UNIVERSITY OF LEEDS

Optimised Vehicular Networks With Distributed Processing

Fatemah Sadeq Behbehani

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

The University of Leeds
School of Electronic and Electrical Engineering

December 2020

Intellectual Property Statement

The candidate confirms that the work submitted is his/her own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

The work in Chapter 4 has appeared or will partially appear in publications as follows:

F. S. Behbehani, M. Musa, T. Elgorashi and J. M. H. Elmirghani, "Energy Efficient Distributed Processing in Vehicular Cloud Architecture," *2019 21st International Conference on Transparent Optical Networks (ICTON), Angers, France, 2019, pp. 1-4.*

The ideas for the paper were conceived by Prof. Jaafar Elmirghani. The candidate developed the model for energy efficient vehicular cloud using mixed integer linear programming, produced the results, and wrote the paper. Dr. M. Musa revised the model and results, and Dr. El-Gorashi and Prof. Elmirghani revised the paper and approved the results.

The work in Chapter 5 has appeared or will partially appear in publications as follows:

F. S. Behbehani, M. Musa, T. Elgorashi and J. M. H. Elmirghani, “Optimized Distributed Processing in a Vehicular Cloud Architecture”. *2020 22nd International Conference on Transparent Optical Networks (ICTON), Bari, Italy, 2020.*

Prof. Jaafar Elmirghani suggested the consideration of delay. The candidate extended the previously publish model to include energy and delay minimisation in vehicular clouds, produced the results, and wrote the paper. Dr. M. Musa revised the model, Dr. El-Gorashi revised the results and the paper, and Prof. Elmirghani revised the paper and approved the results.

Dedication

This work is dedicated to my **beloved father** (may he rest in peace). I wish you were here; I know you would have been the proudest.

Acknowledgements

First, I would like to express my thanks to my supervisor, Prof. Jaafar Elmirghani for his insights, and guidance throughout this experience. Working under his supervision has been a privilege and enlightening experience.

I would also like to thank my co-supervisor Dr Taisir Elgorashi, for her valuable efforts and continued support.

I am also grateful to Dr Mohammed Musa and all my colleagues and friends in the Institution of Communication and Power Networks (ICaPNet) for all the informative and helpful discussions we had, and the emotional and moral support they provided.

I would like to express my gratitude and my love to my mother, Narjes, for every moment she was there for my family. I deeply thank her and my dear siblings for their unshakable support and believe in me.

Also, to my darling kids Maria, Zainah, and Ahmed, I express my greatest love and thanks for their understanding and for putting up with my busy days and frustrations these past years.

Finally, my deepest gratitude and love to my husband, Abdulaziz. His rock-solid support through the years has motivated me every step of the way in this journey. None of this would have been possible without him by my side, never failing to express his love, understanding and faith.

Abstract

The introduction of cloud computing presented new solutions for the storage and processing of data, to support the limited capabilities of devices of end user. Cloud data centres are geographically distant from the users, which increases latency and networking power consumption, as well as increasingly exhausting the networking and processing capabilities of the cloud. Alternatives are introduced in the form of minimised and distributed data centres closer to end users, which introduced the paradigms of edge processing and edge computing. On the same line of research, modern vehicles are fully equipped with substantial processing capabilities, often un-used, in their on-board-units (OBUs), which led to the introduction of the vehicular cloud paradigm. This thesis addresses the problem of distributed processing allocation in the context of vehicular networks. An end-to-end vehicular cloud architecture is developed, in which vehicles are clustered in vehicular clouds willing to share their computational resources. The vehicular cloud is supplemented by edge processing nodes and central cloud to maintain resources availability. A Mixed Integer Linear Programme is developed to optimally allocate user demands to resource in three available processing layers, vehicular cloud, edge, and conventional cloud layer, while minimising processing-induced and network-induced power consumption. In comparison to conventional clouds, allocation in vehicular clouds can achieve 84% power saving. This is, however, subject to several factors, which are considered in the thesis, such as demand size, number of processing demand splits allowed, and the handling of networking capacity. In addition, the thesis studies the allocation

of the processing demands in distributed processing layers while minimising end-to-end delay and minimising energy, which is an important factor for many latencies sensitive applications. Another issue tackled in this thesis is the resilience of the vehicular cloud architecture presented. Minimisation of power usually comes at the expense of the resilience of a system where the former calls for the utilisation of the minimum number of resources possible, while the latter calls for redundancy to improve resilience. This thesis introduces mechanisms for resilient vehicular cloud architectures and investigates how the power minimisation might be compromised by the introduction of such measures.

Table of Contents

List of Tables

List of Figures

Abbreviations

APR	Active Processing Resilience
APs	Access Points
CaaS	Computation as a Service
CC	Cloud Computing
CCTV	Closed-Circuit Television
DC	Data Centres
DSRC	Dedicated Short-Range Communication
EC	Edge Computing
EED	End to End Delay
FC	Fog Computing
Gbps	Giga bit per second
GHz	Giga Hertz
IaaS	Infrastructure as a Service
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IoV	Internet-of-Vehicles
IP	Internet Protocol
IPR	Idle Processing Resilience
IT	Information Technology

ITS	Intelligent Transportation Systems
kb	kilobits
M2M	Machine to Machine
Mbps	Mega bit per second
MC	Mist Computing
MCC	Mobile Cloud Computing
MEC	Multi-Access Edge Computing
MHz	Mega Hertz
MI	Million Instructions
MILP	Mixed Integer Linear Programme
MIPS	Million Instructions Per Second
NS	Network Slicing
NaaS	Networking as a Service
NFV	Network Function Virtualisation
OBU	On Board Unit
OLT	Optical Line Termination
ONUs	Optical Network Unit
PaaS	Platform as a Service
PON	Passive Optical Network
PUE	Power Usage Effectiveness
QoS	Quality of Service
RAM	Random Access Memory

RSUs	Roadside Units
SaaS	Software as a Service
SnaaS	Sensing as a Service
StaaS	Storage as a Service
SDN	Software-Defined Network
SLA	Service-Level Agreement
ToR	Top of Rack
V2E	Vehicle to Edge communication
V2I	Vehicle-to-Infrastructure communication
V2V	Vehicle-to-Vehicle communication
V2X	Vehicles-to-everything
VANET	Vehicular Ad Hoc Network
V	Vehicle Only Scenario
VC	Vehicular Cloud
VE	Vehicle-Edge Scenario
VEC	Vehicle-Edge-Cloud Scenario
VMM	Vehicular Virtual Machine
VMs	Virtual Machines
VNs	Vehicular Networks
VSDN	Vehicular Software Defined Network
VVMM	Vehicular Virtual Machine Migration
WAVE	Wireless Access for Vehicular Environment

Chapter 1
Introduction

Cloud computing (CC) has introduced new possibilities for data processing and storage. The paradigm provides remote services that relieve the end users from handling them in their own devices [1]. It reduces the cost for users by eliminating the need to deploy and maintain hardware and software resources. On the providers side, the costs of provisioning cloud services are expected to be well compensated through the profit of the growing cloud services. The demand for cloud services is growing exponentially with 30-40% annual traffic growth [2]. This growth has created concerns regarding the monetary cost and energy consumption of networks and data centres. In addition, large data centres tend to be located away from the end users, which increases the latency and power consumption of the networks interconnecting users to the cloud. Also some applications such as smart city applications produce high volumes of data that have only local relevance making storing or processing them remotely in the cloud an unnecessary burden on the network and cloud resources [3].

The increasing demand has encouraged the search for alternative remote resource sharing paradigms of lower energy consumption and latency. Edge and fog computing (EC/FC) were introduced towards this end. In fog and edge computing, processing is done in small, distributed data centres located at the edge of the network close to the end users. Benefitting of the pervasiveness of smart devices in almost every part of modern age, the mini data centres are

envisioned to be built up from any devices with computational capabilities to reduce the deployment and maintenance cost of such distributed architectures.

Facilitating the revolution for smarter cities, vehicles are now equipped to go beyond transportation functionalities. On-Board Units (OBUs) are efficient computers deployed in vehicles to serve safety and non-safety-based applications. Research on vehicular networks (VNs) shows that vehicular resources are underutilised [4, 5] creating potential to compose distributed architectures, referred to as vehicular clouds [6], to augment the network edge processing capabilities used in distributed processing. Vehicular clouds (VC) are mainly used to provide Software as a Service (SaaS) and Infrastructure as a Service (IaaS), which can be further decomposed into Computation as a Service, Storage as a Service (StaaS), and Networking as a Service (NaaS) [7, 8].

The technologies required to implement vehicular clouds are already available or are expected to mature in the near future. Virtualisation [9, 10] supports the vision of vehicular clouds by creating virtual, abstract, copies of the resources from different vehicles. This way, a vehicular cloud can be created regardless of the differences in the vehicles' computing resources. Vehicular networks also benefit from heterogeneous communication technologies. Most of the modern cars are now connected to the Internet through cellular networks or WiFi. Also, the IEEE 802.11p standard and the Wireless Access for Vehicular Environment (WAVE) standard derived from IEEE 802.11 and IEEE 1609, respectively, are used specifically for intelligent transportation systems (ITS) [11].

In literature, research elaborated on the uses of the VC paradigm for energy efficiency or improved latency. However, most efforts were dedicated to

improvements in the routing algorithms and resources allocation strategies for the VC paradigm, without numerically comparing the results to the CC. To the best of our knowledge, our work provides the first that quantitatively evaluated model for the values of energy and delay when using the distributed VC and edge node against the use of the conventional cloud as baseline.

In this thesis, we present a vehicular cloud architecture in which clusters of vehicles come together to form a mini cloud providing computational resources to end users. The vehicular cloud is supplemented by fixed edge nodes and the conventional cloud to maintain service availability. The evaluation of the vehicular cloud architecture in this work focuses on energy efficiency and delay performance. The dynamicity of vehicles is a challenging issue that can affect their availability and therefore the resilience of services offered by the vehicular cloud. Therefore, we investigate improving the resilience of the architecture by backing up demands served (processed) in vehicular nodes.

1.1 Research Objectives

The work reported in this thesis has the following objectives:

1. To propose a vehicular cloud architecture supported by fixed edge nodes and the conventional cloud.
2. To study the feasibility and merits of resource allocation in the different processing layers (vehicular cloud, edge, conventional cloud) to provide Computation as a Service (CaaS).
3. To develop an energy efficient resource allocation approach in the vehicular cloud architecture through power minimisation.

4. To study the delay performance of energy efficient processing allocation in the vehicular cloud.
5. To study the impact of minimising end-to-end delay on the energy efficiency and achieve a trade-off between energy efficiency and delay performance through joint minimisation of energy consumption and end-to-end delay.
6. To study the impact of resilience measures on the power consumption of the vehicular cloud architecture.

1.2 Thesis Contributions

The thesis contributions are summarised as follows:

1. A vehicular cloud architecture supported by fixed edge nodes and the conventional cloud is proposed.
2. A mixed integer linear programming (MILP) model is developed to optimally allocate processing demands in the three layers of the proposed architecture with the objective of minimising the power consumption induced by processing and networking.
3. The energy efficiency of the proposed architecture is evaluated considering different test cases to study varying number of demands, varying demand sizes and the impact of processing demand splitting.
4. A heuristic is developed to allocate processing demands in real time and its performance is compared to the optimal performance established by the MILP model.

5. A MILP model is developed to jointly minimise the energy consumption and end-to-end delay of the vehicular cloud architecture considering on-demand smart city applications where users request a one-off demand.
6. A MILP model is developed to consider a vehicular cloud architecture of improved resilience. It tackles the problem of processing allocation in a way that minimises power consumption while still maintaining quality of service and completion of the job. Test cases that reflect different approaches for a resilient vehicular cloud architecture are studied.

1.3 Related Publications

F. S. Behbehani, M. Musa, T. ElGorashi and J. M. H. Elmirghani, "Energy Efficient Distributed Processing in Vehicular Cloud Architecture," *2019 21st IEEE International Conference on Transparent Optical Networks (ICTON)*, Angers, France, 2019, pp. 1-4.

F. S. Behbehani, M. Musa, T. ElGorashi and J. M. H. Elmirghani, “Optimized Distributed Processing in a Vehicular Cloud Architecture”. *2020 22nd IEEE International Conference on Transparent Optical Networks (ICTON)*, Bari, Italy, 2020.

F. S. M. Behbehani, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Power minimisation in Vehicular Cloud Architecture”, to be submitted to IEEE Access.

F. S. M. Behbehani, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Energy and Delay Minimisation in Vehicular Cloud Architecture”, to be submitted to IEEE Access.

1.4 Thesis Outline

The following chapters in the thesis are organised as follows:

Chapter 2 reviews the literature on vehicular networks. It presents design fundamentals with an emphasis on the evolving paradigm of vehicular cloud. The chapter also tackles the challenges and the issues affecting the deployment of vehicular networks. A section of the chapter introduces enabling technologies such as virtualisation and Software-Defined Networks (SDN), which plays an essential role in supporting distributed processing environments. An overview of cloud computing was presented as issues with the cloud are the main motivation behind the work in this thesis. The chapter also covers fog/edge computing, which is viewed as a supporting paradigm that adds to the attractiveness of vehicular networking. In addition, mathematical programming and Optimisation algorithms are briefly reviewed as they are the main methodology used in the architecture optimisation and evaluations.

In Chapter 3**,** a vehicular cloud architecture supplemented by fog and cloud layers is proposed. The detailed description of the architecture layers, communication interfaces, and control is explained. The methodology used in evaluating the performance of the architecture is presented in the chapter. The Mixed inter linear programming (MILP) as main tool for evaluation is presented with description of the main input data, the energy and delay metrics, the considered applications, and evaluation scenarios.

In Chapter 4, a MILP model is developed to minimise the power consumption of processing demand allocation in the proposed vehicular architecture. A heuristic

is also developed for real-time allocation of processing demands, and to verify the MILP.

Chapter 5 presents a MILP model to jointly minimise the energy consumption and end-to-end delay of on-demand applications. The end-to-end delay is modelled including processing delay, transmission delay, propagation delay, and queueing delay.

In Chapter 6, the MILP model in Chapter 4 is extended to study a resilient vehicular cloud architecture where backup processing resources are allocated to demands primarily served in vehicular nodes. The increase in power consumption resulting from the improved resilience is evaluated.

Chapter 7 presents concluding remarks and sets future directions to develop research in vehicular clouds.

Chapter 2
Overview of Vehicular Networks

2.1 Introduction

In recent years, vehicle manufacturers focused on making the driving experience more comfortable for the driver. Devices with computational and communication capabilities are added to the vehicle design to serve many purposes, such as navigation, road services contacts, and entertainment. This gave rise to a new field in technology that studies the possibility of using these resources to form networks of vehicles. Due to the mobile nature of vehicles, these networks are formed with no specific structure in an ad hoc manner, hence, the term *vehicular ad hoc network* or VANETs emerged. VANETs mainly focus on enhancing road safety, collecting, and exchanging data between vehicles and roadside units [5].

A huge amount of data can be collected from vehicles to improve road conditions and to monitor congestion. A dedicated bandwidth has been allocated to the IEEE WAVE 802.11p standard for vehicles. Significant research efforts were dedicated to developing VANETs. However, there are still a lot of issues related to safety and privacy that need to be resolved. Also, researchers still argue that vehicular resources and allocated bandwidth are in most cases underutilised and the applications should be broadened to include much more than transportation and entertainment. New directions have been taken to connect VANETs to the Internet and cloud, introducing the concepts of Internet-of-Vehicles (IoV) [12] and Vehicular Clouds [13].

In this chapter we present an overview of vehicular networks. In Section 2.2 we explain the components and technologies used to build vehicular networks. We then focus on the vehicular cloud paradigm in Section 2.3 and explain the different models introduced in the literature and the fundamentals of vehicular cloud formation. In the same section, we review some of the services and applications explored in research on vehicular clouds and the challenges to be addressed. We give close attention to the challenging topic of vehicular virtual machine migration in Section 2.4. Section 2.5 discusses the concept of Software-Defined Networks which plays an essential role in supporting distributed processing environments such as vehicular clouds. Section 2.6 discusses topics related to the thesis and motivated our work such as cloud and fog computing, and we conclude the chapter in Section 2.7 with an overview of optimisation modelling and algorithms

2.2 Components

2.2.1 Objects

Three major objects with different roles exist in a vehicular network [14]:

1. **Vehicles**: Manufacturers are now competing to equip vehicles with computing facilities. OBUs are powerful computers installed in vehicles to provide computing, networking, and storage services. The use of the OBUs has started in VANETs to facilitate transportation and driving experience, but it quickly developed to provide other types of applications. In addition, vehicles are loaded with hundreds of sensors that can provide another layer of services in terms of data collection.

2. **Roadside Units:** As the vehicular network applications broadened and became more diverse, the complications of communication increased, and vehicles required easier ways for Vehicle-to-Infrastructure communication. Roadside Units (RSUs) provided a gateway for the vehicles to communicate with fixed infrastructure and base stations. Research efforts have explored the use of RSUs as an extra layer for services provisioning or as a controller for the vehicular network.
3. **Smart on Road Objects:** traffic lights, surveillance cameras, speed cameras, automated information and traffic signs, and environmental data collection sensors can be viewed as peripherals in the vehicular network that can be used when needed for data collection and storage.

2.2.2 Communication Interfaces

Vehicular networks need diverse wireless communication interfaces to facilitate communication among vehicles and communication between vehicles and RSUs and smart road objects. To enable this, vehicles need Vehicle to Infrastructure communication (V2I) interfaces, and Vehicle-to-Vehicle communication (V2V) interfaces. The choice of the communication technology depends on the data rate requirements. Dedicated Short Range Communications (DSRC) are usually used to provide quick transmission especially for emergency and transport awareness messages. It is derived from IEEE 802.11p and IEEE 1609 (WAVE) standards [15], which are amendments of the IEEE 802.11 standard, developed to suit the operating conditions in vehicular environments. Other wireless standards, such as WiFi, have been used for vehicular networks. In many applications more than

one communication standard is implemented [16]. The cellular 4G standards (LTE/WiMAX) and 5G standards are mostly used in vehicular networking due to their suitability for time-critical services and mobility issues [17].

Recently, the term vehicles-to-everything (V2X) has been coined to define communications between vehicles and ANY other object [18]. The term gained popularity with the emerging 5G technology. The introduction of 5G cellular communication technology opened the door for better service and connectivity of vehicular networks [19]. 5G promises improvement in bandwidth, higher data rates, and lower deployment costs than current cellular technologies. All these features are in high demand in vehicular networks, especially in relation to bandwidth-intensive applications and multimedia streaming services [20]. For example, passengers in a crowded urban city or a highway road can enjoy watching a movie without delay or intermittent service. Different transmission methods can be used in 5G architectures, such as Millimetre-Wave (mmWave) transmission, which can support short-range and high-speed applications [21] and Small Cell Basestations, which reduce power/cost with their smaller coverage area [22]. The promise of revolutionary advancement in 5G influenced greatly the work in vehicular networks, and some of the efforts in that area are included inTable 2-1. The work of [23] introduced cell-less network, where the vehicles communicate through moving access points on vehicles, controlled by software-defined cloudlets. The simulation results showed reduced latency and better V2V performance. In [24] the authors focused on the limitations of the current radio access communications in vehicular networks, then proceeded by

comparing beaconing in current LTE-V2I mode to the potential alternatives of LTE-V2V and LTE-V2V with (full-duplex). In [25] an SDN-based controller was designed with a priority manager and load balancer to improve the data offloading in vehicular network with 5G communication interface. Addressing security concerns in vehicular networks, [26] introduced a group-based communication approach for VANET in centralized and decentralized networks. They tackled the issues of secure dynamic network setup and secure mobility management and handover authentication. Work of [27] proposed novel 5G enabled vehicular network to deliver reliable real-time video reporting services that can be used in critical situations like traffic accidents. In [28], the authors discuss the issues inherited from video streaming in vehicular networks over 5G small cells and the disconnections caused by millimetre-waves propagation features. The work in [29] is one of the first to address the optimization of control plane events in the context of 5G with VANET. It studied the trade-off between latency requirements (mainly for infotainment applications) and the high cost of cellular communications.

Table 2-1: 5G implementations in Vehicular Networks

Scope	**references**
Heterogeneity/Handover	[23], [24]
Data Offloading	[25]
Security and privacy	[26], [27]
Infotainment	[28], [29]

2.3 Vehicular Cloud

The Mobile Edge Cloud (MEC) is a new field, where all moving devices with computing capabilities can interact for data collection, processing, or storage. The Vehicular Cloud (VC) is a subset of this emerging technology. It was first introduced by Olariu et al. in [30] who defined it as

> *"A group of largely autonomous vehicles whose corporate computing, sensing, communication, and physical resources can be coordinated and dynamically allocated to authorized users"*

VC [31] reduce the data sent to the clouds and can perform tasks in a more time effective manner. Also, with vehicular clouds some of the data are locally relevant and sending them over the Internet can be unnecessary cost. It will also be time consuming to search the conventional cloud for a specific information due to the huge amount of data it contains. Some of the benefits of VC are the reduced delay in sending and receiving data, reduced bandwidth cost as it is already available locally and is underutilised in many cases. Many more time sensitive applications can be supported, especially in medical and health monitoring fields or transportation fields. The conventional cloud is still needed for data of wider-relevance or data that needs to be stored for longer times or needs complex power-hungry computations [32].

