma 2: In the second phase, multiple items from one knapsack may be selected to move to one or many other knapsacks, but only if

$$\sum_{t=1}^{m} R_{it}^{k} + w - \sum_{l=1}^{m} s_l^k \leq c_i,\ k \in K_i \wedge \tag{5.18}$$

$$\sum_{b=1}^{|R_y|} R_{yb}^{j} + s_y^j \leq c_y,\ j \in K_y,\ \forall y \in \{1, 2, ..., n\}\ \wedge y \neq i \tag{5.19}$$

Lemma 3: In the last phase, multiple items from one knapsack may be selected to move to one or many other knapsacks or to be offloaded, but only if

$$\sum_{t=1}^{m} R_{it}^{k} + w - \sum_{l=1}^{m} s_l^k \leq c_i,\ k \in K_i\ \wedge \sum_{l=1}^{m} q_l \leq p \wedge$$
$$\sum_{b=1}^{|R_y|} R_{yb}^{j} + z_b s_y^j \leq c_y,\ j \in K_y,\ \forall y \in \{1, 2, ..., n\} - i,\ z \in \{0, 1\} \tag{5.20}$$

Therefore, we can conclude:

$$Lemma\ 1\ \wedge\ Lemma\ 2\ \wedge\ Lemma\ 3 \Rightarrow Theorem\ A \tag{5.21}$$

and so prove that EAODA is correct, as it was proved that no item has been removed from the knapsacks; only one may be added, and that all capacity constraints remain satisfied.

6. Results evaluation

To verify the performance of the proposed algorithm, firstly compared it to the Branch & Bound method, which is commonly used for solving knapsack problem. For the evaluation we used linear programming modeler PuLP for Python [47]. We simulated resource utilization with color constraints and time complexity for different amount of nodes for Branch & Bound and EAODA algorithms. The simulations were designed to check the computation time, to see the algorithm's performance capability of solving the task placement problem and if there is an improvement in time complexity and overall knapsack utilization compared to existing methods used for solving knapsack problem.

We performed the simulations in range of knapsacks from 4 to 20. The Branch & Bound method using PuLP was configured with relative gap tolerance set to 0.05 in order to get results in feasible time duration. The relative gap tolerance ensures that if there is a result found within provided tolerance to current best, the solution current best is then considered the best solution for the problem. In tables 6.1 and 6.2 we compared the most relevant amount of nodes, which we also used in later executed simulations as well. In Table 6.1 we compared time complexity of EAODA and Branch & Bound method. From the results we can see significant increase in time performance with EAODA. The average time gain was 91%. We can also see that EAODA performs better with smaller amount of knapsacks.

Table 6.2 presents the overall utilization of resources achieved by both algorithms. Branch & Bound method has better overall resource utilization by 2.4% on average compared to EAODA. It's also clear, that with the increasing amount of knapsacks, the utilization gap grows concurrently. However, considering the time complexity gain with EAODA and overall resource utilization decrease compared to Branch & Bound it's very beneficial for real time or close to real time decision making systems.

Table 6.1: Time Complexity with different methods [ms]

# of Knapsacks	EAODA	Branch & Bound	Greedy
7	58.93	1122.92	27.14
11	133.25	1494.82	53.29
15	228.99	1910.41	97.04
19	302.94	2218.81	112.68

Table 6.2: Resource Utilization with different methods [%]

# of Knapsacks	EAODA	Branch & Bound	Greedy
7	97.71	98.48	98.12
11	96.19	97.82	97.02
15	94.28	97.36	96.43
19	92.11	96.10	94.38

From the preliminary results described above, we assume that the performance of EAODA is suitable for usage in V2X scenarios. To underline the significance of task rescheduling introduced by EAODA, we also performed simulations for 1000 items for different amounts of knapsacks with and without reallocations to recognize the improvement in resource utilization. The results assumed a knapsack with a capacity of 6 units and 16 memory units. In Table 6.3 we can see the CPU and memory utlization with and without reallocation for 7 knapsacks. The CPU utilization is very similar, even slightly better without reallocations. However, with reallocations, we increased memory utilization by 19%. Consequently, the computation time increased rapidly, but still acceptable for real time systems.

Table 6.5 presents the results with and without reallocations for 19 knapsacks. The CPU utilization is similar to the results for 11 knapsacks. However, the memory

Table 6.3: Resource Utilization Among and Computation time with and without reallocations - 7 knapsacks

	1000 items, 7 knapsacks				
	Resources [%]		Tasks		Computation time [ms]
	CPU	Memory	Reallocations	Unassigned	
wo realloc.	99.66	77.35	-	841.94	0.84
w realloc.	99.13	96.29	55.9	476.42	58.93

Table 6.4: Resource Utilization Among and Computation time with and without reallocations - 11 knapsacks

	1000 items, 11 knapsacks				
	Resources [%]		Tasks		Computation time [ms]
	CPU	Memory	Reallocations	Unassigned	
wo realloc.	99.65	77.64	-	752.15	1.30
w realloc.	99.26	93.12	91.96	273.55	133.25

Table 6.5: Resource Utilization Among and Computation time with and without reallocations - 19 knapsacks

	1000 items, 19 knapsacks				
	Resources [%]		Tasks		Computation time [ms]
	CPU	Memory	Reallocations	Unassigned	
wo realloc.	99.67	76.77	-	576.31	2.40
w realloc.	99.43	84.78	99.97	94.76	302.94

utilization increased only by 8%. The computation time difference is also similar to the results for 11 knapsacks.

The outcome of the provided results shows that reallocations improve the performance of fewer knapsacks, which is also underlined by the results of 11 knapsacks shown in Table 6.4. The distribution of all values was random. The CPU distribution was a random value between 0.05 and 0.5 units. The memory distribution was a random value between 0.1 and 1 unit. The color distribution was random between 3 different colors, and the knapsacks had a predetermined color distribution, where the colors were assigned in ascending order from 1 to 3 with an increasing number of knapsacks. Therefore, we assume that the randomized values caused the unequal distribution between CPU and memory resource requests. Also, the reason why EAODA performs better for fewer knapsacks is defined by the algorithm design, where resource requests of an item are normalized, thus we are not favoring any resource type. Consequently, with a fixed amount of items and more knapsacks, we are performing more reallocations to find an optimal solution that favors the priority and meets the resource and color constraints. Moreover, the unassignment rate of tasks with reallocations is significantly better, although it increases the computation time.

6.1 Environment model

Based on the initial evaluation of time performance and resource utilization we realized real world scenario simulations. To be able to perform customized traffic simulations, we utilizedthe MOSAIC simulation framework, which internally uses SUMO simulator for traffic simulation but allows custom applications to be deployed to simulated vehicles [48, 49]. The architecture consists of two main components, MOSAIC framework and Simulation manager, which are further divided. As we already mentioned, MOSAIC internally uses SUMO for traffic simulation and deploys applications on top of it. In our case, we create a simple application which tracks the vehicle speed and based on it creates a new task for processing. Since the change of the speed plays an important role in contextual behaviour of moving vehicle and to simulate the potential changes, we decided to create task when delta of current vehicle speed and last task creation speed was greater or equal to 20 km/h.

The simulation manager is responsible for processing and preparing datasets. Furthermore, it runs the simulations for algorithms evaluation, processes and aggregates the results. Given that we want to verify performance of EAODA algorithm in different real world scenarios, we decided to simulate the traffic in two different cities. Due to the authenticity and proven relevance with real world traffic, we used published simulation scenarios for the cities of Luxembourg Sumo Traffic (LuST) and Berlin Sumo Traffic (BeST) [50, 51]. The major difference between the scenarios is the amount of vehicles through a 24-hour period, where BeST is triple the size, so we were able to evaluate EAODA in bigger and smaller city, thus better determine the suitability for real world deployment. In our simulation, we assumed, that every second vehicle, meaning 50% of all vehicles, are connected to the Internet and could request a computation of a task.

For the simulation, we chose node size of 32000 CPU and 65536 Memory units, but we limited the possible allocated capacity to 90% of the provided size. Additionally, we ran 10 independent simulations for both scenarios, and we aggregated the average values for the final results. We ran the simulations for 7, 11, 15 and 19 nodes respectively, for both scenarios and we summed up the results of both scenarios afterwards into an averaged final comparison. In the figures below, we described only results for 11 and 19 node simulation runs, as they showed the most significant outcomes for the comparison.

For performance evaluation, we compared EAODA to First-In-First-Out (FIFO)