2.3.1 Vehicular Cloud Models

Figure 2.1 shows the different ways vehicular clouds can be formed. The VC can be static, i.e. the vehicles are stationary [33], e.g. parked cars in a parking lot [34], or dynamically formed from moving vehicles [35, 36]. Also, a vehicular cloud can

have communication with infrastructure components such as RSUs, Access Points (APs), traffic lights, which can both act as controllers [37]. VCs can be infrastructure-less with one of the vehicles acting as a controller [23]. The different VC formations vary in their complexity and uses; however, they share the same main structure in their design and formation.

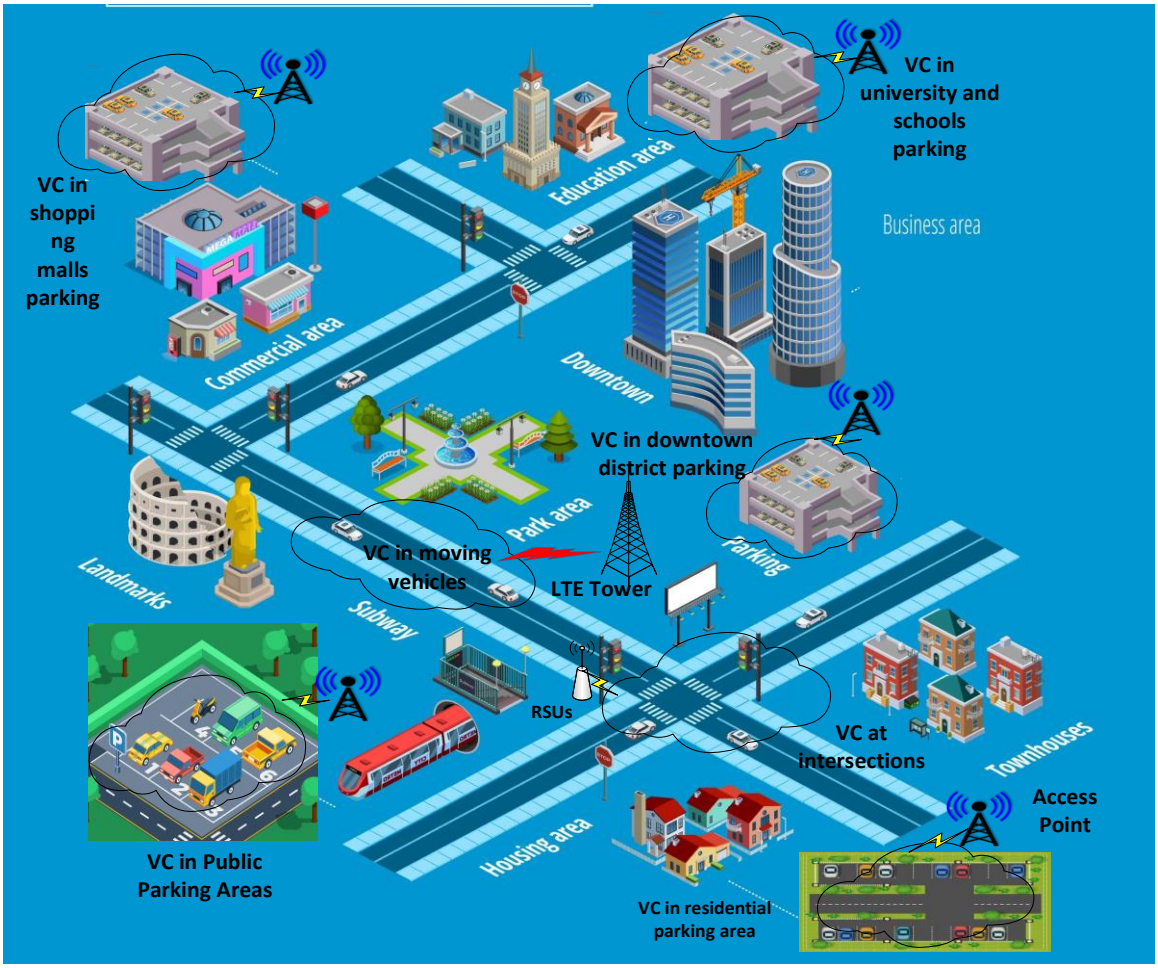

Figure 2.1: Vehicular Cloud Models

2.3.2 Design Fundamentals

There are many design principles that are unique to vehicular clouds and clearly distinguish this new paradigm from the conventional Internet cloud. One of the main distinctions is that vehicular clouds are not permanent, and they are formed as needed by interconnecting several vehicles and RSUs. The vehicular cloud formation is done in a way that achieves maximum benefit from the vehicle's capabilities. The formation of a vehicular cloud follows the following steps [6]:

1. **Candidate resources discovery:** A user with a task that requires the resources available in vehicles or in RSUs requests these resources to form a VC and awaits the response.
2. **Cloud assembling:** Some of the resources (vehicles and/or RSUs) responding to the request are clustered to form the VC. One essential step in the formation of the VC is the selection of the cluster head/coordinator, which can be an infrastructure component such as RSUs or one of the vehicles. The head coordinates and organises the vehicular cloud resources, demand distribution, load balancing between the resources. The selection of the VC head greatly affects the cluster stability, therefore, many algorithms are proposed to make an appropriate selection that has the required resources to make critical decisions in timely manner [38].
3. **Job assignment and results:** The VC head allocates resources to the request and the outcomes of the task will be sent to the user requesting the task. For some applications, the outcomes of the task can also be sent to the conventional cloud for storage.

4. **Cloud maintenance:** The availability of resources in VC is dynamic. When a resource leaves the cloud, the coordinator of the cloud is responsible for maintaining the data and allocating the task to a new resource to complete.
5. **Cloud disassembling:** When a task is completed, the head/coordinator of the cloud releases the resources and makes them free to connect to other VC, as needed.

2.3.3 Applications & Services

The services and applications of vehicles can be grouped into two main categories, IaaS, and SaaS [4, 39]. Infrastructure services can further be broken down into Networking as a Service (NaaS), Storage as a Service (StaaS), Computation as a Service (CaaS), and Sensing as a Service (SnaaS). The software services include safety, transportation, multimedia, and entertainment applications, provided on-demand to the end users. Deployments of the above-mentioned services has taken different forms in VC to achieve different goals.

2.3.3.1 Content Distribution and Load Balancing

In [40], the authors proposed a new approach for content downloading through vehicular networking with road side units to reduce the load on cellular networks. The authors in [41, 42] introduced green and load adaptive caching points for content distribution in vehicular environments.

2.3.3.2 Network Connectivity

In [43] a Parked Vehicle Assistance (PVA) architecture is introduced to benefit from the parked vehicles as static nodes to help the connectivity of the moving vehicles in VANET. The work in [23] introduced cell-less networking, where vehicles communicate through access points on vehicles controlled by software-defined cloudlets.

2.3.3.3 Energy Efficiency

The authors in [44, 45] studied the energy efficiency and QoS of four routing schemes in VC scenarios and investigated improving energy efficiency through reduction of number of basestations in city environment. The authors in [46] addressed the relationship between energy consumption and work offloading in vehicular edge computing environment.

2.3.3.4 Resources Provisioning

In [34], the authors envisioned the use of the resources available in vehicles parked in long term parking in airports as datacentre. Also, a two tier data centre architecture in a paking lot was introduced in [47], in which they studied the trade-off between the provisioning of storage resources and communication costs. A multi objective resource allocation model in vehicular clouds was proposed in [48]. The work in [49, 50] explored Sensors as a Service in vehicular networks where the mobility of vehicles broadens the sensors coverage area.

2.3.3.5 On-road Applications

Developments in the VC were accompanied by an explosive trend to develop applications benefiting of all VC can provide. Several applications can be directly

related to Intelligent Transport Systems (ITS) to improve the driving experience to road users, such as congestion avoidance applications, and traffic management applications [51-53]. On the other hand, recent applications have taken the VC beyond the scope of transportation. Examples of such applications include crisis management in cases of critical situations of natural/unnatural causes in which communication networks can be damaged or affected [54], Mobility Anticipation/Trajectory to predict the position of specific vehicles in the future for assisted driving, [55, 56], and Autonomous driving [57, 58], which is gaining popularity to be used in health care scenarios and emergencies.

2.3.4 Challenges

As mentioned before, vehicular cloud computing is still at its very early phases. Researchers need to address several open issues to achieve the full potential of VC.

1. **Heterogeneity**

In vehicular networks, no single communication technology is sufficient to cover all the connectivity requirements due to mobility, the dynamic ad-hoc nature of the network, and diversity of services provided. Also, the limitations of wireless coverage and the handover from one communication interface to the other need to be considered [18, 59].

2. **Mobility**

The changing nature of the vehicular network as vehicles join or leave the network requires regular changes in the routing of data. A vehicle should know promptly when in the network topology changes or vehicles leave or arrive. This is particularly crucial in latency sensitive applications and safety services [60] . The speed at which these changes happen represents another challenge. Moving vehicles make load balancing and content distribution more challenging. Handling vehicular mobility requires collection of information about the network and sending back any responses in a timely manner to avoid any data loss or service disconnection [61].

3. **Scalability**

The huge number of vehicles roaming the streets throughout the day are all potential resources waiting to be utilised properly. However, this might not be as easy as it seems, since adding more vehicles to an ad hoc network would increase the complexity of the network and complicates its management. The efforts to improve vehicular network scalability usually involve the use of RSUs for control and management [37].

4. **Security**

In vehicular networks, security needs very critical attention due to the dynamic and ad-hoc nature of vehicular networks where a malicious vehicle can join the network, disrupt its functionality, and leave at high speed. The security threats can be against members of vehicular cloud (Vehicles and/or Infrastructure components) or the communication (V2V, V2I, and recently V2X). Special consideration should be given to the vehicles' identification and the need for

authentication and access granting (or denying) in real time to avoid illegitimate members in the cloud [62, 63]. The diversity of wireless communications interfaces opens the door to several types of attacks such as Denial of service (DoS) through flooding of communication channel. The integrity and confidentiality of the data exchanged in the VC can also be threatened by data tampering attempts [64] [65].

5. **Privacy**

Privacy is a major concern in VCs especially for the users, who would need to be assured that their personal data would not be violated [65]. A user needs to be assured of their anonymity in the sense that his private information such as location or address cannot easily be linked to data and services used in the vehicular cloud.

6. **Incentives**

Also, as the vehicular cloud is distributed and there is no centralised provider of services, one perspective of research is how to judge the value of a vehicle participation in a specific cloud and how it should be rewarded.

2.4 Vehicular Virtual Machine Migration (VVMM)

Implementation of virtual machines (VMs) is one of the well-known approaches for resource provisioning and sharing. The dynamicity of the network topology and the mobility of the vehicles at variable speed as well as the ownership, and hence availability control, of the vehicle by individuals makes VVMM both attractive and challenging concept [66]. In the context of vehicular cloud, the VM

is mostly viewed as the virtualised infrastructure resources of the VC such as computation and storage components. When a vehicle (or RSU) leaves the VC, its status and workload need to be migrated to another (host) to avoid service intermission and delay. However, there is still shortage in the research regarding VVMM. Techniques for virtual machine migration in conventional clouds [67] are not suitable for vehicular clouds as the motives and goals of migration in a vehicular cloud scenario can be different. Figure 2.2 shows some of the reasons for conventional virtual machine migration and the additional ones in the vehicular cloud that pose more challenges. Overcoming the intermittent connectivity, heterogeneity of VC components, and finding the best candidate to receive the migrated VM are among the open issues addressed in VVMM. Strategies for the migration are evaluated based on several metrics such as service disruption level, impact on delay, migration success rate (or migration drop rate), and percentage of resources utilisation. The work of [68] developed algorithm for VVM in which the searching for suitable new host one of four schemes, uniform, least workload destination, destination mobility aware, and least workload and mobility aware. The results showed the last scheme to have the best performance in terms of migration drop rates (most successful migrations). In [69], a mixed-integer quadratic model and heuristics algorithm were developed to address the problem of VM placement to roadside cloudlets as vehicle moves, while minimising migration and networking cost. Three layers cloud architecture has ben considered in [70], central cloud, roadside cloud, and vehicular cloud (vehicles only), and migration occurring from one layer to the other to maintain cloud service. A selective dirty page strategy has been adopted to improve

transfer efficiency and optimal resource reservation scheme was proposed to reduce migration drop rate.

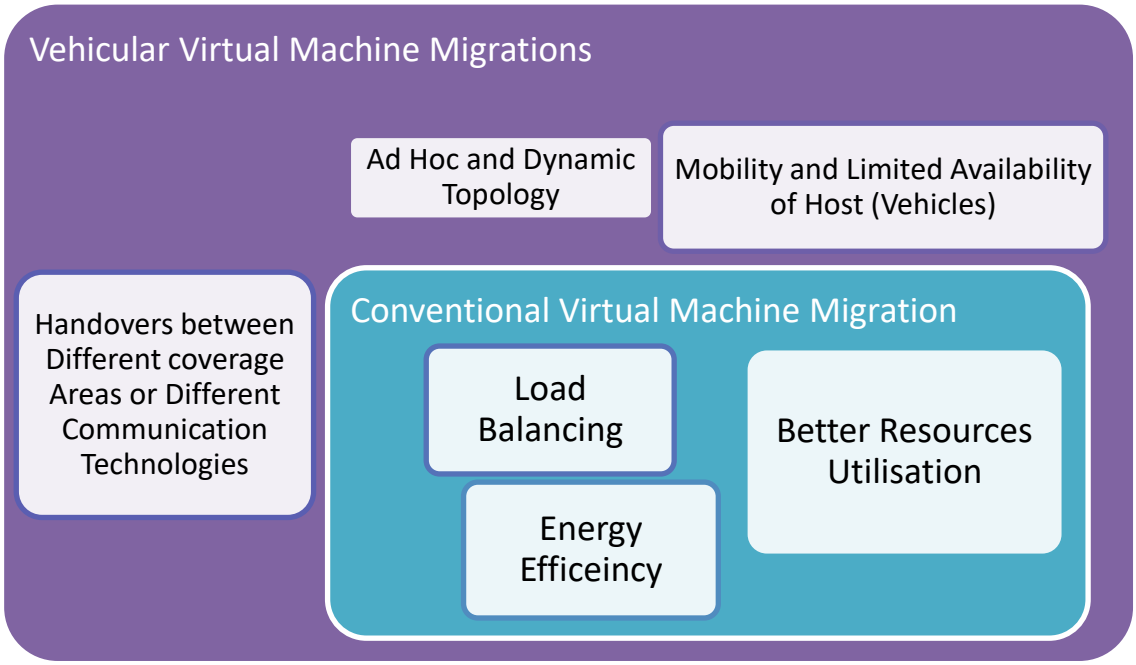

Figure 2.2: Main reasons for virtual machine migration in conventional cloud and vehicular clouds

2.5 Enabling Technologies for Vehicular Cloud

2.5.1 Virtualisation as a Concept

Virtualisation is the most popular technology to achieve optimum resource utilisation and sharing while maintaining QoS [71]. It gives each user the illusion that they are solely using the resources, wither the resource is an application, storage, server, or network bandwidth. In relation to network virtualisation, other concepts have emerged such as Network Function Virtualisation (NFV) [72, 73] and Network Slicing (NS) [74, 75]. NFV uses the standard physical network infrastructure to form virtual machines that serve different functions, as opposed to having dedicated function-specific hardware. In NS, the physical network is

sliced into several virtual networks with their own virtual versions of resources used to serve the client specific application requirements. Research on using the concepts of NFV and slicing in vehicular networks is still in its infancy but is believed to have high potential due to the distributed nature of vehicular networks and the presence of different communication interfaces that allow for network slicing. In [76], introduced an architecture supported by network virtualisation to improve VANETs scalability. Authors of [77] also used network virtualisation by combining previously none solutions for network isolation to form improved solution better suited to VC vision and optimising the quality of service.

2.5.2 Vehicular Software Defined Network (VSDN)

2.5.2.1 General Overview of Software Defined Network

Software defined network (SDN) as defined by Open Network Foundation (ONF) [78] is

"An emerging network architecture where network control is decoupled from data forwarding and is directly programmable"

The approach is characterised by two main features. Firstly, it totally separates the network control (setting of protocols and rules) from the network operation (application of protocols to data). Second, it handles all the management, updates, and configuration through software programs or commands, avoiding the impact of hardware diversity, heterogeneity, and complexity [79-81].

A widely approved structure for SDN is composed of three components as shown in Figure 2.3, control plane, data plane and applications [82].

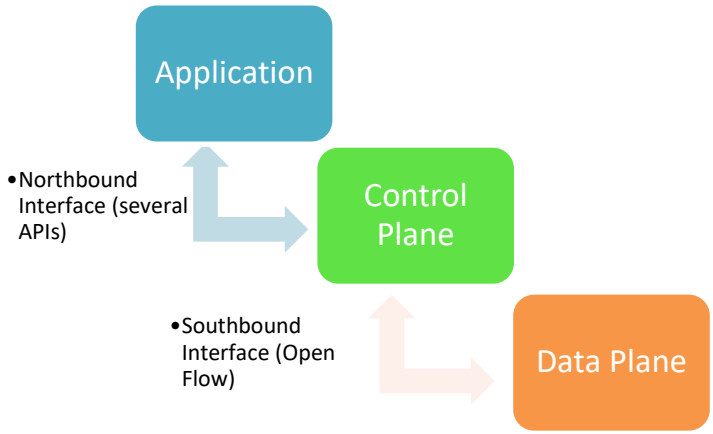

Figure 2.3: Software-Defined Network General Architecture

- **Control Plane:** It forms the core of the SDN approach between data plane and applications. It consists of controllers. The controllers are responsible for receiving the requests from applications and network status from data plane. They set and update forwarding rules and send them to data planes to be included in lookup tables. The communication with the data plane and applications is done using Application Program Interfaces (APIs). The interface with the data plane is known as the southbound interface and is implemented using OpenFlow, which is the most popular and successful protocol for SDN so far [83]. The interface with the application layer is known as the northbound interface and its implementation varies with application.
- **Data Plane:** It represents the network devices (switches and routers) that perform actions on data packets based on rules in lookup tables. The network devices are also responsible for collecting information about the

network, so these tables can be updated accordingly. SDN switches can be hardware switches, e.g., RackSwitch from IBM and MX-series from Juniper, or software switches (open sources and commercial), e.g., Open vSwitch and OpenFlowClick.

- **Applications:** The applications send requests to the network. The data packet forwarding rules are set in a way that serves the requests efficiently. They compromise the different kinds of services required to achieve the tasks in the network. They can be either general purpose services related to the network status and operation like network management, security, utilisation, or they can be requirement-specific applications dedicated to specific use cases, such as wireless networks [84, 85], optical networks [86, 87], and edge computing [88].

2.5.2.2 VSDN Architecture

Vehicular SDN (VSDN) is becoming an essential enabler of vehicular networking, as the decoupling of planes and programmability of controllers are suitable for the mobile and dynamic nature of vehicular networking [89]. In the most generalised form, the taxonomy of SDN-based vehicular networks consists of the following components:

- **Data Plane:** In the context of vehicular networking, vehicles are the main devices that are used in routing and forwarding of data packets, replacing the regular switches in other networks. In addition, other components usually found in vehicular networks can also be included as data plane devices such as the RSUs, cameras and traffic lights.

- **Communication Technologies:** Vehicular networks are unique in their communication interfaces heterogeneity. This can cause a challenge for the controller to smoothly run each interface. On the other hand, with the amount and diversity of data expected from vehicles and RSUs, controllers would have better chances of choosing a suitable technology when needed.
- **Controller:** The SDN controller coordinates the entire network. The controller has a general and abstract view of the network components that allows it to develop an informed decision regarding packet forwarding rules.

One of the earliest designs for VSDN was done in [90], which provided an architecture with a centralised controller and an abstracted view of the data plane with multiple communication interfaces available. One major argument against this architecture is the centralisation of control of network at a single point, which increases the risk of failure as well as reducing the flexibility and scalability of the network considering the mobile nature of vehicles where a moving vehicle can quickly move out of the coverage area of the controller.