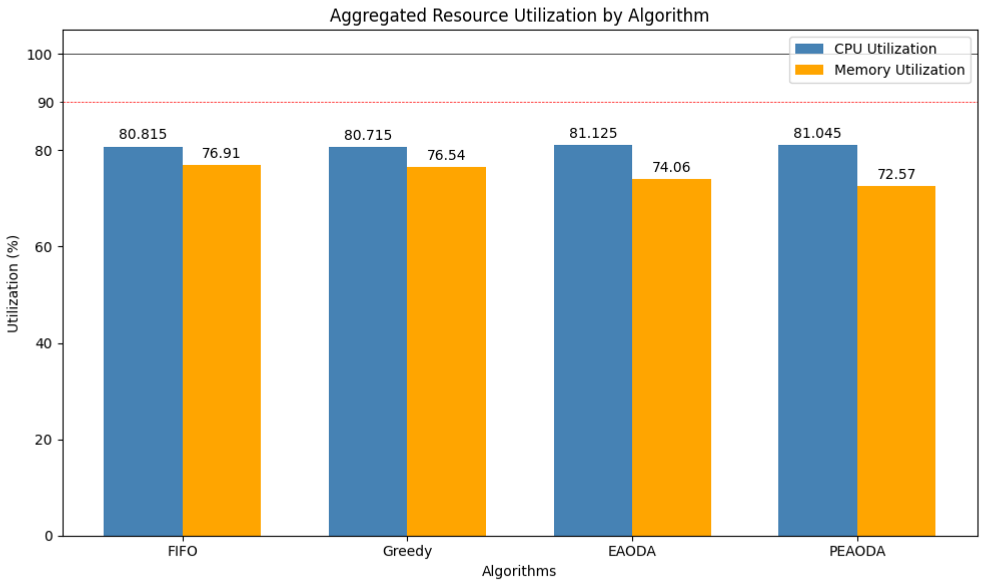

Figure 6.1: Aggregated Average Resource Utilization by Algorithm

and Greedy algorithms, which are one of the most commonly used algorithms for resource load balancing strategy [52].

Since the primary goal is to allocate highest priority tasks to the edge nodes, we also introduced a prioritized version of EAODA. Prioritized Edge-to-Cloud Offloading Decision Algorithm (PEAODA) is modified to favor higher priority tasks, specifically priorities 4 and 5, more decisively compared to EAODA, thus ensuring more critical tasks are allocated on the edge layer but slightly decreasing the overall resource utilization.

6.1.1 Resource utilization evaluation

In Figure 6.1 we can see the resource utilization aggregated from both simulated scenarios. The CPU usage is almost identical among all algorithms. Regarding Memory utilization, FIFO and Greedy outperform EAODA by approximately 2% and PEAODA by roughly 4% overall. To comprehend how these differences would decrease overall memory utilization, we note that the simulation spanned 24 hours with an average of 13 nodes. EAODA and PEAODA decreased memory usage by 20 and 33 GB, respectively, from an overall capacity of 832 GB, indicating that the potential waste is minimal concerning the cost of 1 GB of memory.

To elaborate more on why resource utilization is between 70 and 80 %, there are three main reasons for that. The one is the task distribution is not constant over the whole simulation duration and there are some peeks in both directions. The second is that we limit the maximal usage to be 90% of overall capacity and this cannot be exceeded. The importance of not exceeding this hard limit is on the contrary defined for a situation, where the task distribution would be continual through the whole simulation duration, meaning that the resources would be fully utilized, leading the nodes to crash because of no spare capacity [53]. The third reason is the actual color constraint, restraining some tasks to be assigned to particular nodes even if the node has enough capacity.

Based on the discussed results, we can conclude that the EAODA and PEAODA are able to effectively utilize the underlying infrastructure resources while still aiming to prioritize tasks based on their priorities.

6.1.2 Task allocation evaluation

Since the primary goal of EAODA is to balance the load among nodes while prioritizing high priority tasks, the most important performance indicator is the total amount of allocated tasks with highest possible priorities. As we already presented in Section 6.1.1, EAODA and PEAODA showed the capability to efficiently utilize the underlying resource. In this Section we will demonstrate the priority allocation capabilities of EAODA and PEAODA and how they compare to other algorithms. Nevertheless, another key factor is the distribution of rescheduled and offloaded tasks, as it causes interruptions to the task execution. As of now, the priority represented by 5 is considered highest and 1 lowest.

The aggregated allocation ratios shown in Figure 6.2 acknowledge that EAODA and PEAODA perform significantly better regarding the prioritized workload allocation. Both methods are outperforming FIFO and Greedy meaningfully, averaging the higher allocation ratio by 48%.

To summarize, we discussed the performance of EAODA and PEAODA methods compared to FIFO and Greedy algorithms. Based on the overall objective of maximizing the overall resource utilization while prioritizing important tasks we observed that EAODA and PEAODA significantly outperform both, FIFO and Greedy algorithms, which is underpinned by almost 50% higher allocation ratio success for high priority tasks. Further, we discovered that EAODA is strongly outperforming Greedy algorithm for high priority tasks, which is caused be EAODA's

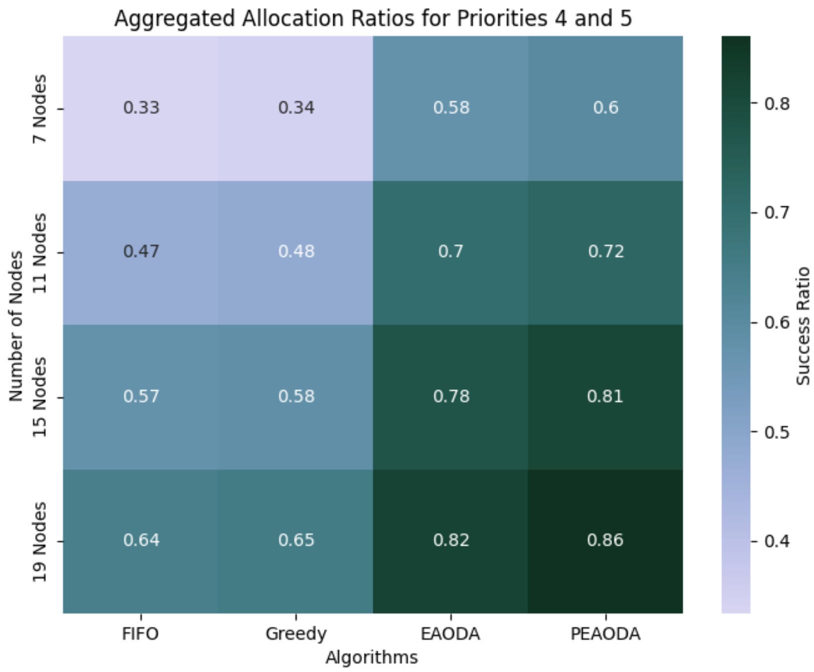

Figure 6.2: Aggregated Allocation Ratios for Priorities 4 and 5

ability to free up space and reschedule or offload some tasks in order to favor the high priority ones. We also determined that EAODA and PEAODA perform best in scenarios, with very limited resources capacity and huge load to be processed. We also identified, that PEAODA yields better allocation ratios for scenarios with huge amount of tasks to be processed, such as BeST.

7. Conclusion

In recent years, the number of connected cars has increasing rapidly. Such a network of connected vehicles can be referred to as Internet of Vehicles. To support different requirements of such a network and applications within, there is a demand for effective and reliable task scheduling orchestration, providing a computation platform for the needs of IoV.

There already exist several solutions aiming to improve the process of task offloading in IoV environment. Many of the existing algorithms are only enhancing resource utilization of the computational resource. Hence, new solutions to challenge this problem are appearing. They are based on network utilization, mobility support or load-balancing.

In our work, we analyzed several competent approaches for task offloading in V2X communication. Based on the analysis, we formulated new objectives for our novel approach of task offloading in the IoV environment and defined different challenges in the problem-solving process. According to the formulated objectives, we briefly discussed the nature of the optimization problem to solve. In accordance to the inherent characteristics, we decided to formulate the optimization problem as a Knapsack Problem. Since Knapsack problem is NP-hard, we need to adjust the algorithm design accordingly. Hence the different constraints included in this problem, such as resource capacity and adhering to time-sensitive computation requirements coming from the V2X environment, we successfully designed **EAODA** method and formally verified some of it's characteristics.

Based on the design, we've implemented the final algorithm, although we're not able to test in real environment due to the limitations we discovered. However, we simulated the real environment and tested the algorithm using real-world traffic scenarios in different cities. We've proven how adopting the consensus of higher priorities take precedence in the context of task offloading in IoV networks can increase the overall user experience and road safety.

While solving the given goals, we've successfully designed a task offloading method by modifying Knapsack problem to consider task priority and color affinity while still maintaining the highest possible resource usage.

The **EAODA** method is designed to prioritize high-priority tasks to be executed on the edge layer, closer to the end users, while utilizing the underlying infrastructure based on the given thresholds. The method processes incoming tasks at their time of arrival and places them on nodes according to their current availability. It is very important to clarify, that we consider edge layer to be hard limited by the number of resources present, thus our goal is to maximize the utilization on this layer. Since we cannot apply other infrastructure optimization tools like autoscaling, the primary goal of **EAODA** is to keep the high priority tasks on this layer by all means.

We verified that **EAODA** is performing as designed and thoroughly tested it's application on simulated scenarios. We then evaluated that EAODA and PEAODA enhance the prioritized task allocation by almost 50% better compared to conventional algorithms for high-priority tasks.

We assume that in the overall task placement process, **EAODA** is performing worse than Greedy, but our primary focus is only on high priority ones. Furthermore, it can be concluded that as the available space on the edge layer increases, or as the number of tasks to be processed decreases, the applicability of the **EAODA** method is becoming less relevant since sufficient space negates the necessity for task prioritization.

For future work, we pointed out several modifications and improvements to be implemented in order to better serve the purpose of optimal task offloading in IoV networks. The application of machine learning techniques in the task offloading decision-making process would improve overall performance, particularly with enhanced resource utilization. Enhancements would undoubtedly arise from the incorporation of other environmental elements and features, such as network delay, jitter, or vehicular motion. It is essential to accurately simulate job priorities to reflect the actual traffic simulation and other conditions that affect the QoE of participants in the IoV environment.