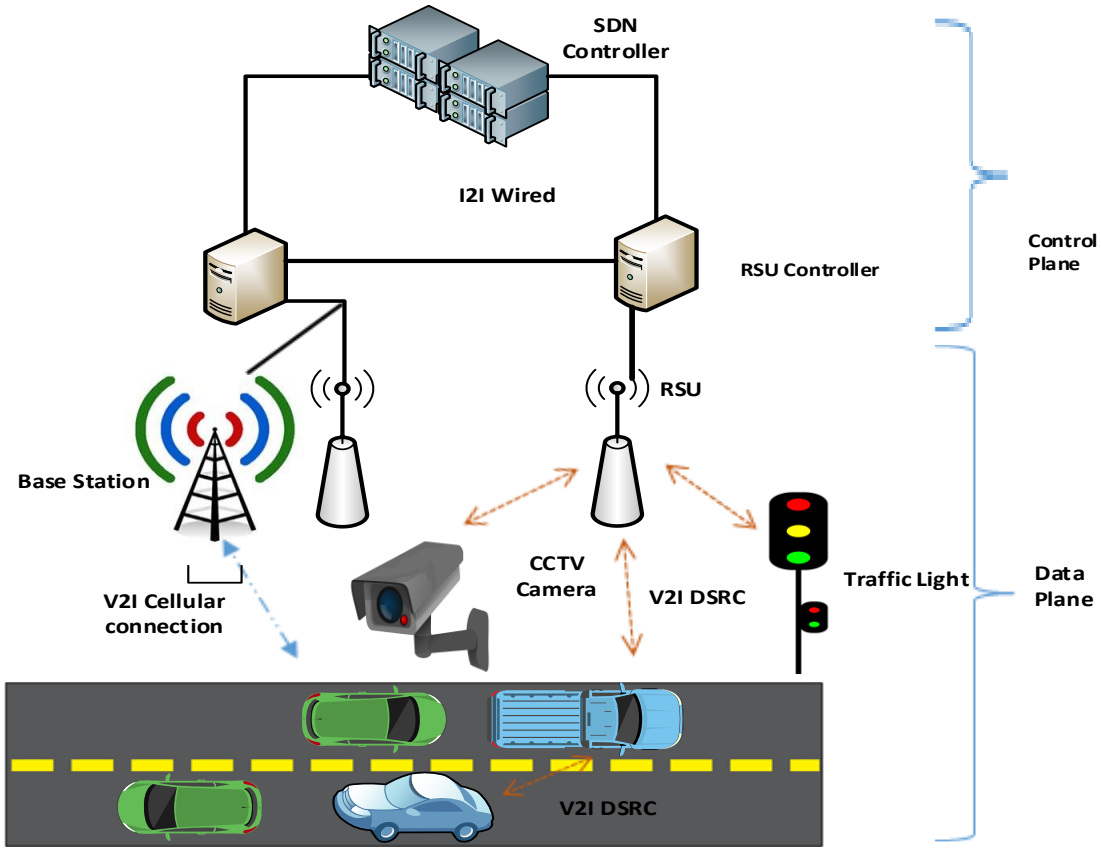

Figure 2.4: SDN-Based Vehicular Architecture with Hybrid Control

A simplified version of an enhanced distributed (or Hybrid) control is shown in Figure 2.4, which makes the backbone for many current vehicular network proposals. In this architecture, the control is hierarchal. The first level is mini controllers used to coordinate the operation of a group of RSUs and vehicles in a small area. The mini controllers communicate with each other and control the vehicles if they are within their limited coverage area. The mini controllers communicate with a higher level SDN controller when vehicles are approaching the edge of a mini controller area or when more global decisions need to be made.

2.5.2.3 VSDN Implementation

The implementation of SDN in vehicular networks is still evolving. It can be used to unlock the potentials of many use cases, especially in conjunction with other technologies such as 5G, virtualisation, and edge processing nodes. Research in this area takes many directions due to the multiple VN issues that are believed to be solvable with SDN. In general, the work can be categorised into four main groups as shown in Figure 2.5

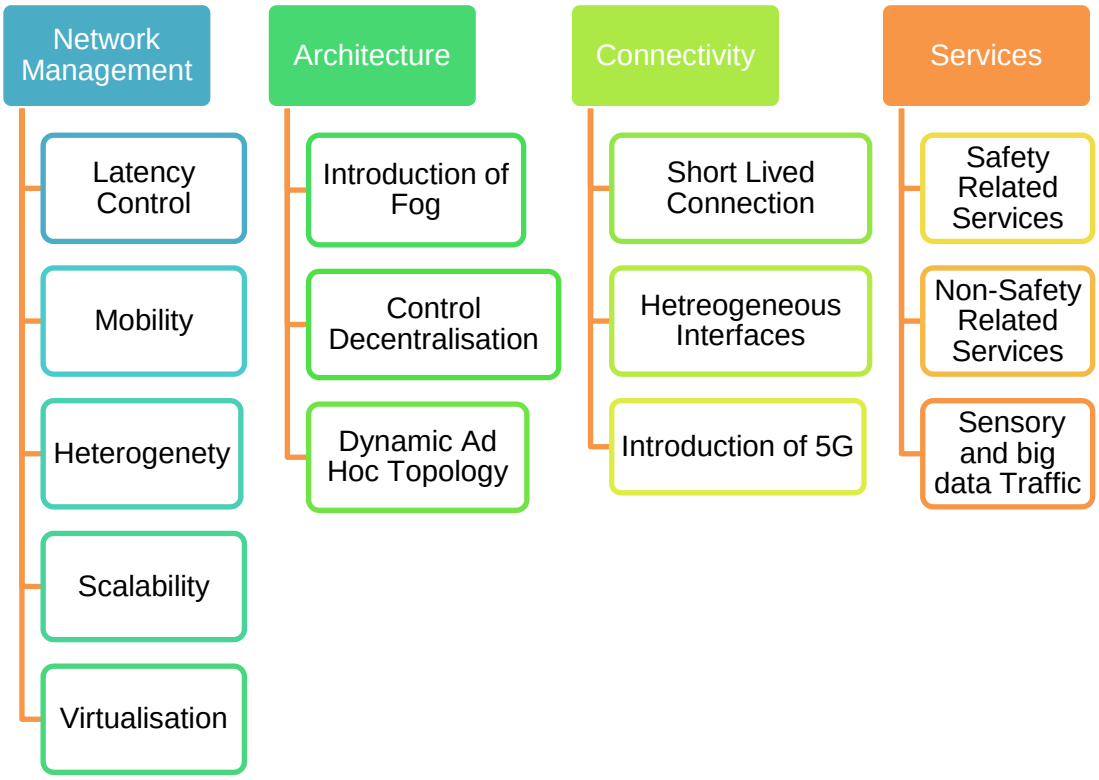

Figure 2.5: Software-Defined Network implementation in vehicular networks

1. **Network Management:** SDN can be used to improve the management of vehicular networking in light of the mobility of the vehicles, as well as handling the heterogeneity of the network components like vehicles and RSUs. It is also believed to increase the scalability through handovers between different SDN controllers over wider areas. As explained above, SDN and virtualisation techniques are used to address a lot of these issues [91-93].

2. **Architecture**: Many of the SDN implementations in vehicular network introduce new architectural designs that benefit more from hybrid or distributed type of control that is believed to be more suited to the mobile and ad hoc nature of the vehicular networking, which suffers the loss of connection with SDN controller [94]. Using fog concepts, distributed SDN control is being implemented. The distributed controllers are strategically placed to manage as many areas as possible with communication between controls to be updated about the movement of vehicles. An architecture that combines SDN and Fog was developed in [95], which aimed toward better resource management and reduced latency. Another architecture in [96] places Edge servers and SDN controller at BS, making it easier to communicate without affecting the overall performance and latency of the system. The choice of the access technology used for a specific application is to be decided according to the application specific requirements.

3. **Connectivity**: One of the many issues of vehicular networks is the variation of communication technologies available and the intermittent connectivity. Both issues are believed to be addressable with SDN by means of communication interface handover through SDN controllers [29, 97]. In [98] authors proposed new architecture that combined 5G technology and SDN to reduce the handover in the network. Only a gateway vehicle is directly connected to the RSU, while the vehicles are connected to each other, to avoid frequent handovers to RSU which greatly impact QoS. The work of [23] introduced cell-less network, where the vehicles communication through moving access points on vehicles and controlled by software-defined cloudlets, producing better V2V performance.

4. **Services**: Having SDN is expected to improve the performance of the vehicular network applications, in terms of reliability, better content distribution, and improved safety and driving experience. It also promises to open the door for other types of applications related to big data and sensory networks, where a lot of data collection and analysis is required [99-101].

2.6 Related Topics to the Thesis Work

2.6.1 Cloud Computing

Revolutionary developments in application types have taken the technology sector by storm in the last decade. The number of users has also been rapidly increasing and they increasingly demand higher computation, storage, and

bandwidth resources. This was accompanied with an explosive trend towards remote resource provisioning to accommodate these demands. Cloud computing was introduced as the first paradigm for remote resource provisioning. Cloud computing providers such as Google and Microsoft present their computation, storage, and networking resources to users around the world through strategically located data centres (DC). Any user, whether individual or an enterprise that needs a specific kind of resource or service, can rent it from the provider over the Internet and pay only for what they need. The types of services provided through cloud can be classified into three main types [1]:

1. Software as a Service (SaaS): In this model, users in need of a software application can use an online version over cloud on a pay-as-you-go basis. The user does not have to worry about purchasing the application or licensing or compatibility with hardware.
2. Platform as a Service (PaaS): The cloud provider has the required environment for building applications or services.
3. Infrastructure as a Service (IaaS): This model includes the most diverse kinds of services as it includes all the requirements for users' infrastructure. It can be further divided intro Computation-as-a-Service, Storage-as-a-Service, Network-as-a-Service, Database-as-a-Service.

A key benefit of the CC model is the reduced cost for users and providers. The users save the cost of buying, establishing, and maintaining hardware and software resources. For the providers, the cost of providing the services is

expected to be well compensated through the profit they make from renting the resources.

2.6.1.1 Data Centre Architectures

A lot of effort was dedicated to designing new data centres architectures to accommodate the diverse applications and services of cloud computing. The earlier more conventional designs were based on a tree topology with three layers, as shown in Figure 2.6. The lower layer consists of racks containing servers connected through an access switch, Top of Rack (ToR) switch, that connects them to the next layer. The second layer is made of aggregation switches connected to the core routers in the last layer, which is responsible for moving data into and out of the DC [102].

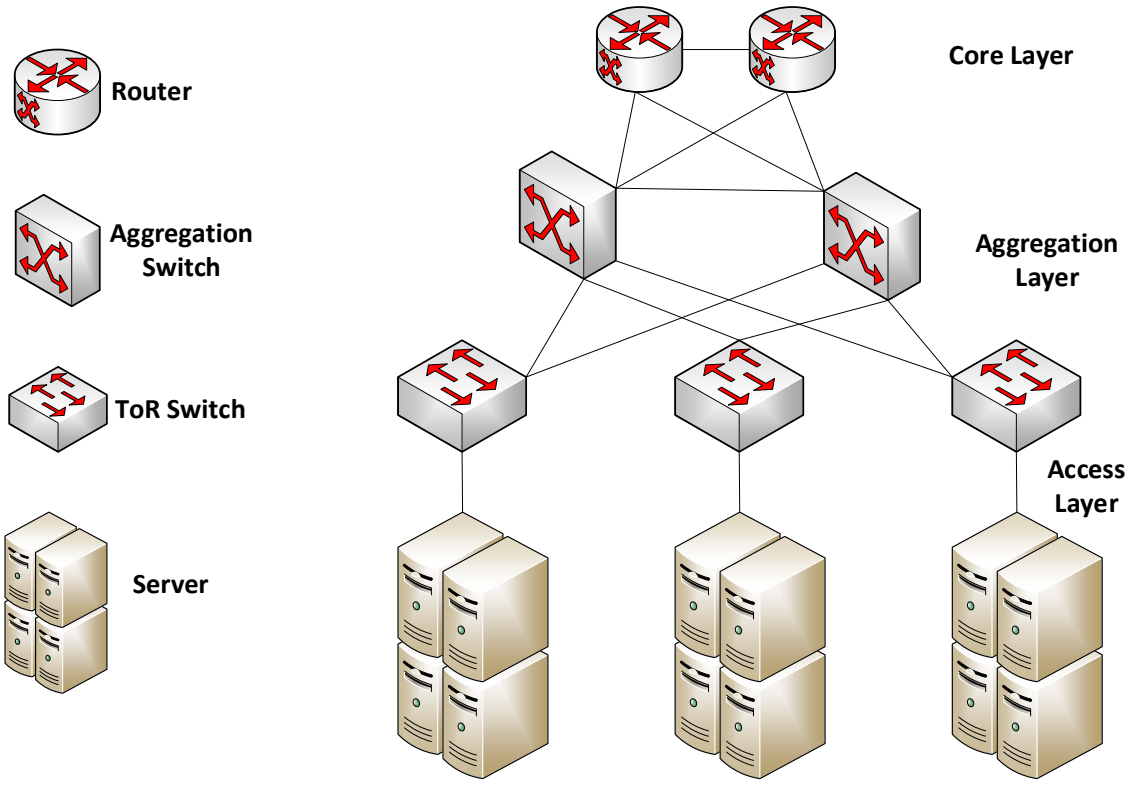

Figure 2.6: Tree Topology DC architecture

In [102, 103] the drawbacks of conventional DC architectures are discussed. One of the problems is congestion due to oversubscription as servers compete for the bandwidth. Another problem is packet dropping due to overloading of the switches' buffers. Fault tolerance mechanisms are not available in conventional DCs due to the low connectivity of the components in the upper layers. Hardware failure is hard to overcome and can lower the efficiency of the DC. Poor utilisation of resources is another problem as well as the uneven load balance over the links of upper layers. The huge power consumption is also an objectionable issue with conventional data centres. Improvements on DC architecture fall into three main categories. The first is known as switch centric data centre, where switches are

the main component in routing and interconnection. Examples of this type are fat tree topology [104], VL2 [105], and QFabric [106]. The second type is server centric data centres, in which servers play a role in packet forwarding and routing. Examples of this type are the BCube architecture [107], and FinConn [108]. The last type of architecture is the optical data centres, which uses optical fibres and optical switches, as in Helios [109] and Petabit [110].

2.6.1.2 Challenges

Cloud computing was introduced as the first paradigm for remote resource provisioning. However, the growing dependency on cloud is curtailed by the increased costs and power consumption. The Cisco Visual Networking Index 2017-2022 report predicted a threefold increase in global IP traffic between 2017-2022 reaching monthly traffic 396 Exabytes (EB) by 2022 [111]. The report also predicted the number of **connected** devices to the IP network to reach 28.5 billion by 2022 [111]. The increasing use of cloud computing challenges the existing networking paradigms in many ways. In the following, we mention a number of these challenges that are increasingly considered in research.

1. **Security and Privacy**

   One of the main concerns that affect the user decision to use remote service is their trust in the providers' ability to handle their data and requests privately and securely. Sending the data over long-distance networks makes the data prone to attacks, so not only the cloud servers need to be secure but also the network over which data is transmitted. In addition, one user's environment, if unprotected, might cause security threats even with the implementation of

virtualisation and resources sharing mechanisms [112]. The share paradigm of cloud computing raise security questions regarding the servers availability, access control, multitenancy, and the security of big data computation [113]. Closely related to security is the privacy issue. Placing user's data in remote servers removes them from his control. Strategies must be provided to preserve personal data confidentially, identity protection, and privacy preservation [114].

2. Energy efficiency

Research showed that CC has become one of the major power-hungry technologies, which in turn put strains on the power / emission budget allotted to the ICT sector [115]. This made energy efficiency research one of the most active fields. Energy-efficient distributed clouds were studied in [116]. Green data centres and the use of renewable energy were investigated in [117-119]. The energy efficiency of virtual machine embedding was studied in [120]. In [121], the authors proposed a system for dynamic reallocation of the virtual machines that guarantees both energy efficiency and quality of service (QoS) of cloud services.

3. Latency

For most users, cloud data centres are geographically far. This is especially significant when considering delay-sensitive applications. The latency introduced by the network and inter-cloud operations should be carefully studied to understand which applications are better suited to the cloud and when to make the decision to off-load to the cloud [122].

4. **Load balancing**

As cloud data centres handle large volumes of traffic, it is crucial that the traffic is distributed fairly between cloud servers to achieve better utilisation of resources and avoid points of failure [123]. Virtualisation is one promising solution that is expected to solve issues related to load balancing [124, 125].

5. **Resilience**

In literature, the definition of resiliency can be diverse and interpreted differently by each author. Generally, it is the ability of the system to continue providing services with level of quality, when unusual events occur. The efficient resources of data centres make cloud computing resilient by nature [126]. However, unexpected faults and failure can happen and require recovery plans, or else both users and cloud services providers would suffer the failure costs [127]. In the CC paradigm, resilience strategies are formed for the infrastructure (composed of servers and network) [128] and the applications. Strategies for infrastructure resiliency include process checkpoints [129] and redundancy, switches and routers redundancy (examples in well-known architectures like BCube, VL2, and QFabric), and virtual machine migration [67]. Applications resiliency is defined as the survival strategies of applications and the data content against infrastructure failure and cyber-attack. It includes mechanisms such as applications self-correction and data and code replication.

2.6.2 Edge/Fog Computing as an Alternative to Cloud Computing

The rapid growth in global traffic and applications diversity is exhausting cloud resources making cloud computing unaccommodating to some application-

specific requirements. For latency-sensitive applications and applications in need of continuous connectivity, the cloud might be a poor candidate as a service provider. Also, as smart cities and their ubiquitous computing requirements grow, sending all the data to the cloud becomes infeasible, as well as bandwidth wasting. At the user end, devices with different kinds of resources and capabilities are becoming pervasive [23]. As an alternative to cloud computing, dedicated resources deployed in users' proximity and the users' resources can be exploited collectively to serve other users in need of resources leading to the concept of Edge Computing / Fog Computing (EC/FC). In this paradigm, computation, storage and networking resources, and management are available in the proximity of end users at the edge of access networks. Note that edge/fog computing is not meant to replace cloud computing. Cloud computing will be needed for highly demanding applications and to maintain QoS. A comparison between Edge computing and cloud computing to show the points of strength for edge computing is shown in Table 2-2.

Table 2-2: Cloud Computing vs Edge Computing

Points of Comparison	**Cloud Computing**	**Edge Computing**
Location & Size	Centralised large data centres installed in specific geographical locations away from end user	Distributed in small devices and components at the edge in proximity of the end users
Operation	Cloud providers fully operate and control the cloud services.	Control can be handed over to devices to communicate with each other with

	Usually these are the big players in the field of technology	minimum intervention and operation can be carried out by small companies
Connectivity	Requires Internet connectivity to use the service	Can be Internet independent and services can be provided through the local area network
Applications	Suited for larger applications of longer lasting data or requires more processing power	Suited to latency-sensitive or locally relevant applications
Cost	Costs more to build new data centres and network connections, costly maintenance Consumes more power for network transmission which add to the cost	Idle End users' devices are exploited Costs less to integrate fog with existing network

2.6.2.1 Edge/Fog Computing Paradigms

In the literature, the concept of shared resources at proximity to end users has been forked into different paradigms. The works in [130, 131] provide an extensive review of the different forms to implement this concept. Some of the most common paradigms and their distinctive features are shown in Figure 2.7.

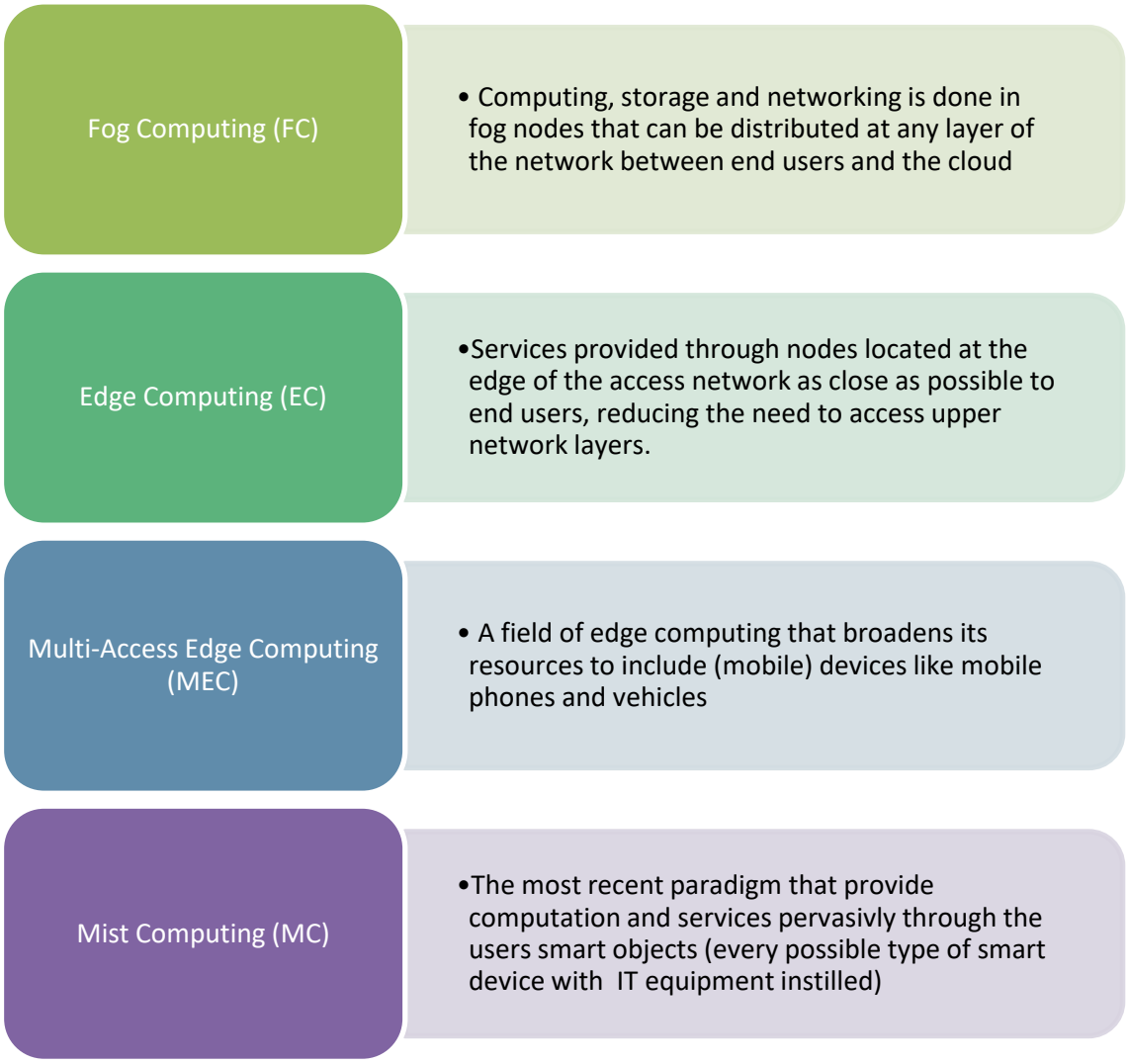

Figure 2.7: Paradigms of Edge/Fog Computing

2.6.2.2 Implementations of Edge Computing

Edge computing attracted attention especially in combination with Internet of things (IoT) and Issues addressed in the literature include energy consumption, service delay, and resources allocation and provisioning [35]. In the following we give a brief review of work related to these areas.