8. Publications

This chapter summarizes a list of my publications to date.

Scientific work published in international journals registered in Web of Science or SCOPUS databases

1. Ján Nemčík, Lukáš Šoltés, Galinski Marek and Kotuliak Ivan. "Edge to Cloud Task Offloading Optimization in Internet of Vehicles Networks". Strojnícky časopis - Journal of Mechanical Engineering Slovak University of Technology, 75, no. 1 (2025): 123-134. doi: 10.2478/scjme-2025-0013
2. Marek Galinski, Dominik Šalgovič, Ján Nemčík et al. "Empirical analysis of end-to-end QoS performance in urban cellular vehicular scenarios". e+i Elektrotechnik und Informationstechnik (2025). doi: 10.1007/s00502-025-01324-2

Published papers at international scientific conferences

1. J. Nemčik, P. Kănuch and I. Kotuliak, "Content Distribution in Private Networks," 2022 International Symposium ELMAR, Zadar, Croatia, 2022, pp. 67-70, doi: 10.1109/ELMAR55880.2022.9899785.

   Cited in:

   - Wang, Zhongren, Yuyang Zhang, Ping Dong, and Xiaoya Zhang. "Design and Implementation of Abnormal Diagnosis Mechanism in Private Network." In 2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), vol. 6, pp. 276-281. IEEE, 2024.
   - Pardosi, Victor Benny Alexsius, Sutariyani Sutariyani, Muhammad Ikhsanudin, and Abdurrahman Naufal. "Addressing DNS Propagation

Challenges with Repurposed STBs, ZeroTier Networking, and Indonesian ISP Integration." Journal of Intelligent Systems and Information Technology 1, no. 2 (2024): 85-93.

2. Nemčík, Ján, Edvin Mako, and Tibor Krajčovič. "Smart indoor greenhouse." In 7th Conference on the Engineering of Computer Based Systems, pp. 1-2. 2021. doi: 10.1145/3459960.3461558

   Cited in:

   - Hosny, Khalid M., Walaa M. El-Hady, and Farid M. Samy. "Technologies, Protocols, and applications of Internet of Things in greenhouse Farming: A survey of recent advances." Information Processing in Agriculture 12, no. 1 (2025): 91-111.
   - Farooq, Muhammad Shoaib, Rizwan Javid, Shamyla Riaz, and Zabihullah Atal. "IoT based smart greenhouse framework and control strategies for sustainable agriculture." IEEE Access 10 (2022): 99394-99420.
   - Hosny, Khalid M., Walaa M. El-Hadya, and Farid M. Samy. "Information Processing in Agriculture."
   - Wicaksana, Yudistira Aria, and Hastha Sunardi. "RANCANG BANGUN SMART SYSTEM RUANG GREENHOUSE BERBASIS IOT DENGAN MENGGUNAKAN NODEMCU ESP8266." Journal of Intelligent Networks and IoT Global 2, no. 2 (2024): 77-84.

Published papers at domestic scientific conferences

1. Vanek, Samuel, Peter Kaňuch, and Ján Nemčík. "Statistical Metadata Analysis of Mobile Phone Communications." In 2021 19th International Conference on Emerging eLearning Technologies and Applications (ICETA), pp. 414-419. IEEE, 2021.

Bibliography

[1] Seyed Mohsen Hosseinian, Hamid Mirzahossein, and Robert Guzik. "Sustainable Integration of Autonomous Vehicles into Road Networks: Ecological and Passenger Comfort Considerations". In: *Sustainability* 16.14 (July 2024), p. 6239. DOI: 10.3390/su16146239. URL: https://doi.org/10.3390/su16146239.

[2] Anshuman Sharma and Zuduo Zheng. "Connected and Automated Vehicles: Opportunities and Challenges for Transportation Systems, Smart Cities, and Societies". In: *Advances in 21st century human settlements*. Jan. 2021, pp. 273–296. DOI: 10.1007/978-981-15-8670-5_11. URL: https://doi.org/10.1007/978-981-15-8670-5_11.

[3] Riccardo Trivisonno, Qing Wei, and Clarissa Cassales Marquezan. "QoS Enhancements for V2X Services in 5G Networks". In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 2020, pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9129065.

[4] Qi Wu et al. "Tasks Offloading for Connected Autonomous Vehicles in Edge Computing". In: *Mobile Networks and Applications* 27.6 (Aug. 2021), pp. 2295–2304. DOI: 10.1007/s11036-021-01794-6. URL: https://doi.org/10.1007/s11036-021-01794-6.

[5] Anushka Biswas and Hwang-Cheng Wang. "Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain". In: *Sensors* 23.4 (Feb. 2023), p. 1963. DOI: 10.3390/s23041963. URL: https://doi.org/10.3390/s23041963.

[6] Junhua Wang, Kun Zhu, and Ekram Hossain. "Green Internet of Vehicles (IoV) in the 6G Era: Toward Sustainable Vehicular Communications and Networking". In: *IEEE Transactions on Green Communications and Networking* 6.1 (2022), pp. 391–423. DOI: 10.1109/TGCN.2021.3127923.

[7] Luca Milani, Detlev Mohr, and Nicola Sandri. *Built to last: Making sustainability a priority in transport infrastructure*. en. Oct. 2021. URL: https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/built-to-last-making-sustainability-a-priority-in-transport-infrastructure.

[8] Juan Contreras-Castillo, Sherali Zeadally, and Juan Antonio Guerrero-Ibañez. "Internet of Vehicles: Architecture, Protocols, and Security". In: *IEEE Internet of Things Journal* 5.5 (2018), pp. 3701–3709. DOI: 10.1109/JIOT.2017.2690902.

[9] Zoran Constantinescu and Monica Vladoiu. "Towards Vehicular Fog Computing: an Architecture for Connected Vehicles and Vehicular Clouds". In: *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. 2020, pp. 1–6. DOI: 10.1109/RoEduNet51892.2020.9324868.

[10] Wei Duan et al. "Emerging Technologies for 5G-IoV Networks: Applications, Trends and Opportunities". In: *IEEE Network* 34.5 (2020), pp. 283–289. DOI: 10.1109/MNET.001.1900659.

[11] Abdus Samad et al. "Internet of vehicles (IoV) requirements, attacks and countermeasures". In: *Proceedings of 12th INDIACom; INDIACom-2018; 5th international conference on "computing for sustainable global development" IEEE conference, New Delhi*. 2018.

[12] Ishita Seth et al. "A Taxonomy and Analysis on Internet of Vehicles: Architectures, Protocols, and Challenges". In: *Wireless Communications and Mobile Computing* 2022 (May 2022). Ed. by Rashid A Saeed, pp. 1–26. DOI: 10.1155/2022/9232784. URL: http://dx.doi.org/10.1155/2022/9232784.

[13] Juan Contreras Castillo, Sherali Zeadally, and Juan Guerrero-Ibañez. "A seven-layered model architecture for Internet of Vehicles". In: *Journal of Information and Telecommunication* 1 (Jan. 2017), pp. 4–22. DOI: 10.1080/24751839.2017.1295601.

[14] Jian Wang et al. "A Survey of Vehicle to Everything (V2X) Testing". In: *Sensors* 19.2 (2019). ISSN: 1424-8220. DOI: 10.3390/s19020334. URL: https://www.mdpi.com/1424-8220/19/2/334.

[15] Jianhua He, Kun Yang, and Hsiao-Hwa Chen. "6G Cellular Networks and Connected Autonomous Vehicles". In: *IEEE Network* 35.4 (2021), pp. 255–261. DOI: 10.1109/MNET.011.2000541.

[16] Omprakash Kaiwartya et al. "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects". In: *IEEE access* 4 (2016), pp. 5356–5373.

[17] Yijia Feng et al. "The Overview of Chinese Cooperative Intelligent Transportation System Vehicular Communication Application Layer Specification and Data Exchange Standard". In: *Information Technology and Intelligent Transportation Systems*. IOS Press, 2017, pp. 516–526.

[18] 3GPP. *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on LTE Support for Vehicle to Everything (V2X) Services (Release 14)*. 2015.

[19] Min Chen et al. "Cognitive Internet of Vehicles". In: *Computer Communications* 120 (May 2018), pp. 58–70. DOI: 10.1016/j.comcom.2018.02.006. URL: http://dx.doi.org/10.1016/j.comcom.2018.02.006.

[20] Kashif Naseer Qureshi et al. "Internet of vehicles: Key technologies, network model, solutions and challenges with future aspects". In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1777–1786.

[21] Baofeng Ji et al. "Survey on the internet of vehicles: Network architectures and applications". In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 34–41.

[22] Fangchun Yang et al. "An overview of Internet of Vehicles". In: *China Communications* 11.10 (2014), pp. 1–15. DOI: 10.1109/CC.2014.6969789.

[23] Haibo Zhou et al. "Evolutionary V2X Technologies Toward the Internet of Vehicles: Challenges and Opportunities". In: *Proceedings of the IEEE* 108.2 (2020), pp. 308–323. DOI: 10.1109/JPROC.2019.2961937.

[24] B. Padmaja et al. "Exploration of issues, challenges and latest developments in autonomous cars". In: *Journal of Big Data* 10.1 (May 2023). DOI: 10.1186/s40537-023-00701-y. URL: http://dx.doi.org/10.1186/s40537-023-00701-y.