2.6.2.3 Energy Efficiency

One of the main objectives of using edge computing to complement cloud computing is to reduce energy consumption. A comparison of energy consumption of fog computing and cloud computing architectures is found in [132]. The authors in [133] presented an IoT architecture for energy efficient distributed services. A good example of energy-efficient mist computing is given in [134], where services are embedded in IoT objects. Energy-efficient piece delivery using fog servers in vehicular environments is found in [135].

2.6.2.4 Delay Control

Resource allocation, carried out with response time and the QoS as metrics, is presented in [136]. The impact of fog computing on the service delay is studied in [137]. The use of fog computing in combination with vehicular networks for balanced task allocation and minimised delay is presented in [138]. The work in [139] jointly minimises the transmission and processing delay of fog computing in access networks.

2.6.2.5 Resources Allocation

Aside from energy and delay, resource allocation studies in the context of edge computing focus on improving resources utilisation, reliability, and load balancing. In [140], the authors studied computational offloading to distributed nodes, which requires mechanisms for load balancing to achieve quality of service. New techniques for hierarchically clustering and load balancing of resources are introduced in [141] to avoid congestion. A cooperative load

balancing scheme in which overloaded edge servers exchange requests with ones less congested to achieve lower delay and quality is presented in [142].

2.6.2.6 Leverage on Vehicular Network

The use of edge computing to augment vehicular network capabilities has been thoroughly investigated. One of the earliest works is in [37]. where road-side units (RSUs) are used as cloud nodes for VANET management and delivery of non-safety applications. A load balancing scheme for task offloading between vehicular networks and edge nodes is introduced in [143]. A mobility support scheme using the Radio-Access Network and the edge node infrastructure to control the vehicular network traffic is introduced in [144]. A security mechanism using fog computing to support vehicular cloud is developed in [145].

2.7 Optimisation

2.7.1 Mathematical Programming

Optimisation aims to find the optimum solution of a problem. Optimisation has applications in almost every discipline of business and science. The problem to be solved is represented in mathematical form as an objective function to be maximised or maximised. The objective function is a function of the problem variables, usually bounded by lower and upper bounds, and is to be optimised. The model is controlled by set of constrains that need to be satisfied for the solution to be valid [146]. A general representation of the problem is as follows:

$$X = x_1, x_2, \ldots, x_n \quad (2.1)$$

$$lb_i \leq x_i \leq ub_i \qquad i = 1, 2, \ldots, n \qquad (2.2)$$

$$Minimize\ or\ Maximize: f(X) \quad (2.3)$$

$$Constraints:\ C1(X)(\leq or \geq or =)\ c1 \ldots\ Cm(X)\ (\leq or \geq or =)\ cm$$

$$m = number\ of\ constraints \quad (2.4)$$

The process of presenting an optimisation model in this way is called mathematical programming and the resulting model is known as mathematical programme.

Optimisation problems can be divided into two main types, linear and nonlinear Optimisation. Linear Optimisation requires the objective function and all constraints to be in linear mathematical form. The more general form, namely nonlinear optimisation does not have this restriction, so functions can be of any mathematical solvable form. The feasible values of the variables are given by a continuous space of values. Problems with variables that can only take discrete values are referred to as discrete optimisation problems, and in case of linear problems, they are also known as integer linear programmes [146, 147]. Problems with a combination of integer and non-integer variables are referred to as mixed integer linear programmes (MILP), which is used in this work

2.7.2 Modelling Network Problems

Network optimisation problems are very popular due to their many real-life applications, and linear programming is a powerful tool in modelling and solving

them. Whether the network under study is a city transportation system, a delivery map, or a telecommunication network, the concepts are the same and models can be generalised to represent any of these examples. Networks are modelled as nodes and links that connect these nodes. The nodes can have resources and/or demands. A node (source) would want to send a demand to another node (sink or destination). The demand flows through the links from source to destination forming the traffic of the network. The flow of traffic in and out of nodes is subject to flow conservation to ensure that all traffic is routed from source to destination. Traffic flows are also subject to constraints on links capacity to ensure the physical capacity of links is not violated. The links are also associated with costs that can represented financial cost, length of link, or time taken in transferring the data along the link. Examples of objectives of network optimisation problems include minimising the cost of routing traffic, finding the shortest routes, maximising the served demands…etc. [147, 148].

2.7.3 Solving Algorithms

Algorithms are the ordered set of instructions used to solve a specific problem. In optimisation, the optimisation algorithms try to find the best solution to the optimisation model from the set of feasible solutions. Solving an optimisation model tends to be computationally hard and the hardness increases as the problem size increases. Therefore, optimisation algorithms not only try to find the optimal solutions but also try to reduce the computation complexity. We list some of the well-known algorithms below.

1. **Simplex method**

The simplex algorithm was invented by Dantzig in 1940s to solve linear optimisation problems. The concept is to start at one of the corner values of the feasible area of the solution space and work through adjacent corner values and compare the objective value in search for a better objective value. When no improvements can be made in the objective, then the optimum solution is found [148].

2. **Branch-and-bound method**

The algorithm starts with a set of all feasible solutions of the problem forming the root of the solutions tree. The solutions are then branched from the root into smaller groups of solutions that satisfy the upper and lower bounds on values (the constraints of the model), and for which the objective function is calculated. The solutions are branched until all constraints are met and the optimal objective is chosen [149].

3. **Cutting-plane method**

This algorithm applies linear inequalities to cut down the optimal solutions. A combined algorithm of branch-and-bound and cutting-plane is known as Branch-and-cut and is known to be more effective in solving MILPs.

A cost, referred to as computation complexity, is associated with the executions of the optimisation algorithms and the number of arithmetic (and memory) operations it requires. The number of operations in optimisation models is known to increase rapidly and exponentially with the size of the optimisation problem

(larger variables, constraints, input data, the space of possible solutions... etc) [146]. For example, if L is the problem size and n is a constant, then as L increases, the increase in the operations would be in the order of n^L. This place optimisation problems in the class known as NP-hard (non-deterministic polynomial time) complexity [150]. In practical sense, the exponential growth is exhausting to the hardware resources such as processing and storage and it exponentially increases execution time that can extend to hours or even days.

There are many commercial solvers now to facilitate the running of optimisation programmes and find the solutions in relatively short time. In our work we wrote our MILP optimisation programmes using the AMPL modelling language and used the CPLEX solver on high performance computers (HPC) to run the models.

2.7.4 Heuristics

By definition, a heuristic is an approach for solving problems that helps speed up the procedure by finding a good solution, but not necessarily the best one. In the hierarchy of our research, heuristics come at the end after finding the optimal solutions using MILP optimisation models. A heuristic is developed to mimic the behaviour of the MILP optimisation model. A programming language such as C++, Python or MATLAB (used in this thesis for the heuristics in Chapter 4), is used to code the heuristic. The development of heuristics serves two purposes:

1. Heuristics give an independent approach to verify the results of the MILP optimisation models.
2. As the size of the optimisation problems grows, their computation complexity grows as well. Heuristics can be used to give real-time sub-

optimal solutions for larger problems and is expected to have polynomial time growth.

2.8 Summary

This Chapter introduced vehicular networks and focused on the vehicular cloud paradigm. We presented the existing components that enable the vehicular cloud vision, design fundamentals, different vehicular cloud models, examples of vehicular cloud implementations, and the challenges that face the full realisation of the paradigm. The topic of vehicular virtual machine migration was also introduced. Some of the enabling technologies for VC were discussed such s virtualisation and vehicular software defined networks. Sections for cloud computing and edge computing were included as they are closely related to the work of the thesis. A brief overview of the optimisation concept is also presented, including discussion of major networking optimisation problems and optimisation algorithms.

Chapter 3

Vehicular Cloud Architecture & Evaluation Methodology

3.1 Introduction

In this chapter, we introduce an energy efficient stationary vehicular cloud architecture, to be used in distributed processing. In addition to vehicular processing, the architecture provides processing at edge nodes and the conventional cloud. Generic smart city applications are considered to evaluate the performance of the architecture. The smart city applications can be classified based on the nature of their data generation as periodic / continuously running applications or on-demand applications. The first type is subject to service-level agreement (SLA) specifying the data rate and processing speed requirements. The second type represents on-demand smart city applications where users request a one-off demand. The problem of optimally allocating processing resources with energy and delay minimisation is addressed by developing Mixed Integer Linear Programming Model. The model results are further verified by the development of heuristic to run in real-time. The following subsections explain the architecture and evaluation methodologies in details.

3.2 Vehicular Cloud Architecture

In the following subsections we present the proposed energy efficient vehicular cloud architecture from three different perspectives, the processing layers, network communication interfaces, and control and coordination.

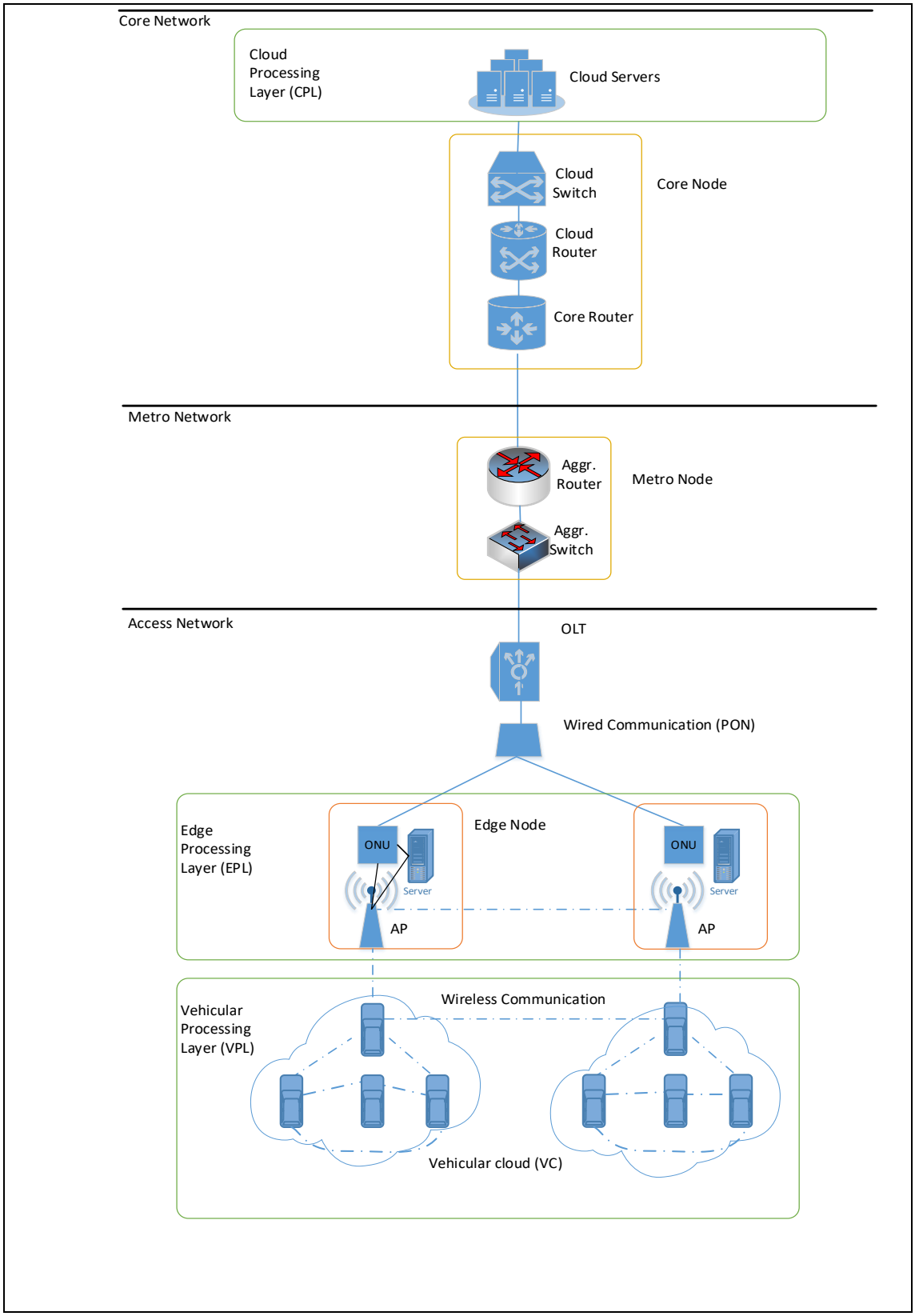

Figure 3.1: End-To-End Vehicular Cloud Architecture

3.2.1 Processing Layers

As seen in Figure 3.1 the proposed distributed architecture is composed of three processing layers:

Vehicular Processing Layer: The first layer is composed of stationary vehicles equipped with high-performance on-board units. Other vehicles can serve the demand if they are willing to share their resources. The vehicles can dynamically cluster to form temporary clouds. Each vehicular cloud is formed under the control of an edge node.

Edge Processing Layer: The second layer is formed by edge nodes equipped with mini servers dedicated for smart city applications. This layer provides users with another nearby processing destination. In addition to the mini servers, the edge node encompasses an access point (AP) to communicate with vehicles and an optical network unit (ONU) to connect to the passive optical access network (PON).

Cloud Processing Layer: The last layer is the conventional cloud, which is geographically distant but of powerful computing capabilities. The cloud is connected to a core network node through switches and routers.

The MILP formulation introduced in the next section identifies the optimum solutions given the trade-off between having powerful energy efficient servers at far located clouds, accessed by traversing multiple network layers, and using the less powerful, less energy efficient closer processing resources offered by vehicles and edge nodes.

3.2.2 Communication Interfaces

The use of vehicles provides communication technologies heterogeneity, which is both an attractive and challenging feature of vehicular networks. Vehicles support different communication interfaces including dedicated short-range communication (DSRC), bluetooth, WiFi, and cellular, and a lot of effort is dedicated to the optimal usage of these interfaces [18, 59]. In the proposed architecture, vehicles communicate with each other using DSRC as it provides high data rate and good coverage [11, 16]. The vehicle to vehicle (V2V) communication is not limited to the same vehicular cloud. Vehicles belonging to different clouds within the communication range of each other can communicate peer to peer using DSRC. For communication between vehicles and edge nodes (V2E), WiFi is used. Edge nodes communicate with each other in peer-to-peer manner using WiFi. Higher data rate V2V communication can also be supported through the WiFi interface. Edge nodes, as mentioned above, are equipped with an ONU to connect to higher layers through a PON access network. The inter-communication between the devices composing an edge node is through Ethernet of high speed and low energy per bit, so the power consumed is negligible.

3.2.3 Control and Coordination

The decisions of where to serve a user demand, how much of this demand is to be served in a specific location, how the data associated with the processing

demand is routed to that location, fall under the umbrella of system control. These decisions need to be optimised to reduce the power consumption.

Generally, the main challenge of the vehicular architecture is the dynamicity and variation of the resources. Keeping track of these changes is crucial to making educated decisions. As mentioned before, each vehicular cloud is controlled by an edge node. Each edge node is assumed to have knowledge of the vehicles under its control and Edge nodes exchange information about their vehicular clouds resource availability. Based on this knowledge each edge node takes decisions on where to process demands coming from its vehicular cloud. In this work, the overhead created by the control and coordinating data is not considered. In such a distributed control architecture the concept of software- defined networks (SDN) comes into play [79]. The architecture can be further enhanced with the addition of a layer of *centralised* controller with global view of the network, to oversee and coordinate the edge nodes. Whether using the distributed or centralised approach, the need for dynamic response and frequent updates on the architecture remain the same.

3.2.4 Applications

The work of [151, 152] provided an extensive list of VC smart city applications that range from traffic management, urban surveillance, datacentres, infotainment, healthcare, and emergency management. Similar applications are anticipated to use the services of the proposed architecture. The application demands in this work are generated by the vehicles. However, the model is generic to accommodate having the demands produced by any node type, and

this would provide further test cases for evaluation in future work. The demands cannot be locally processed, i.e., it is assumed that a vehicle cannot process its own demand (partly due to capacity constraints and partly to encourage cooperation with other vehicles). However, a vehicle having a demand can still process the demands of other vehicles. Our assumption is that for vehicular clouds to be applicable, cooperation of vehicles owners is required. This can be achieved if they are provided with the proper incentive to provide services to others while discouraging local service of their own jobs.

3.3 Methodology

The performance of the architecture is addressed through the problem of resource allocation optimisation. A MILP model was developed to optimally allocate the resources according to set of constraints and to achieve minimisation of energy and delay. The MILP is a well-known tool for optimisation in engineering and science that have been around for many years, with available solvers that implement various algorithms and techniques. It addressed the cost, networking, and routing problems thoroughly and allows for easy mapping between the known procedures and the research problem to be tailored according to its specific objectives and constraints. The subsequent sections describe in general the MILP models developed in this work and explains the main parameters, variables, and objectives. Also, a heuristic has been developed to support the findings of the MILP and get almost-optimal results in real time.

3.3.1 MILP

As a model has been developed/extended for each of the evaluation chapters. The coming sections explain the main components shared by all MILP models. However, the actual formulation of the model equations and constraints are explained in each chapter separately for simplicity and improved readability.

3.3.1.1 Input data

1. Traffic / Processing demand

For evaluation of the model, applications' demands are generated by the vehicles and are composed of two parts, the data to be sent and the processing it requires. The demands can be classified based on the nature of their data generation as:

- Periodic (chapter 4 and chapter 6): Subjected to service-level agreement (SLA) specifying the data rate and processing speed requirements.
    - Examples: public service applications such as CCTV camera, climate sensors.
    - Traffic is generated in bps and processing is defined in MIPS.
- continuously running applications (chapter 5): On-demand generated data by users.
    - Examples: user request to process a document or media file.
    - Traffic is generated in bits and processing is defined in MI.

The traffic demand (Mbps) and processing demand (MIPS) of a request are related based on the estimation in [153] for smart environment applications, which are summarised in Table 3-1. On average, one Mbps is sent for each 2000 MIPS of processing demand.

Table 3-1: Traffic and Processing Requirements in Smart Environment Applications

Application Type	Average Data (kbps)	Estimated instructions/bit	Calculated Processing Requirement (MIPS)
City Monitoring Through WSN	200	500-5000	100-1000
Connected Vehicles	200	50-2000-5000	10-400-1000

For the processing demand, the use of MIPS metric to indicate processor efficiency is not a fully accurate choice. However, an exact and concrete answer about processor efficiency is not dependent on a single metric but requires careful study and comparison of several point such as hardware architecture, clock speed, number of cycles, and MIPS. Such an answer is out of the scope of this work, and so the MIPS was chosen has it provide adequate presentation of the demand and resources to our purposes, as long as the hierarchy of the processing efficiency is kept intact with the vehicles having the lowest efficiency and the conventional cloud having the highest. Also, the MIPS metric has been used in literature previously, such as in [48, 154], to address resource allocation problems.

2. Number of splits

The VC architecture derives its usefulness from the utilisation of number of resources that can collectively deliver services like the ones provided by conventional cloud.

3. Physical Distance

The far geographical location of the conventional cloud is one of the motivations to have VC and edge processing. Therefore, the physical distance between any two nodes is a parameter that impacts the delay calculation. Also, power consumption of the traffic traversing the wireless network is a distance-depend. the distance factor has a special importance in VC architectures, due to the ad hoc topology of the vehicles (affecting the networking power consumption), or/and the mobility of the vehicles which makes the distance a changing parameter.