[25] Jiamin Cheng et al. "Research on User Experience Quality Evaluation Method of Internet of Vehicles Based on sEMG Signal". In: *Simulation Tools and Techniques* (2021), pp. 694–703. DOI: 10.1007/978-3-030-72792-5_54. URL: http://dx.doi.org/10.1007/978-3-030-72792-5_54.

[26] Md. Noor-A-Rahim et al. "6G for Vehicle-to-Everything (V2X) Communications: Enabling Technologies, Challenges, and Opportunities". In: *Proceedings of the IEEE* 110.6 (2022), pp. 712–734. DOI: 10.1109/JPROC.2022.3173031.

[27] Sakshi Garg et al. “Accessible review of internet of vehicle models for intelligent transportation and research gaps for potential future directions”. In: *Peer-to-Peer Networking and Applications* 14.2 (Jan. 2021), pp. 978–1005. DOI: 10.1007/s12083-020-01054-6. URL: http://dx.doi.org/10.1007/s12083-020-01054-6.

[28] Issam W. Damaj et al. “Connected and Autonomous Electric Vehicles: Quality of Experience survey and taxonomy”. In: *Vehicular Communications* 28 (Apr. 2021), p. 100312. DOI: 10.1016/j.vehcom.2020.100312. URL: http://dx.doi.org/10.1016/j.vehcom.2020.100312.

[29] Dimitar Minovski, Christer Åhlund, and Karan Mitra. “Modeling Quality of IoT Experience in Autonomous Vehicles”. In: *IEEE Internet of Things Journal* 7.5 (2020), pp. 3833–3849. DOI: 10.1109/JIOT.2020.2975418.

[30] Shaik Ashfaq Hussain et al. “A Review of Quality of Service Issues in Internet of Vehicles (IoV)”. In: *2019 Amity International Conference on Artificial Intelligence (AICAI)*. 2019, pp. 380–383. DOI: 10.1109/AICAI.2019.8701299.

[31] Elmustafa Sayed Ali et al. “Machine Learning Technologies for Secure Vehicular Communication in Internet of Vehicles: Recent Advances and Applications”. In: *Security and Communication Networks* 2021 (Mar. 2021). Ed. by Fawad Ahmed, pp. 1–23. DOI: 10.1155/2021/8868355. URL: http://dx.doi.org/10.1155/2021/8868355.

[32] Ansif Arooj et al. “Big Data Processing and Analysis in Internet of Vehicles: Architecture, Taxonomy, and Open Research Challenges”. In: *Archives of Computational Methods in Engineering* 29.2 (May 2021), pp. 793–829. DOI: 10.1007/s11831-021-09590-x. URL: http://dx.doi.org/10.1007/s11831-021-09590-x.

[33] Haruna Chiroma et al. “Deep Learning-Based Big Data Analytics for Internet of Vehicles: Taxonomy, Challenges, and Research Directions”. In: *Mathematical Problems in Engineering* 2021 (Nov. 2021). Ed. by Mohammad Yaghoub Abdollahzadeh Jamalabadi, pp. 1–20. DOI: 10.1155/2021/9022558. URL: http://dx.doi.org/10.1155/2021/9022558.

[34] Abdallah H. Salem, Issam W. Damaj, and Hussein T. Mouftah. “Vehicle as a Computational Resource: Optimizing Quality of Experience for connected vehicles in a smart city”. In: *Vehicular Communications* 33 (Jan. 2022), p. 100432. DOI: 10.1016/j.vehcom.2021.100432. URL: http://dx.doi.org/10.1016/j.vehcom.2021.100432.

[35] Guilu Wu, Zhengquan Li, and Huilin Jiang. “Quality of experience-driven resource allocation in vehicular cloud long-term evolution networks”. In: *Trans-*

actions on Emerging Telecommunications Technologies 31.8 (July 2020). DOI: 10.1002/ett.4036. URL: http://dx.doi.org/10.1002/ett.4036.

[36] Tobias Kain et al. "C-PO: A Context-Based Application-Placement Optimization for Autonomous Vehicles". In: *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2021, pp. 1288–1293. DOI: 10.23919/DATE51398.2021.9473948.

[37] Tobias Kain et al. "Optimizing the Placement of Applications in Autonomous Vehicles". In: *Proceedings of the 30th European Safety and Reliability Conference (ESREL 2020)*. 2020.

[38] Muhammad Awais Javed and Sherali Zeadally. "AI-Empowered Content Caching in Vehicular Edge Computing: Opportunities and Challenges". In: *IEEE Network* 35.3 (2021), pp. 109–115. DOI: 10.1109/MNET.011.2000561.