4. Maximum & idle power consumption

The idle power consumption is significant in conventional cloud servers. It is expected to have major impact on the power consumption of the system. Whenever possible, a value for the maximum and idle power is obtained from datasheets when running the model. The idle power value specifically is not always presented. Therefore, idle power for networking devices is taken as 90% of the maximum power, based on estimations in [132]. According to [111], machine to machine (M2M) traffic will be 7% of the global traffic by 2022. Connected cars traffic and connected cities, as part of the M2M traffic, are the

fastest growing types of applications. Together they are assumed to make 13% of the traffic [111]. So, for the portion of the idle power consumption of network devices attributed to our application types, we are assuming (0.07 * 0.13) of the total idle power of each device. According to [155], the idle processing power consumption of servers is about 60% of the maximum, but since the reference is an old one, we are assuming a more efficient modern serve which consumes an idle power of around 50% of the maximum.

5. Maximum data rate

Each node in the architecture has a maximum data rate that should not be exceeded by the traffic through that node. For the vehicles and edge nodes, as mentioned earlier, two different interfaces are used and so, two different maximum data rates that need to be preserved. For the wireless interfaces, noisy links are out of the scope of this work and all the data rate is used in serving the demand.

6. Transmission and reception energy per bit

To calculate the networking power consumption, the energy per bit needs to be known. It is a function of the device power consumption and the maximum data rate. For the wireless communication, the energy per bit is divided into transmission and reception energy per bit.

7. Processing capacity & Efficiency

To calculate the processing power consumption, the power per million instructions, referred to as processing efficiency, needs to be known. It is a

function of the processor power consumption and the maximum processing capacity.

8. Power usage effectiveness (PUE)

The PUE is the ratio of the total power consumed by the node's IT equipment (networking and processing devices) and non-IT equipment (cooling, ventilation,..etc) to the power consumed by the IT equipment alone. It is an important measure of efficiency as modern computing and networking nodes require non-computing components for their operation, such as cooling and ventilation systems. An ideal PUE is equal to 1, which means all the power is consumed in performing IT operations.

3.3.1.2 power consumption

The processing and networking devices are assumed to follow a linear profile where the power consumption is composed of an idle power consumption which is the power consumed to activate the device and load dependent power consumption as seen in Figure 3.2. The load dependent power consumption is obtained by multiplying the device load by the energy per bit. For the processing power consumption, the load dependent part is calculated by multiplying the processing demand by processing efficiency. For the wireless communication (DSRC/WiFi), the networking power consumption is also distance dependent as power amplification is required to avoid signal fading over distance.

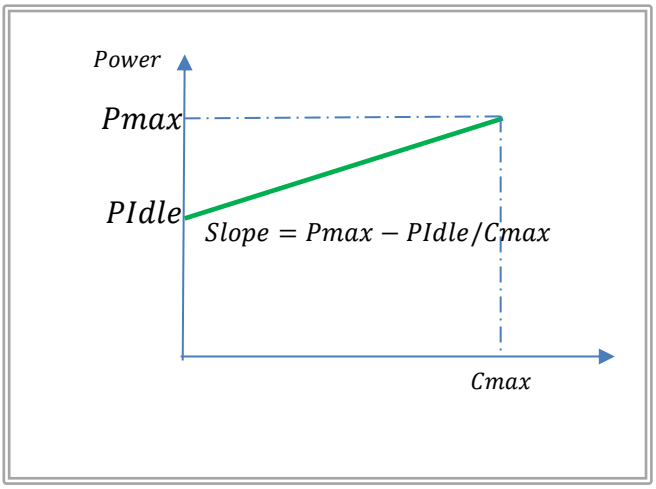

Figure 3.2: Power consumption vs load Profile

3.3.1.3 Delay calculation

The delay calculation in the architecture is the additive end-to-end delay of each hop from source to distention. For each hop, the delay is composed of five components:

1. Transmission Delay: the delay experienced at a node is the reciprocal of its transmission rate and it gives the time required to put the bits in a transmission link.
2. Propagation Delay: This is the time needed to travel through the network physical links between two nodes.
3. Processing Delay: the time needed to carry out the computing tasks, and it depends on the servers processing speed.
4. Queueing Delay: The queueing delay refers to the waiting time at relay nodes before re-transmission, which is a function of traffic arrival rate and transmission rate of relay nodes. Another queueing delay can be

calculated for the tasks waiting to be processed by the server. This queueing delay is set to zero in our model as the model is set up so that it does not assign a task to a busy / full server and would re-assign such a task to another location. We are also assuming infinite buffer size for each node.

5. Reception Delay: The time a node takes to receive the full traffic. This delay is ignored in the model as it is assumed a file need only to be partially received before starting re-transmission or processing.

For the edge nodes, which are composed of three devices (access point, ONU, server), the internal delay between the constituent devices of the node is ignored as they are integrated together.

3.3.1.4 Objectives

The evaluation of the VC architecture was carried out for processing resources allocation with the objective of energy and/or delay minimisation. The allocation of resources was subject to number of constraints such as flow conservation, bandwidth/processing capacity, and splits number limitation. Variations of the objective were introduced as follows:

1. Power minimisation (chapter 4 & chapter 6): energy efficiency has been tackled through power minimisation (in Watt). As the resources were allocated to serve processing demand, the total networking and processing power consumption was calculated for each node in the architecture. For chapter 6, the model from chapter 4 was further

extended to include mechanisms for resilience through redundancy in resource allocation, and the impact on power consumption was evaluated.

2. Delay and Energy minimisation (chapter 5): The study of the energy efficiency of on-demand applications needs to consider the energy consumption (in Joule) of the applications as the time required to serve the application varies with the processing and data rate requirements. In addition to the energy, the model proposed delay minimisation (in ms), and studied the impact of the proximity of the VC and edge node in comparison to the conventional cloud. It also compared the delay and energy of the centralised processing in the conventional cloud and the distributed approach of the VC. The delay and energy minimisation objective have been further subdivided to include:

    a. Joint minimisation of delay and energy with equal priority

    b. Joint minimisation of delay and energy with inequal priority

    c. Single minimisation of each delay component separately (transmission delay minimisation, processing delay minimisation, propagation delay minimisation, and queueing delay minimisation).

3.3.1.5 Evaluation test cases

The main question in this work is the comparison of the distributed resources in the VC and edge to the use of the conventional cloud as baseline. The evaluation test cases were chosen to reflect several factors that were expected to have an effect in a distributed architecture such as traffic demand size, processing

demand splitting and the competition over resources between multiple demands. We have test cases with only one vehicle generating a demand and other cases where we have demands generated by several vehicles. The number of splits allowed in the processing demand provide another evaluation perspective. As the processing demand is split, the associated traffic is delivered to each allocated destination. Also, the traffic associated with processing demand can be delivered fully or partially to the allocated processing node. An example of the full traffic delivery is a pedestrian collision avoidance application where one image is delivered to two processors and one of the processors is assigned to search for pedestrians for example, while the other processor searches for vehicles. An example of partial traffic delivery is sending half of the image to one processor and sending the other half of the image to the second processor where each processor searches for pedestrians in front of vehicles. Figure 3.3 shows the different test cases used in evaluating the VC architecture.

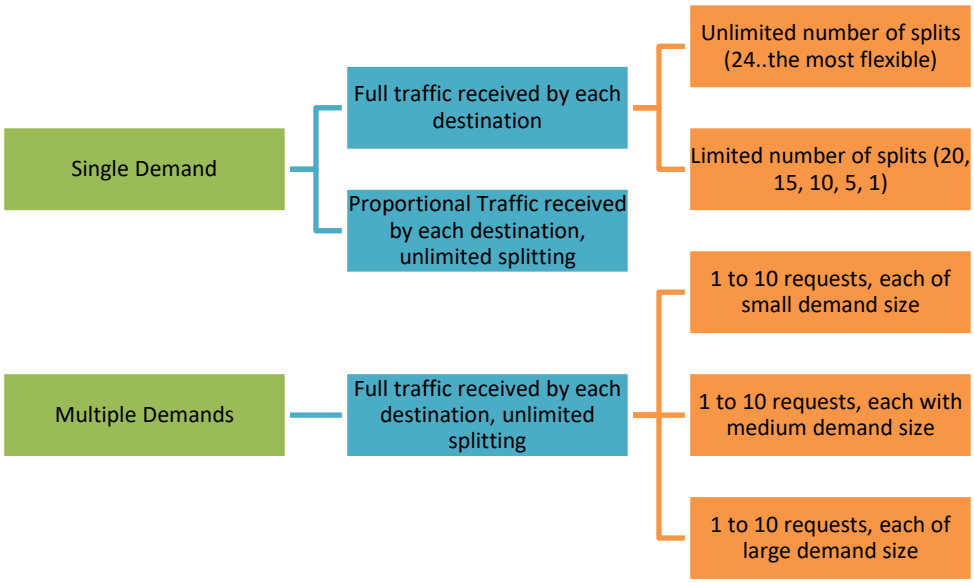

Figure 3.3: Evaluation scenarios

3.3.2 Heuristics

In chapter 4, a heuristic is developed based on insights obtained from the model to allocate processing demands in real time. The heuristics details are explained in chapter 4 for better readability and comparison with the MILP. The results of the heuristics were compared with the MILP for each of the above-mentioned test cases. Due to time limit, the heuristics was not extended to include results for chapter 5 and chapter 6, so manual verification for the MILP results were conducted for those chapters.

Chapter 4
Power Minimisation in Vehicular Cloud Architecture

4.1 Introduction

In this chapter, we consider the periodic / continuously running applications for which evaluation of energy efficiency can be based on the power consumption. We evaluate scenarios to establish the merits of the energy efficient vehicular cloud architecture. A MILP model was developed to optimally allocate processing demands to the three layers of the architecture with the objective of minimising the power consumption. The model compared the energy efficiency of processing scenarios considering different processing layers, and evaluated different test cases considering varying demand sizes, varying number of demands and the impact of processing demand splitting,

The subsequent sections of this chapter are organised as follows: MILP model is explained in Section 4.2 and the results are presented and analysed in Section 4.3. The heuristic and its results are given in Section 4.4 and the chapter is summarised Section 4.5.

4.2 MILP Model

A MILP model is developed to optimise the selection of processing destinations used to place processing demands and the routing of traffic demands between source nodes and processing destinations in the proposed architecture while minimising the power consumption, which is composed of processing power consumption and networking power consumption. Table 4-1 and Table 4-2 list the parameters and variables of the model.

Table 4-1: Model Parameters

ND	Set of vehicles
ED	Set of edge nodes
SD	Set of cloud servers
OLT	Set of OLT devices
MD	Set of metro nodes
CD	Set of core nodes
N	Set of all nodes in the architecture
Nm_n	Set of neighbouring nodes of node n, $n \in N$
U_s	Processing demand generated by node s (MIPS), $s \in N$
V_s	Traffic demand generated by node s (Mbps), $s \in N$
C_n	Processing capacity of node n (MIPS), $n \in N$

K_n	Processing efficiency of node n (W/MIPS), $n \in N$
S	Maximum number of processing nodes to process a demand
B_{nm}	Maximum data rate on the link between $n\ and\ m$ (Mbps), $n \in N, m \in Nm_n$
B_n	Maximum data rate node *n* can support (Mbps), $n \in N$
B^{VE}	Maximum data rate of the WiFi interface of a vehicle (Mbps) (this parameter is defined in addition to $B_n\ n \in ND$ to account for the two communication interfaces of vehicles).
B^{ONU}	Data rate of ONU at an edge node (Mbps) (this parameter is defined in addition to $B_n\ n \in ED$ to account for the two communication interfaces of edge nodes).
D_{nm}	Distance between node pair (m) $n, m \in N$
NM_n	Maximum power consumption of networking at node $n \in N$ (W)
NI_n	Idle power consumption of networking at node $n \in N$ (W)
PM_n	Maximum power consumption of processing at node $n \in N$ (W)
PI_n	Idle power consumption of processing at node $n \in N$ (W)

NM^{ONU}	Maximum power consumption of ONU at an edge node (W)
NI^{ONU}	Idle power consumption of ONU at an edge node (W)
TX	The maximum transmission power consumption of wireless interface (W)
T_{nm}	Wireless transmission energy per bit over link(n,m) where $n,m \in ND \cup ED$ (W/bps)
RX	Receiver sensitivity of wireless interface (W)
R_{mn}	Wireless reception energy per bit at node n over link(m,n) where $n,m \in ND \cup ED$ (W/bps)
ϵ	Power amplifier factor for wireless communication j/b.m^2
E_n	Energy per bit of networking at node n, $n \in OLT \cup MD \cup CD \cup ED$ (W/bps)
PUE_n	Power usage effectiveness of node n $n \in N$
A, M	large constants

Table 4-2: Model Variables

TP	Total power consumption of the architecture (W)
W_n	Total power consumption at node $n \in N$ (W)
WN_n	Networking power consumption at node $n \in N$ (W)
WP_n	Processing power consumption at node $n \in N$ (W)

Ω_{sd}	The amount of processing demand of source node s served by processing node d, $s, d \in N$ (MIPS)
F_{sd}	Traffic demand between source node s and processing node d, $s, d \in N$ (Mbps)
λ_{nm}^{sd}	The amount of traffic demand between source node s and processing node d traversing link(n, m) where $s, d, n, m \in N$ (Mbps)
α_{sd}	$\alpha_{sd} = 1$ if demand of source node s is served by processing destination d, otherwise $\alpha_{sd} = 0$.
Q_s	Total number of processing nodes serving demand of source s, $s \in N$
β_n^{NET}	$\beta_n^{NET} = 1$ *if node* n *is used for networking*, $n \in N$, *otherwise* $\beta_n^{NET} = 0$
β_n^{PR}	$\beta_n^{PR} = 1$ *if node* n *is used for processing*, $n \in N$, *otherwise* $\beta_n^{PR} = 0$
β_n^{ONU}	$\beta_n^{ONU} = 1$ if ONU at node n is used, $n \in ED$, otherwise $\beta_n^{ONU} = 0$

The objective of the model is to minimise the architecture power consumption by optimising the allocation of processing resources and routing of traffic. The total power consumption (TP) of the architecture is given as:

$$TP = \sum_{n \in N} W_n \quad (4.1)$$

where

$$W_n = PUE_n \ (WN_n + WP_n \ ) \quad \forall \ n \ \in N \quad (4.2)$$

Equation (4.2) gives the total power consumption of each node in the architecture. The power consumption is composed of processing-induced part and networking-induced part. In addition, the impact of the power usage effectiveness (PUE) is accounted for at each node.

In the following we give a detailed model of the network power consumption and processing power consumption of the different nodes in the network.

For vehicular nodes:

$$WN_n = \beta_n^{NET} \ NI_n + \sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n} \lambda_{nm}^{sd} \ T_{nm} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n} \lambda_{mn}^{sd} \ R_{mn} \quad \forall \ n \in \ ND \quad (4.3)$$

Equation (4.3) gives the networking power consumption of a vehicular node as the sum of the OBU communication interface idle power consumption, the traffic-dependent, distance-dependent transmission power consumption, and the traffic-dependent reception power.

For edge nodes:

$$WN_n = \beta_n^{NET}\ NI_n + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n\cap(ND\cup ED)} \lambda_{nm}^{sd}\ \ T_{nm} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n\cap(ND\cup ED)} \lambda_{mn}^{sd}\ R_{mn}\ + \beta_n^{ONU} NI^{ONU} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n\cap(OLT)} (\lambda_{nm}^{sd} + \lambda_{mn}^{sd})E_n \quad \forall\ n\ \in ED \tag{4.4}$$

The edge node has two communication interfaces, the WiFi interface through its AP, and PON interface through the ONU. In equation (4.4), the first three terms calculate the AP (WiFi interface) power consumption, while the last two terms are for the PON interface power consumption. The PON interface power consumption is found by multiplying the traffic routed from the edge node ONU to the OLT by the energy per bit of the ONU. The idle power of the ONU is also added to the calculation.

Wireless transmission and reception energy per bit:

The traffic-dependent distance-dependent energy per bit for wireless transmission (T_{nm}) is given as

$$T_{nm} = \frac{TX}{B_{nm}} \quad + \ \epsilon * \ D_{nm}^2 \tag{4.5}$$

$$\forall\ n \in ND \cup ED, m \in Nm_n \cap (ND \cup ED)$$

The first term of Equation (4.5) gives the traffic dependent part found by dividing the transmitter maximum power consumption (TX) by the link maximum data rate. The second term gives the distance-dependent power consumption as a function of transmission distance and the power amplifier factor, which has the unit (j/b.m^2) indicating the decay in the bit energy per distance square

The reception energy per bit for wireless transmission (R_{mn}) is found by dividing the node receiver sensitivity RX by the link maximum data rate. Receiver sensitivity is defined as the minimum power required to enable reception of transmitted signal.

$$R_{mn} = \frac{RX}{B_{mn}} \qquad (4.6)$$

$$\forall\, n \in ND \cup ED, m \in Nm_n \cap (ND \cup ED)$$

The energy per bit $E_n$given as:

$$E_n = \frac{(NM_n - NI_n)}{B_n} \qquad (4.7)$$

$$\forall\, n \in OLT \cup MD \cup CD$$

$$E_n = \frac{(NM^{ONU} - NI^{ONU})}{B^{ONU}} \qquad (4.8)$$

$$\forall\, n \in ED$$

For OLT, metro, core nodes:

$$WN_n = \beta_n^{NET}\, NI_n + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n} \lambda_{mn}^{sd}\;\; E_n \qquad (4.9)$$

$$\forall\, n \in OLT \cup MD \cup CD$$

The networking power consumption for the nodes from OLT to the core node is calculated by multiplying the networking energy per bit of each node, which is calculated in equations (4.7) - (4.8) by the traffic traversing it (the full traffic demand), as shown in equation (4.9).

Processing power consumption:

The processing power consumption at any node is given in equation (4.11) considering the processing idle power consumption and the processing load dependent power consumption which is a function of the node processing efficiency (the power consumed per MIPS as shown in equation (4.10)).

$$K_n = \frac{(PM_n - PI_n)}{C_n}\;\; \forall\, n\;\in N \qquad (4.10)$$

$$WP_n = \beta_n^{PR} PI_n\; + \sum_{s\in N} \Omega_{sn}\; K_n\, \forall\, n\;\in N \qquad (4.11)$$

The model is subject to the following constraints:

$$U_s = \sum_{\substack{d \in N \\ d \neq s}} \Omega_{sd} \qquad \forall\, s \in N \qquad (4.12)$$

Constraint (4.12) states that the processing demand for a source node must be fully served by the processing destinations and the demand cannot be locally processed.

$$\sum_{\substack{s \in N \\ s \neq d}} \Omega_{sd} \leq C_d \quad \forall\, d \in N \qquad (4.13)$$

$$\Omega_{sd} \geq \alpha_{sd} \qquad \forall\, s, d \in N, s \neq d \qquad (4.14)$$

$$\Omega_{sd} \leq A\, \alpha_{sd} \qquad \forall\, s, d \in N, s \neq d \qquad (4.15)$$

Constraint (4.13) ensures that the processing demands served by a processing node do not exceed its processing capacity. A binary variable is set to indicate that a node is selected as a processing destination for a demand source as shown in constraints (4.14) and (4.15).

$$F_{sd} = V_s \alpha_{sd} \qquad \forall\, s, d \in N, s \neq d \qquad (4.16)$$

The processing demand can be served in several nodes. Each one would receive the full traffic demand from the source, as stated by constraint (4.16).

$$\sum_{m \in N m_n} \lambda_{nm}^{sd} - \sum_{m \in N m_n} \lambda_{mn}^{sd} = \begin{cases} F_{sd} & if\ n = s \\ -F_{sd} & if\ n = d \\ 0 & otherwise \end{cases} \quad (4.17)$$

$$\forall\ s, d, n \in N\ , s \neq d$$

Constraint (4.17) is a flow conservation constraint. It ensures that the amount of traffic received by an intermediate node is equal to the amount re-transmitted. It also ensures that the traffic enters and leaves the node fully at the source and destination nodes respectively.

$$\sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n} \lambda_{nm}^{sd} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n} \lambda_{mn}^{sd} \leq B_n \quad (4.18)$$

$$\forall\ n\ \in OLT\ \cup MD\ \cup CD$$

$$\sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n \cap ND} \lambda_{nm}^{sd} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n \cap ND} \lambda_{mn}^{sd} \leq B_n \quad (4.19)$$

$$\forall\ n\ \in ND$$

$$\sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n \cap ED} \lambda_{nm}^{sd} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n \cap ED} \lambda_{nm}^{sd} \leq B^{VE} \quad (4.20)$$

$$\forall\ n\ \in ND$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap(ND\cup ED)} \lambda_{nm}^{sd} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap(ND\cup ED)} \lambda_{mn}^{sd} \leq B_n \quad (4.21)$$

$$\forall\, n \in ED$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT} \lambda_{nm}^{sd} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT} \lambda_{mn}^{sd} \leq B^{ONU} \quad (4.22)$$

$$\forall\, n \in ED$$

Constraint (4.18) ensures that the maximum data rates of the OLT, metro, and core nodes is not exceeded. For vehicles, constraint (4.19) preserves the DSRC interface data rate (i.e., ensures that the interface data rate is not exceeded), which is used to communicate with vehicles. Similar constraints for the WiFi interface between vehicles and edge node are separately implemented in constraint (4.20), (4.21). Similarly, for the edge node when using the optical communication link through the ONU, the data rate is preserved through constraint (4.22).

$$Q_s = \sum_{\substack{d\in N \\ d\neq s}} \alpha_{sd} \qquad \forall\, s \in N \quad (4.23)$$

$$Q_s \leq S \qquad \forall\, s \in N \quad (4.24)$$

To benefit from the distributed processing resources, the division of the processing demand into smaller sub-tasks is allowed. Constraints (4.23) and (4.24) state the number of splits allowed.