[39] Shuo Xiao et al. "Research on a Task Offloading Strategy for the Internet of Vehicles Based on Reinforcement Learning". In: *Sensors* 21.18 (Sept. 2021), p. 6058. DOI: 10.3390/s21186058. URL: http://dx.doi.org/10.3390/s21186058.

[40] JiuJun Cheng et al. "Routing in Internet of Vehicles: A Review". In: *IEEE Transactions on Intelligent Transportation Systems* 16.5 (2015), pp. 2339–2352. DOI: 10.1109/TITS.2015.2423667.

[41] Philipp Meyer et al. "Demo: A Security Infrastructure for Vehicular Information Using SDN, Intrusion Detection, and a Defense Center in the Cloud". In: *2020 IEEE Vehicular Networking Conference (VNC)*. 2020, pp. 1–2. DOI: 10.1109/VNC51378.2020.9318351.

[42] Hiroya Matsumoto, Bo Gu, and Osamu Mizuno. "A V2X Task Offloading Method Considering Automobiles' Behavior in Urban Area". In: *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 2019, pp. 1–4. DOI: 10.23919/APNOMS.2019.8892838.

[43] Firdose Saeik et al. "Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions". In: *Computer Networks* 195 (2021), p. 108177. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2021.108177. URL: https://www.sciencedirect.com/science/article/pii/S1389128621002322.

[44] Chen Chen et al. "A Multihop Task Offloading Decision Model in MEC-Enabled Internet of Vehicles". In: *IEEE Internet of Things Journal* 10.4 (2023), pp. 3215–3230. DOI: 10.1109/JIOT.2022.3143529.

[45] Salman Raza et al. "Task Offloading and Resource Allocation for IoV Using 5G NR-V2X Communication". In: *IEEE Internet of Things Journal* 9.13 (2022), pp. 10397–10410. DOI: 10.1109/JIOT.2021.3121796.

[46] European Telecommunications Standards Institute. *5G; Service requirements for enhanced V2X scenarios (3GPP TS 22.186 version 17.0.0 Release 17)*. ETSI Standard RTS/TSGS-0122186vh00. Available: https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=65210. Apr. 2022.

[47] Stuart Mitchell, Michael OSullivan, and Iain Dunning. "PuLP: a linear programming toolkit for python". In: *The University of Auckland, Auckland, New Zealand* 65 (2011).

[48] Karl Schrab et al. "Modeling an ITS Management Solution for Mixed Highway Traffic With Eclipse MOSAIC". In: *IEEE Transactions on Intelligent Transportation Systems* 24.6 (2023), pp. 6575–6585. DOI: 10.1109/TITS.2022.3204174.

[49] Pablo Alvarez Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL: https://elib.dlr.de/124092/.

[50] Lara Codeca et al. "Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation". In: *IEEE Intelligent Transportation Systems Magazine* 9.2 (2017), pp. 52–63. DOI: 10.1109/MITS.2017.2666585.

[51] Karl Schrab, Robert Protzmann, and Ilja Radusch. *A Large-Scale Traffic Scenario of Berlin for Evaluating Smart Mobility Applications*. Jan. 2023. DOI: 10.1007/978-3-031-23721-8_24. URL: https://doi.org/10.1007/978-3-031-23721-8_24.

[52] Jian Li et al. *Improved FIFO Scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing*. Feb. 2017. DOI: 10.3390/info8010025. URL: https://doi.org/10.3390/info8010025.

[53] Haibo Zhang, Zixin Wang, and Kaijian Liu. "V2X offloading and resource allocation in SDN-assisted MEC-based vehicular networks". In: *China Communications* 17.5 (2020), pp. 266–283. DOI: 10.23919/JCC.2020.05.020.

A. List of Acronyms

BeST Berlin Sumo Traffic 26

CAV Connected Autonomous Vehicle 1

EAODA Edge-to-Cloud Offloading Decision Algorithm 17

FIFO First-In-First-Out 26

FIFS First-In-First-Serve 10

IoT Internet of Things 3

IoV Internet of Vehicles 1

LuST Luxembourg Sumo Traffic 26

MEC Mobile Edge Computing 8

MKCP Multiple Knapsack Problem with Color Constraints 17

OOM Out of Memory 20

PEAODA Prioritized Edge-to-Cloud Offloading Decision Algorithm . 27

QoE Quality of Experience 8

QoS Quality of Service 1

RAKP Resource Allocation Knapsack Problem 18

SDN Software Defined Network 3

V2C Vehicle-to-Cloud 1

V2I Vehicle-to-Infrastructure 1

V2N Vehicle-to-Network 6

V2P Vehicle-to-Pedestrian 6

V2V Vehicle-to-Vehicle 6

V2X Vehicle-to-Everything 1

VEC Vehicular Edge Computing 10