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n} \lambda_{nm}^{sd} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n} \lambda_{mn}^{sd} \geq \beta_n^{NET} \qquad \forall\, n \in N \quad (4.25)$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n} \lambda_{nm}^{sd} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n} \lambda_{mn}^{sd} \leq A\,\beta_n^{NET} \qquad \forall\, n \in N \quad (4.26)$$

Equations (4.25) and (4.26) set a binary variable to 1 for nodes used in networking, i.e., transmit, receive, or relay nodes.

$$\sum_{\substack{s\in N\\ s\neq d}} \Omega_{sd} \leq \beta_d^{PR} \quad \forall\, d \in N \qquad (4.27)$$

$$\sum_{\substack{s\in N\\ s\neq d}} \Omega_{sd} \geq A\,\beta_d^{PR} \quad \forall\, d \in N \qquad (4.28)$$

Another binary variable is set to 1 in (4.27) and (4.28) to identify nodes used for processing.

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}\lambda_{nm}^{sd} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}\lambda_{mn}^{sd} \geq \beta_n^{ONU} \quad (4.29)$$

$$\forall\, n \in ED$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}\lambda_{nm}^{sd} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}\lambda_{mn}^{sd} \leq A\,\beta_n^{ONU} \quad (4.30)$$

$$\forall\, n \in N$$

A binary variable is set to 1 in (4.29) and (4.30) to indicate the use of the ONU in an edge.

4.3 Scenarios Studied and Results

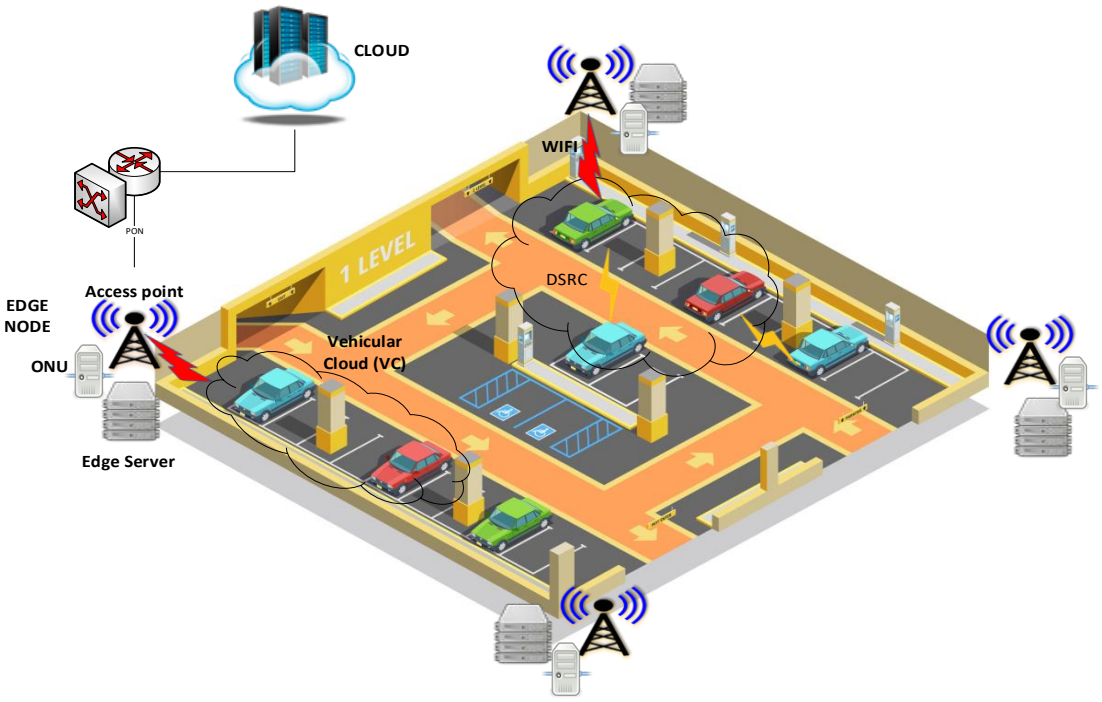

Figure 4.1: Car Park Setting

We evaluate the energy efficiency of the proposed architecture considering stationary vehicles in a parking lot. Vehicles in a parking lot offer resources for variable lengths of time, from short-term (half-hour to 3 hours) to long-term (over 3 hours to days and weeks) [156]. This accounts for and is due to vehicular mobility in and out of the car park. These resources are idle in congested business districts (e.g., employees' cars during working hours), or in urban areas with supermarkets and shopping malls, creating opportunities to exploit these resources in smart city applications.

Figure 4.1 illustrates a small parking area of 45 meters x 45 meters, accommodating up to 25 vehicles, with a standard parking space of 4.8 meters x 2.4 meters per car [157]. In our setting, we have 16 vehicles in the parking lot with the distance between two vehicles ranging from 2 meters to 24 meters. The parking lot is surrounded by 4 edge nodes, placed at the four corners of the parking area at an average distance of 30 meters away from the vehicles. Both DSRC and WiFi have communication ranges of several hundreds of meters [16, 158]. Each edge node serves a vehicular cloud of 4 vehicles.

For the demand size, we choose the minimum size of traffic demand as 2 Mbps, which gives a processing demand of 4000 MIPS. The choice is made to allow for distributed processing among vehicles (vehicle processing capacity is 3200 MIPS in this study, see Table 4-3).

To calculate parameters for the vehicles as shown in Table 4-3, the following points were considered:

- Based on [159], highly efficient intel servers execute 4 instructions per cycle. 2 instructions per cycle per core are assumed for OBUs as they not expected to carry highly intensive processing tasks.
- From [160], OBU processor has 2 cores with Speed = 800 MHz which will be used to calculate the processing capacity in MIPS. Also, Maximum power is (OBUMAX = 10 W) and the idle power is (OBUI = 5 W).
- Based on [135], a general-purpose computer spends 58% of its operational power on processing, 21% on storage (RAM and Disk), 21% on communication.
- So, for vehicles OBU, Processing Max/idle power = ((0.58 processing + 0.21 storage) * OBUMAX/OBUI).
- For networking, Networking Max/idle power = (0.21 * OBUMAX/OBUI)
- For WiFi, separate low power transceiver is added to the vehicle.
- For DSRC, modulation power (TX = +22 dBm [161] = 158 mW) and transceiver sensitivity (RX = -77 dBm [160]) values are used to find the energy per bit for transmission and reception, respectively.
- Similarly, for WiFi, modulation power (TX = +14 dBm [162] = 25 mW), and transceiver sensitivity (RX = - 72 dBm [162]) are used.
- For the vehicles PUE, the OBU is assumed to be small and efficient enough not to require ventilation system of any significant power consumption.

Table 4-3: Model Parameters for Vehicles

Notation	value
C_n	2 instructions/cycle x 2 cores x 800 MHz= 3200 MIPS [160]
PM_n	OBU = 7.9 W [160]
PI_n	OBU = 3.95 W [160]
K_n	(7.9-3.95)/3200=0.00123 W/MIPS
NM_n	2.712 W [160] [162]
NI_n	1.05007 W [160] [162]
B_n	27 Mbps [161]
B_{nm}	DSRC = 27 Mbps, WiFi = 150 Mbps
B_n^{VE}	150 Mbps [163]
ϵ	100 pj/bit.m^2 [164]
PUE_n	1

To calculate parameters for the edge nodes as shown in Table 4-4, the following points were considered:

- As stated before, the edge node is composed of access point, server, and ONU, collocated in one place.
- For a server, a raspberry Pi processor (*3 Model B*) is used.

- Based on [159], we assume 2 instructions per cycle for the raspberry Pi processor, with 4 cores of speed = 1200 MHz [165, 166].
- The power consumption of the raspberry Pi is dedicated for processing, and this assumption is used to find the processing efficiency.
- For the transmission and reception energy per bit, the transmit power (TX =28 dBm [163] = 630 mW) and reception sensitivity (RX = -104 dBm [163]) of the access point are used.
- The idle power of an edge node = sum of idle power of the three devices (Raspberry + AP + ONU). Using ON/Off power profile so only the component used is included the calculation.
- The three devices of the edge node are collocated in one place, but they are not contained or boxed, which provides natural cooling and ventilation and the PUE can be set to 1.
- The devices of the edge node are integrated together, so the inter-communication between them is ignored in this work.

Table 4-4: Model Parameters for Edge Nodes

Notation	value	
C_n	Rasberry Pi	2 instructions/cycle x 4 cores x 1200 MHz = 9600 MIPS [166]
PM_n	Rasberry Pi	12.5 W [166]
PI_n	Rasberry Pi	2 W [166]
K_n	(12.5-2) /9600 = 0.0011 W/MIPS	
NM_n	Access Point	25 W [163]
NI_n	Access Point	5.5 W [163]
B_n	150 Mbps [163]	
B_{nm}	150 Mbps	
NM_n^{ONU}	15 W [167]	
NI_n^{ONU}	13.5 W [167]	
B_n^{ONU}	10 Gbps[167]	
PUE_n	1	

To calculate parameters for the cloud as shown in Table 4-5, the following points were considered:

- Based on [159], the server assumed to run 4 instructions per cycle (highly efficient)

- The power consumption of the cloud is dedicated for processing, and this assumption is used to find the processing efficiency.
- Transmission power of cloud is ignored as it only receives data, the return of the result (small) to the source node is not considered.
- The cloud server idle power is set as 50% of the maximum. The idle power consumption of the cloud server is much higher than that of the distributed vehicles and edge nodes, it is expected to contribute significantly to the processing allocations decisions. Even as cloud servers improves by consuming less idle power, the distributed allocation can be expected to serve low and medium demand volumes, especially since the efficiency of vehicles OBU and edge node are expected to improve as well.

Table 4-5: Model Parameters for Cloud

Notation	value
C_n	4 instructions/cycle x 10 cores x 2.8 GHz = 112000 MIPS [168]
PM_n	115 W
PI_n	57 W (50% of maximum)
K_n	(115--57)/112000=0.000518 W/MIPS
PUE_n	1.1 [116]

To calculate parameters for metro and core routers and switches, as shown in Table 4-6, the following points were considered:

- The values for the PUE in the network devices (routers and switches) are derived from [116]
- For switches, it is assumed the devices become more power hungry as they get closer to the core and datacentre. For example core switches are in the backbone of the network and deal with multiple aggregations and require more complicated functionalities for security, fault tolerance, and networking. Therefore, the aggregation switches are set to consume the typical power values in [169] and the cloud switches consume the typical power value stated in the datasheet [169].
- For the core and cloud routers, the port power consumption is estimated from [170], by diving the total power consumption (1450 W) by the number of ports (48 ports) as all ports have the same capacity.
- For the aggregation router, the port power consumption is estimated from [170], by dividing the maximum power consumption (420 W) by the maximum throughput (800 Gbps) to get the W/Gbps, which was then multiplied by the port capacity of 10 Gbps.
- The processing capacity of OLT, routers and switches in metro and core is set to zero.

Table 4-6: Network Devices Parameters

Device Type	NM_n (W)	NI_n (W)	NI_n $(\%)$	B_n $(Gbps)$	E_n $(\frac{nj}{bit})$	PUE_n
OLT [171, 172]	1940	60	0.546	8600	0.219	1.5 [173]
Agg. Switch	210	189	1.72	6*40 = 240	0.088	1.5
Agg. Router port	5.25	4.725	0.043	10	0.053	1.5
Core Router port	30	27	0.246	40	0.075	1.5
Cloud router port	30	27	0.246	40	0.075	1.5
Cloud Switch	470	423	3.85	6*100=600	0.078	1.5

We evaluate 4 scenarios of processing resources availability. In the first scenario requests can only be processed in vehicles (V scenario). The second scenario optimises the allocation of the processing request at vehicles and edge nodes only (VE scenario). In the third scenario, only conventional cloud has processing capacity (C scenario). This is the scenario used as benchmark for comparison. The last scenario optimises the allocation of processing resources at the three processing layers (VEC scenario).

4.3.1 Demand Size Variation

We consider a single vehicle to generate the demand. The processing demand is varied between 4000-60000 MIPS to reflect tasks with low processing requirements to high processing requirements. A processing demand can be split among any number of processing destinations, i.e., S is set to the maximum number of available processing nodes in the MILP model.

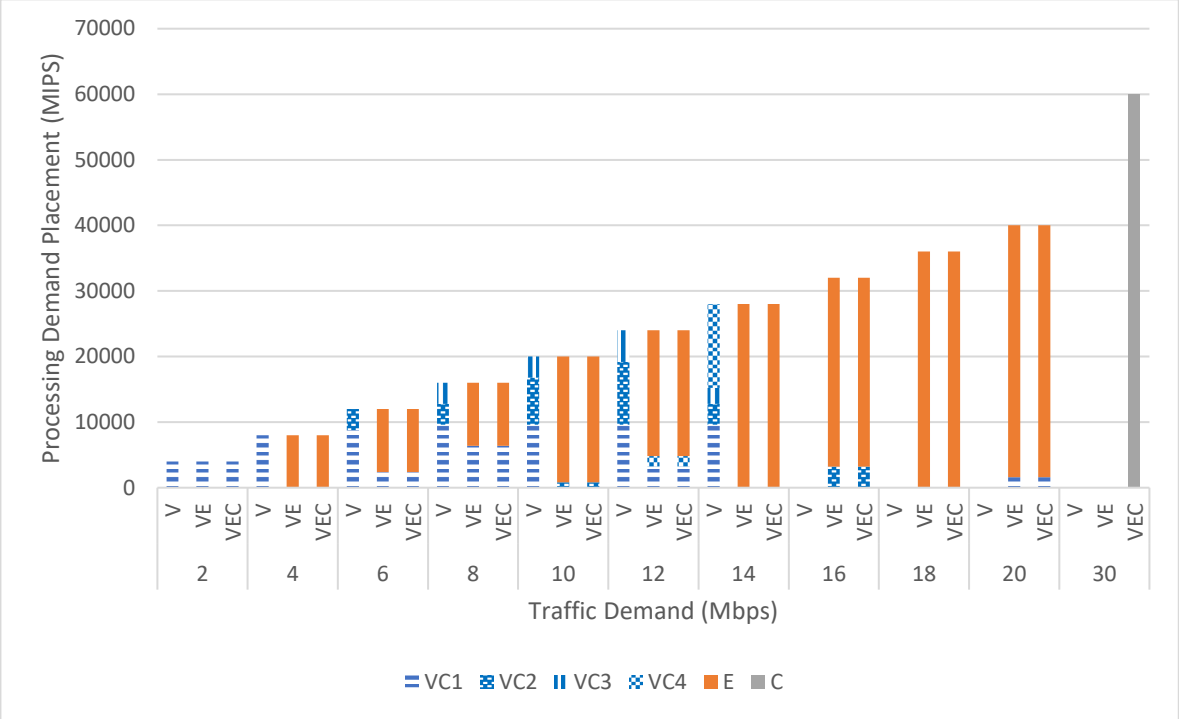

Figure 4.2: Processing Demand placement when serving a single demand considering the different processing scenarios

Figure 4.2 shows the processing demand placement in the three processing layers in each processing scenario. For the V scenario, the model saturates the vehicles in the same vehicular cloud of the vehicle generating the demand (VC1) before moving to other VC. In the VE and VEC scenarios, edge nodes are preferred because of their higher processing capacity and efficiency, so one edge node is packed for example, before activating more vehicular processing nodes as the demand grows larger in size.

Figure 4.3 shows the distribution of the architecture total power consumption between networking and processing. It clearly shows that processing is the dominating contributor to the power consumption. Figure 4.3 shows the merits of having the edge nodes as supporting processing resources. The VE scenario

matches the optimal solution given by the VEC scenario, which emphasises the optimal use of the vehicles and edge nodes in the presence of the cloud as third location for processing. The only exception of the results is for the last demand where the optimal solution for the VEC scenario is to serve the request in the cloud due to capacity limits of the vehicle and edge nodes. Further inspection of Figure 4.3 shows that for the V scenario, the networking power consumption started to increase significantly as the demand increases, until the total power exceeds the power consumed in the conventional cloud scenario. Also, it is worth mentioning that for the larger demands not served in the V and VE scenarios, the bottleneck is the networking capacity and not the processing capacity, e.g. for the largest traffic demand of 30 Mbps, the associated processing demand is 60000 MIPS while the total capacity of vehicles and edge nodes is 86400 MIPS. Therefore, increasing the bandwidth capacity can go a long way in improving the performance and power saving of the architecture.

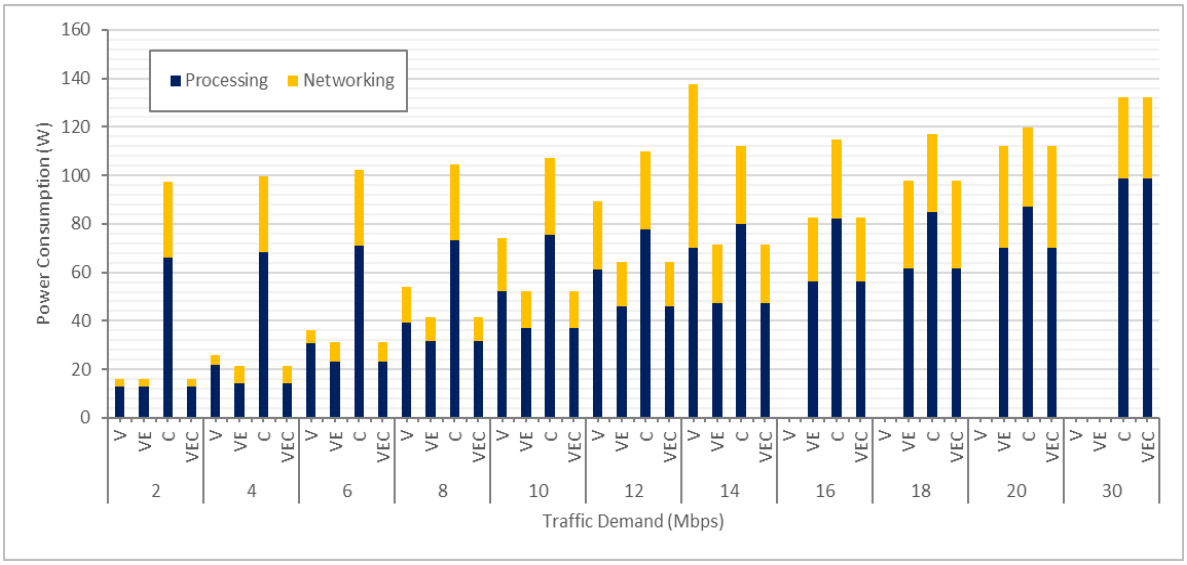

Figure 4.3: Total power consumption when serving a single demand considering the different processing scenarios

In Figure 4.4, we break down the networking power consumption. Figure 4.4 shows a surge in the networking power consumption of the V scenario serving a traffic demand of 8 Mbps. This is because 5 processing vehicles are needed to fully serve the demand (8 Mbps is associated with 16000 MIPS). This generates a total traffic of 40 Mbps to be transmitted from source to the processing destinations, which exceeds the capacity of the DSRC and necessitate the use of the WiFi and therefore leads to the increase seen in the power consumption. The other surge of the V scenario at 14 Mbps is due to edge nodes communicating through the PON access networks as the edge nodes WiFi APs cannot support the total traffic generated from this demand (14 Mbps is associated with 28000 MIPS which requires 9 vehicles and a total of 126 Mbps uplink + 126 Mbps downlink).

Figure 4.5 breaks down the processing-induced power consumption. For the VEC scenario, the vehicle OBU is optimal as long as the total idle power of processing vehicles is lower than that of a single edge node. For example, at 2 Mbps (4000 MIPS), it is optimal to split the demand between two vehicles, while for the 4 Mbps demand (8000 MIPS), activating one edge server is more efficient than activating 3 vehicles OBUs. A combination of OBUs and edge node servers resulting in minimum power consumption is activated to serve higher processing demands.

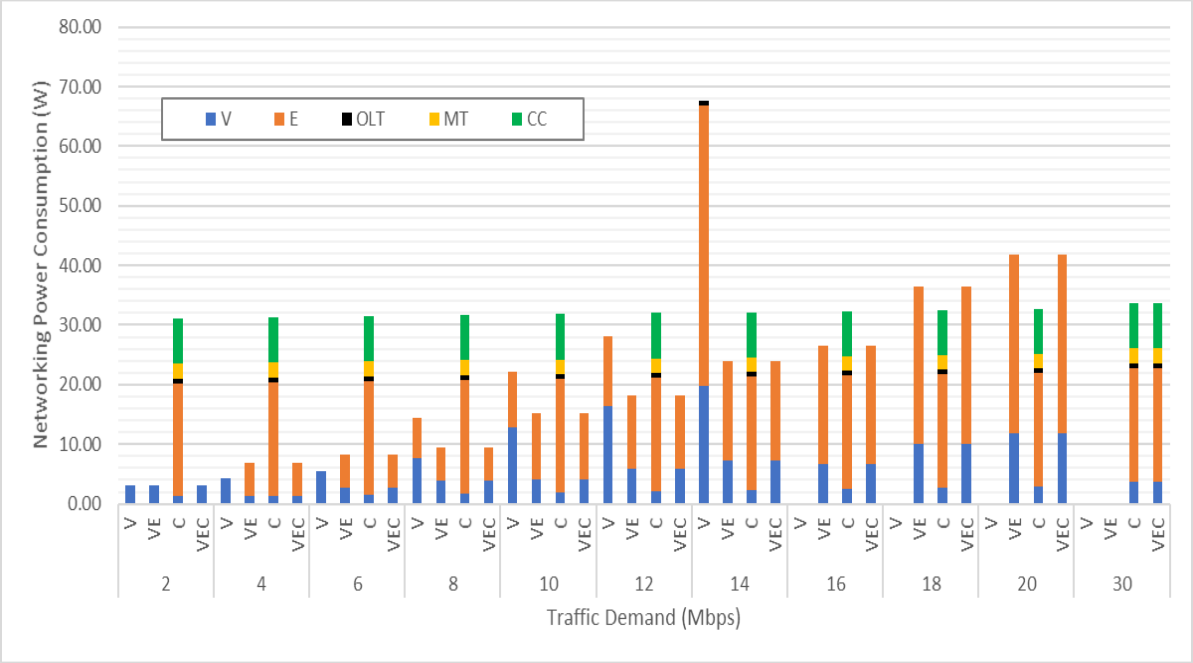

Figure 4.4: Networking power consumption when serving a single demand considering the different processing scenarios

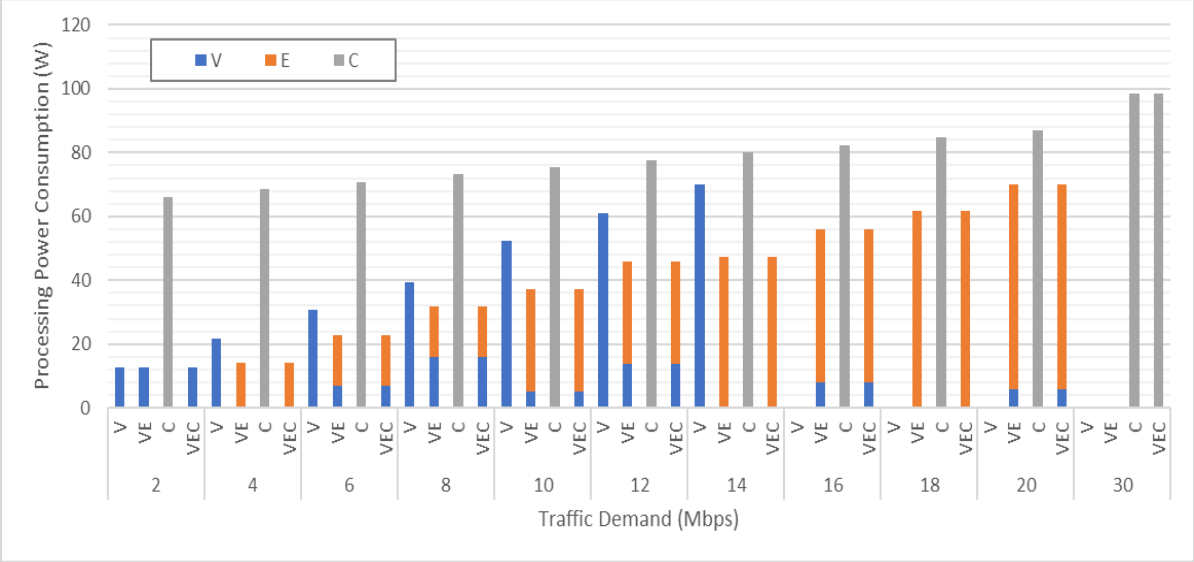

Figure 4.5: Processing power consumption when serving a single demand considering the different processing scenarios

Figure 4.6 shows the power savings of the three distributed processing scenarios (V, VE and VEC) in comparison with processing in the conventional cloud. Optimised processing in the VEC scenario resulted in power savings up to 84% compared to processing in the cloud. As mentioned before, the power savings for the VE and BEC scenarios are identical since they led identical results with preference of vehicles and edge nodes over cloud.

Limiting processing to the vehicles and edge nodes (VE scenario) gave the maximum power savings for traffic demands as high as 20 Mbps. The energy efficiency of the vehicular only processing decreases as the size of the demand increases. Processing a demand of 14 Mbps in the vehicular cloud proves to be less efficient than cloud processing by 23%.

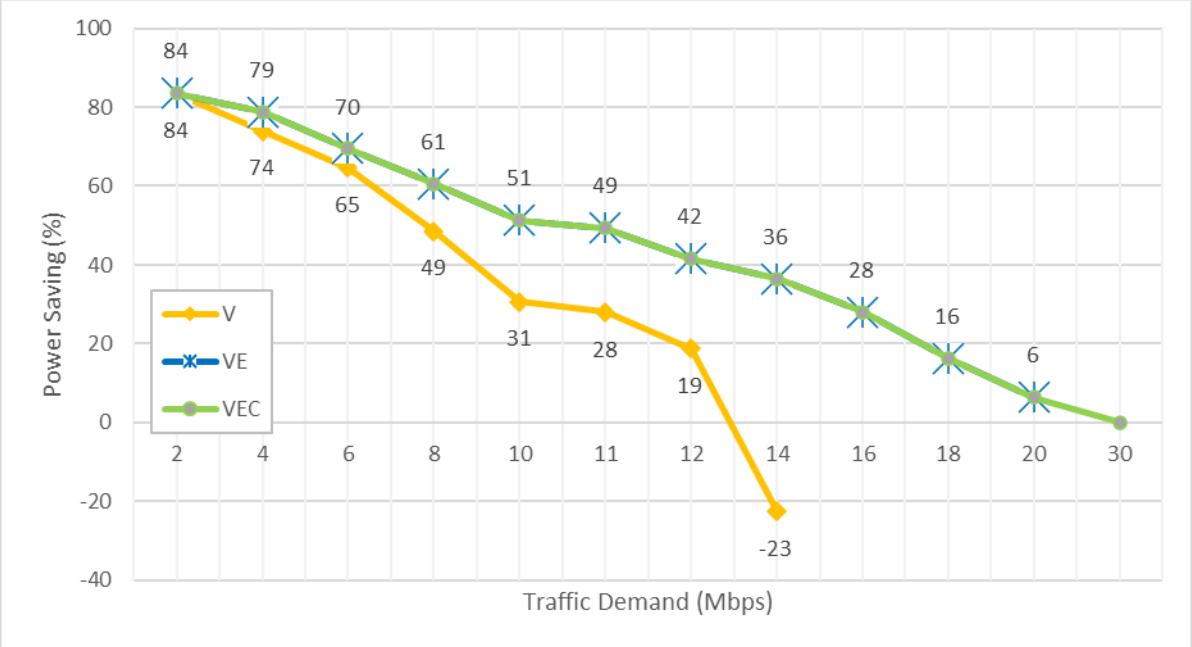

Figure 4.6: Power saving of the three distributed processing scenarios (V, VE, VEC) in comparison with conventional cloud (single demand)

4.3.2 Processing Demand Splitting Limitations

Processing demands splitting can improve processing resources utilisation and consequently the energy efficiency of serving demands. It can also reduce the total processing time and avoid exhaustion of computational resources. In this subsection we study the impact of limiting the number of processing nodes that can serve the request as opposed to the unlimited processing demand splits studied in the previous subsection.

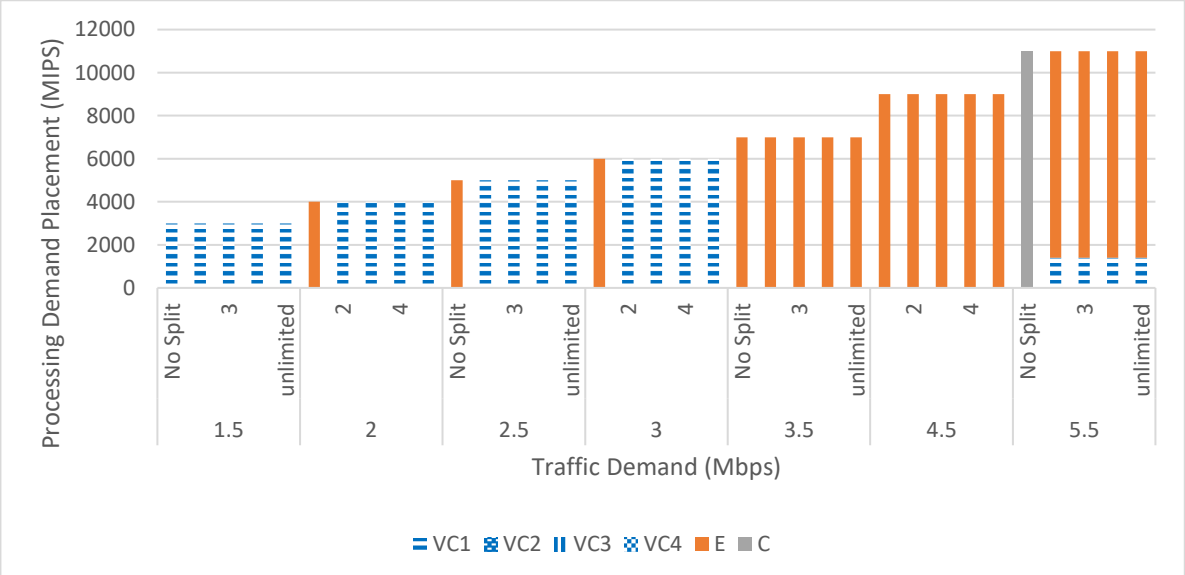

Figure 4.7: Processing demand placement of the VEC scenarios with varying splits limits

Figure 4.7 shows the processing placement. For smaller demand size, processing allocation was optimally done in VC. The limitations in splitting have more impact as the demand increases. No splitting for demand sizes 2.5 and 3 Mbps forced the allocation to the edge node, while allowing splitting allowed the use of the VC. However, the flexibility of splitting does not necessarily make it the

optimal option, as the model continued to optimally allocate the demand 3.5-5.5 Mbps fully to an edge node without splitting.

Figure 4.8 shows the total power consumption of the VEC scenario under different splitting limits. It shows that a splitting limit of 2 is enough to achieve the minimum power consumption in this case. Splitting a request of 5.5 Mbps traffic demand between two processing destinations in a VEC scenario improves the energy efficiency by 71% compared to processing without splitting which results in processing the request in the cloud as seen in Figure 4.9.

From Figure 4.9, we can also infer that the usability of V scenario is limited when no splitting was allowed. Splitting traffic demands of 2-3 Mbps allows them to be processed in the vehicular cloud which slightly increases the processing power consumption, as seen in Figure 4.9, due to the lower processing efficiency of the OBUs. On the other hand, avoiding communicating with the edge nodes by processing in the vehicular cloud significantly reduces the networking power consumption as seen in Figure 4.10 although traffic is replicated to the two vehicular processing destinations serving the demand. The overall power reduction from the no splits case is 2-3%.

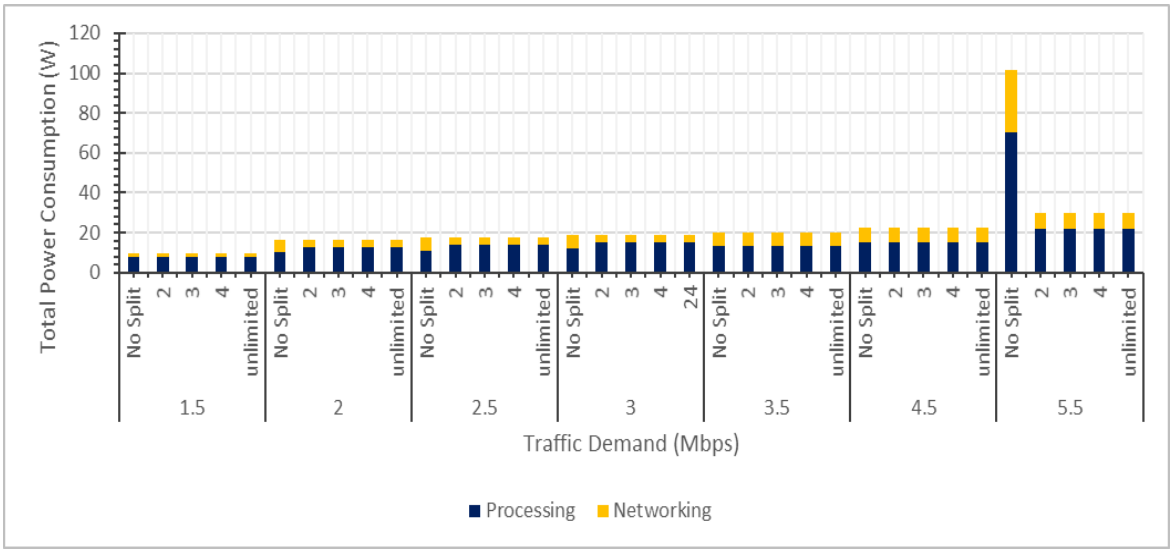

Figure 4.8: Total power consumption of the VEC scenarios with varying splits limits

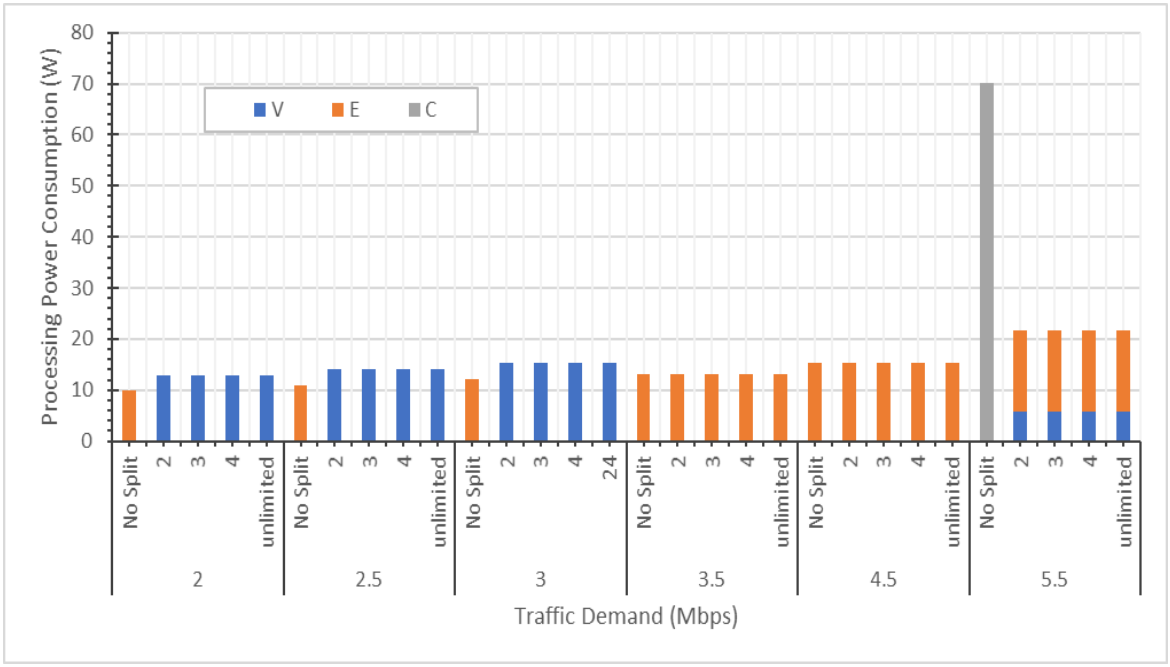

Figure 4.9: Processing power consumption of the VEC scenarios with varying splits limits

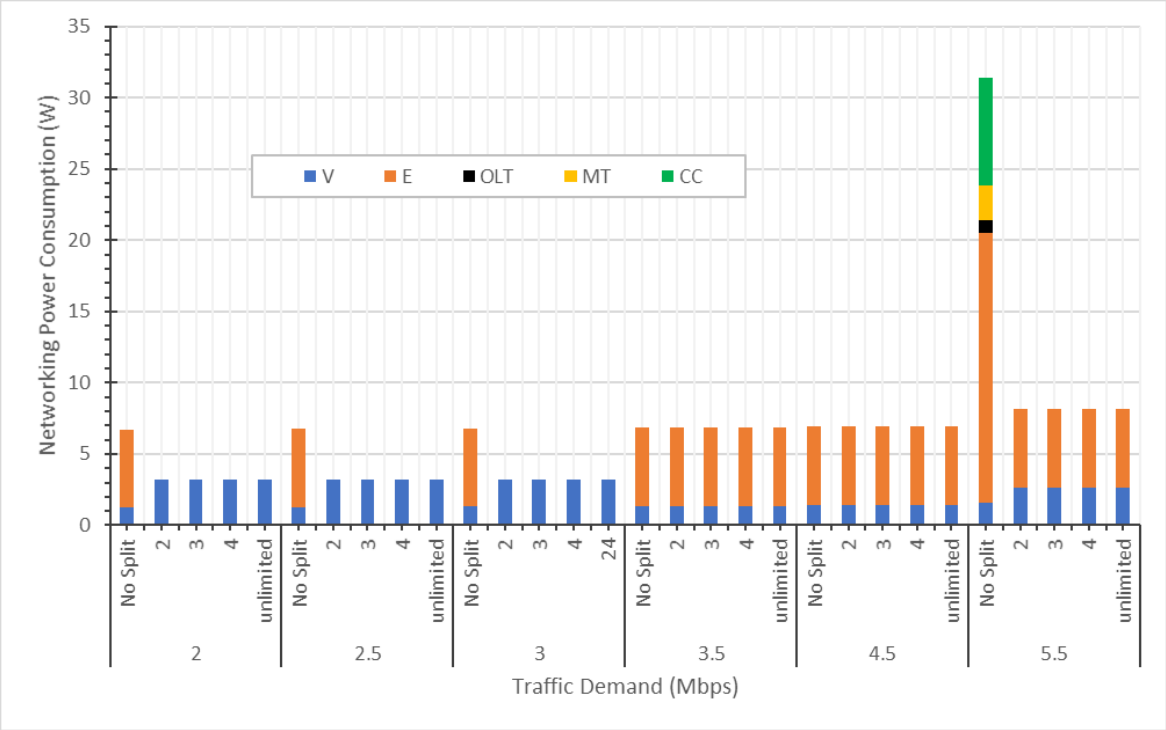

Figure 4.10: Networking power consumption of the VEC scenarios with varying splits limits

4.3.3 Proportional Traffic Assignment

In all the previous subsections, we assumed that all processing destinations receive the traffic demand in full, even when serving part of the processing demand. This limits the efficiency of processing demands splitting as it burdens the network. However, for some types of applications, a processing node serving part of the processing demand requires access to only part of the data to be processed. An example of this can be an application processing multiple images or multiple videos as discussed earlier. In this subsection we optimise the processing of a demand with traffic that can be proportionally split among the nodes serving the demand, referred to as proportional traffic (PT) demand. This test case is compared to the one in Section 4.3.1, in which the full traffic (FT) demand is delivered to every processing destination.

The MILP model is updated to represent PT requests, by replacing constraint (4.16) with the following equation:

$$F_{sd} = V_s \frac{\Omega_{sd}}{U_s} \qquad \forall\, s, d \in N\,, s \neq d \quad (4.31)$$

Equation (4.31) ensures that the traffic delivered to a processing node from a source node is proportional to the processing carried by the processing node. For example, a demand of 2 Mbps and 4000 MIPS can be split between two vehicles as follows: The first vehicles receive 3200 MIPS and accordingly (2 * 3200/800 = 1.6 Mbps) of the traffic, while the second vehicle receives 400 MIPS and 0.4 Mbps of the traffic.

The comparison considers a single demand of varying sizes for the 3 scenarios (V, VE, VEC), the C scenario is unaffected by this change as the whole demand is sent to the cloud.

Proportionally splitting the traffic relieves the V scenario traffic bottleneck observed for FT for traffic demands higher than 14 Mbps in subsection 4.3.1. Comparison to Figure 4.3 for serving with full traffic (FT) shows that some cases in the V scenario that were infeasible before due to networking capacity limitations, are served in this modified case.

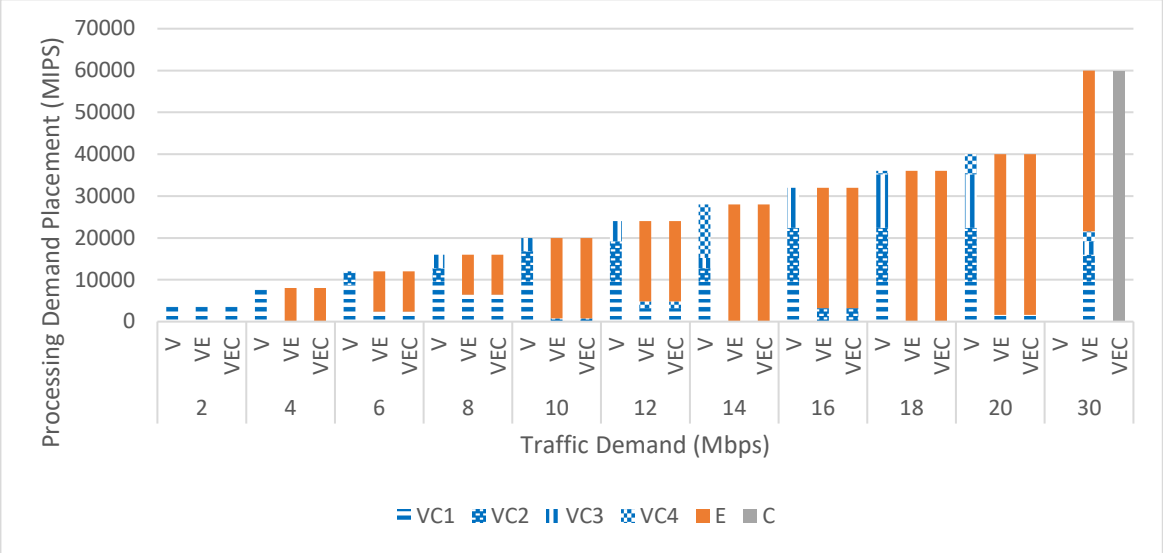

Figure 4.11: Processing demand placement considering the three processing scenarios (V, VE, VEC) with proportional traffic (PT)

Figure 4.11 shows the processing placement in each layer and it shows that in the FT case, the V scenario packed one VC before moving to the next one, and it was able to serve bigger demands.

Figure 4.12 and Figure 4.13 compare the FT and PT cases under the V scenario in terms of processing and networking, respectively. Proportionally splitting the traffic among the processing destinations has not changed the number of processing destinations compared to the FT case. However, the networking power consumption was reduced. It improved the utilisation of the DSRC communication bandwidth and therefore reduced the networking power consumption as the WiFi interface and the PON network are used less. Proportionally splitting traffic also relieves the traffic bottleneck observed for FT case for traffic demands higher than 14 Mbps. Similar trends are observed for the VE scenario as shown in Figure 4.14 and Figure 4.14. Improved utilisation of the

network bandwidth under the PT case allowed the VE scenario to serve 30 Mbps demands.

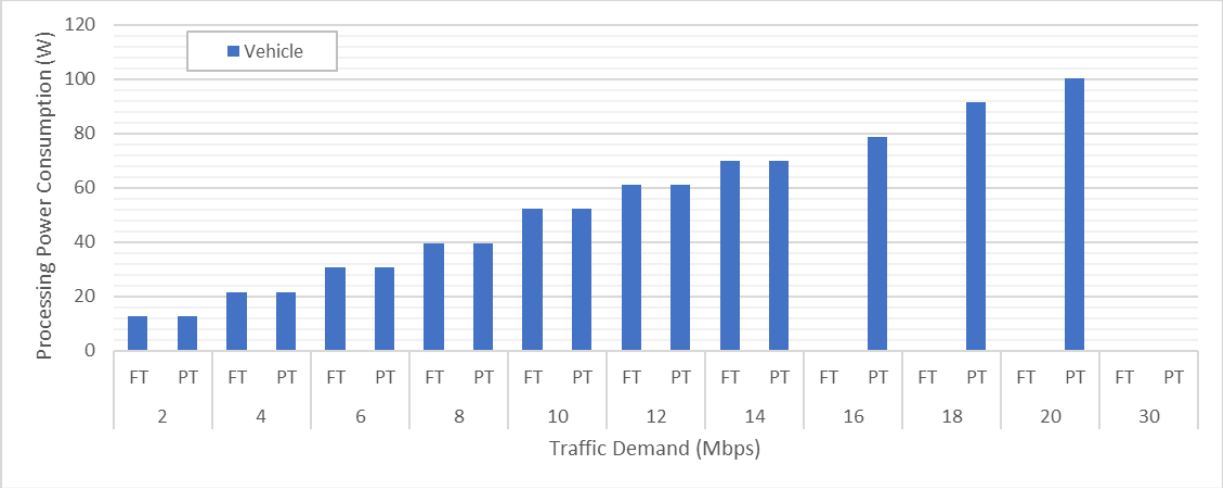

Figure 4.12: Processing power consumption of the V scenario considering full traffic (FT) and proportional traffic (PT)

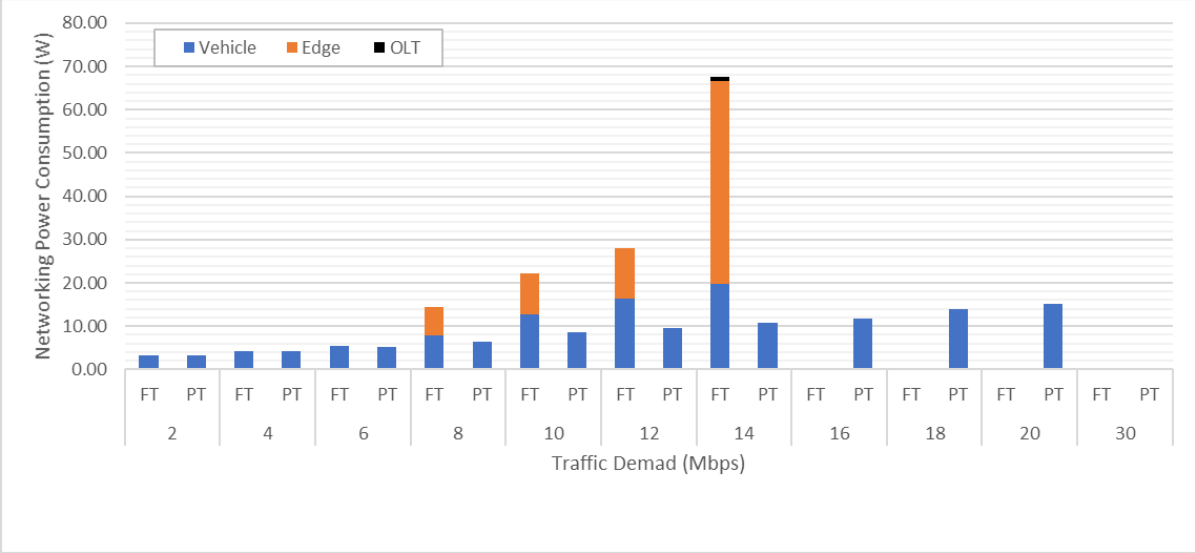

Figure 4.13: Networking power consumption of the V scenario considering full traffic (FT) and proportional traffic (PT)

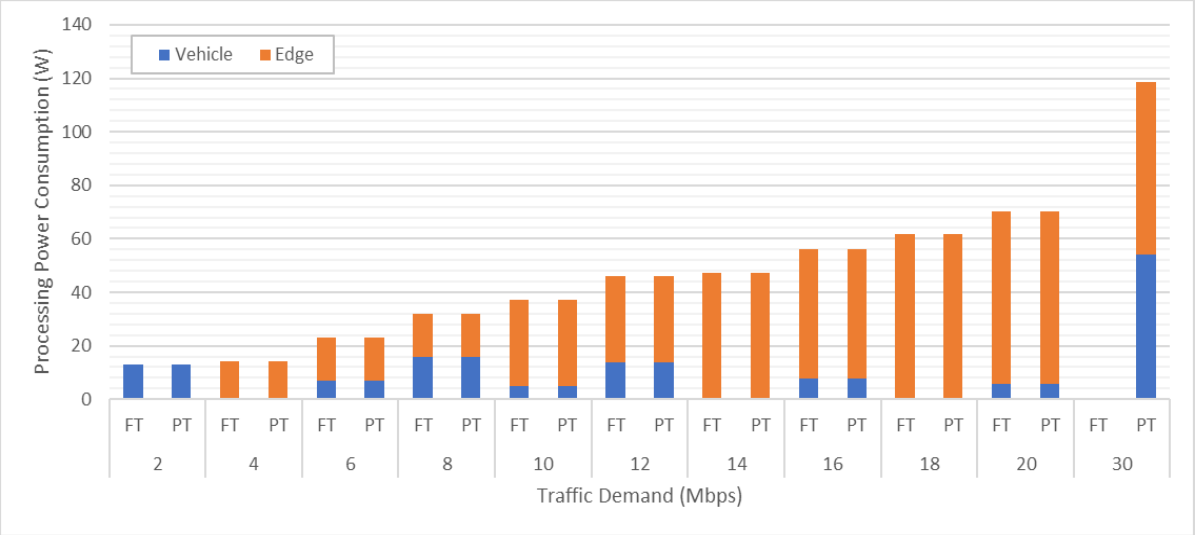

Figure 4.14: Processing power consumption of the VE scenario considering full traffic (FT) and proportional traffic (PT)

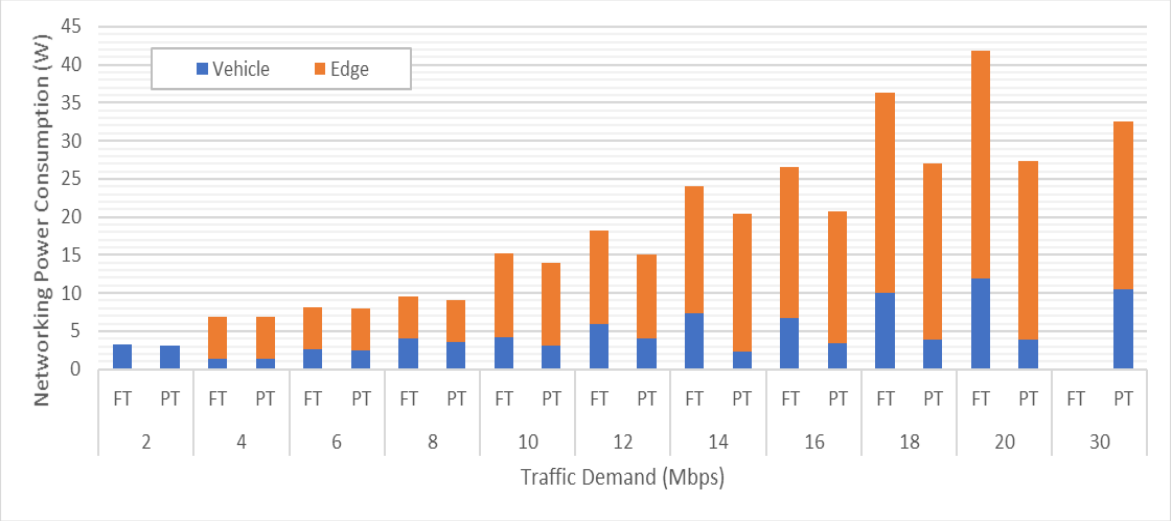

Figure 4.15: Networking power consumption of the VE scenario considering full traffic (FT) and proportional traffic (PT)

Comparing the PT and FT cases under the VEC scenarios in Figure 4.16 and Figure 4.17 confirms that the cloud is the optimal processing destination for the 30 Mbps demand even when the demand traffic resulting from distributed processing in the vehicular and edge layers can be supported by the network. This is due to the energy efficiency of the cloud in processing such a large

demand compared to processing in the vehicular cloud and edge nodes which requires activating multiple processing destinations.

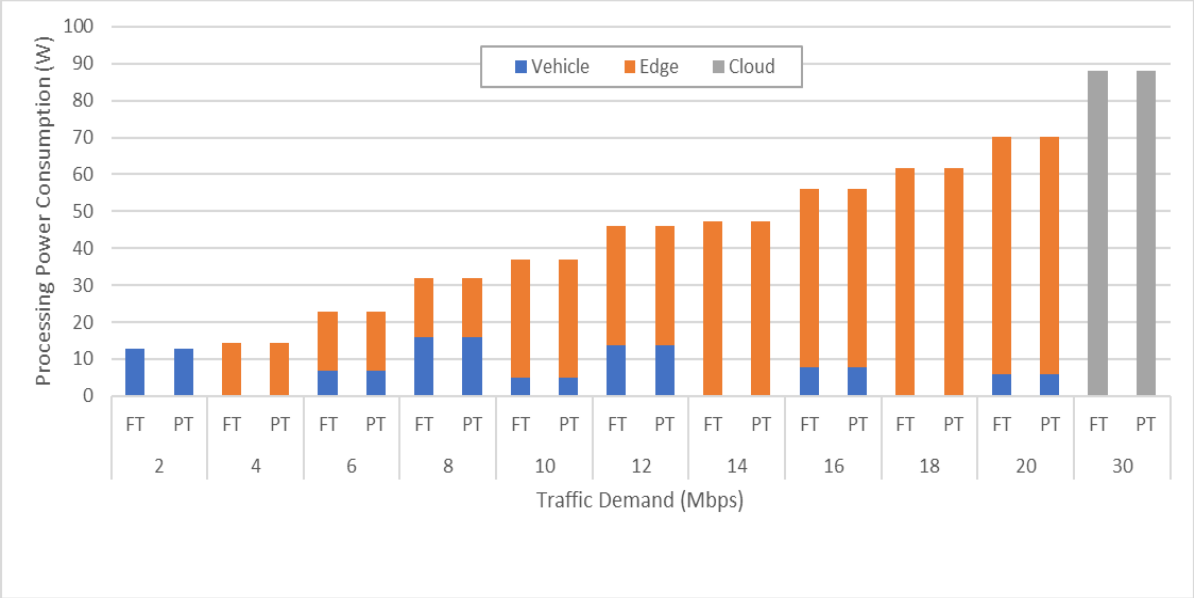

Figure 4.16: Processing power consumption of the VEC scenario considering full traffic (FT) and proportional traffic (PT)

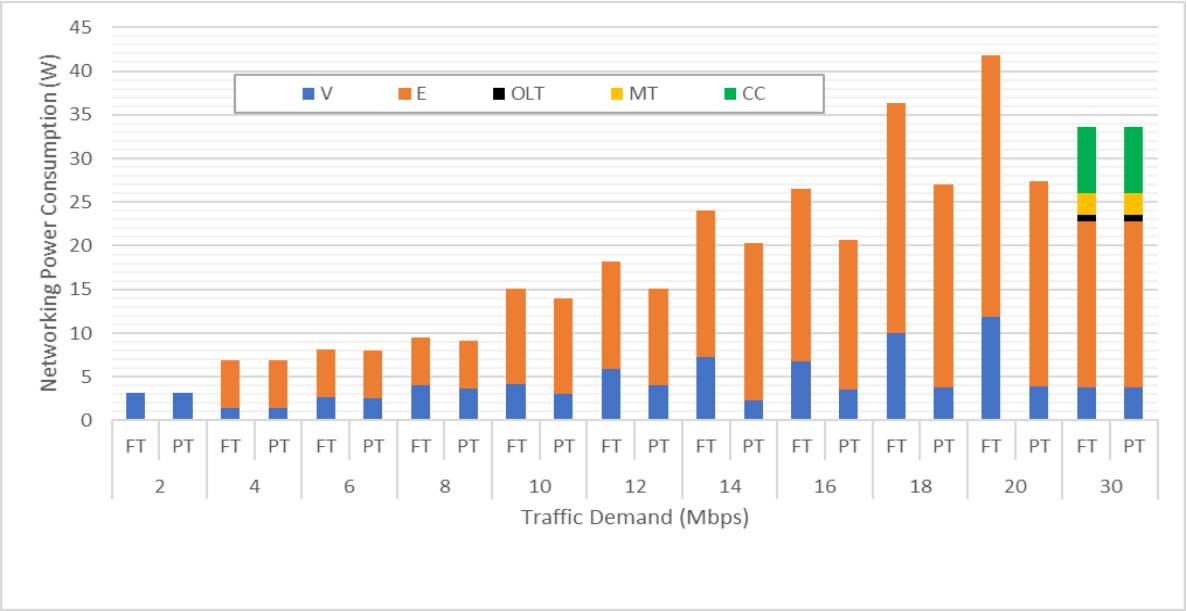

Figure 4.17: Networking power consumption of the VEC scenario considering full traffic (FT) and proportional traffic (PT)

Proportional traffic impacts the power savings achieved by the different processing scenarios compared to processing in the conventional cloud, as seen in Figure 4.18. Proportionally splitting the traffic improves the energy efficiency of the V scenario compared to the FT case making processing demands as high as 20 Mbps in the vehicular cloud more efficient than cloud processing. As mentioned above, under the PT case the VE scenario can process demands as high as 30 Mbps. This is, however, less efficient than processing in the cloud by 14%. For the VEC scenario, the power savings improved from 6% for the FT case to 19% for the PT case for a demand of 20 Mbps compared to processing in the cloud.

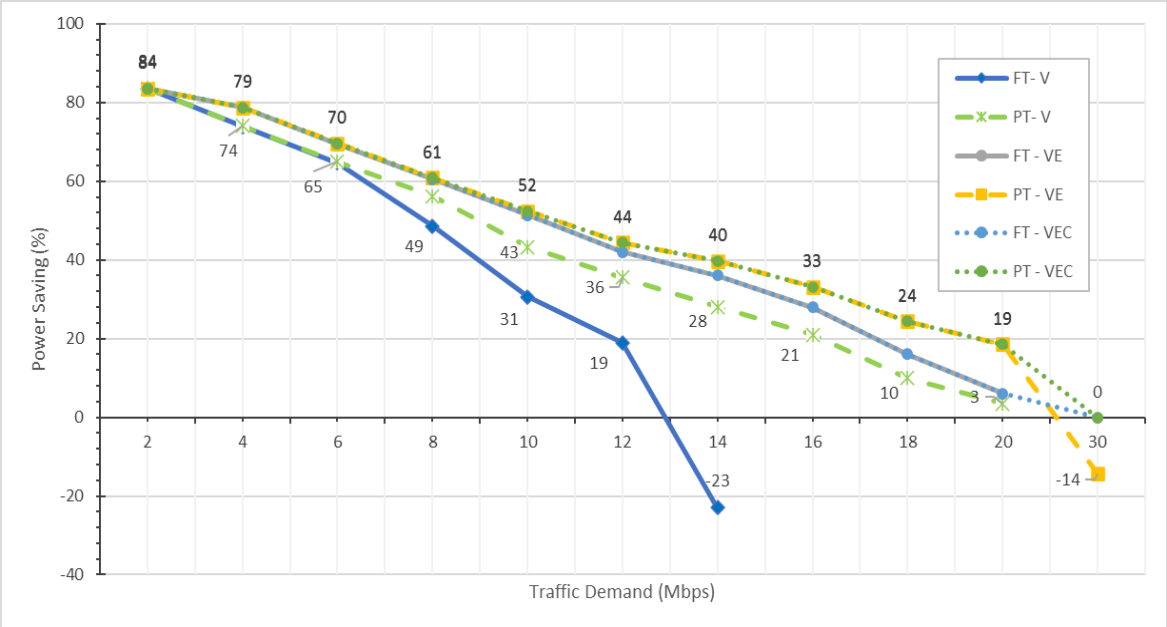

Figure 4.18: Power saving of the (V, VE, VEC) FT and PT cases in comparison to the C scenario

4.3.4 Multiple Demands Service

In this section we examine multiple requests competing for the available resources under the VEC scenario considering full traffic replication to all processing destinations. Relative to the processing resources in vehicles and edge nodes, three demand profiles are examined: low (Traffic 1 Mbps, Processing 2000 MIPS), medium (Traffic 3 Mbps, Processing 6000 MIPS), and high (Traffic 5 Mbps, Processing 10000 MIPS).

The low demand can be served in single vehicle, medium demand can be served in single edge node, and the high demand exceeds the capacity of a single edge node. Figure 4.19 shows the processing demand placements as the demands grow in size/number, and it shows a tendency to use edge nodes over vehicles. Figure 4.20 evaluates the total power consumption of up to 10 co-existing requests and illustrates the impact of varying demand size/number on the processing and networking power consumption. With the growing requirements, it becomes less costly to operate the cloud as the difference between the high idle power consumption of the cloud server and the idle power consumption of the multiple vehicles and edge nodes required to serve the demands decreases. Also, for higher demand sizes the increase in the networking requirements cannot be accommodated with the use of the vehicles and edge only. The power savings of the VEC scenario rapidly decreases as the demands increase in size/number, as shown in Figure 4.21. For medium and high demands as the increase in the number of demands forced the use of the cloud, the power savings drops to 0%.

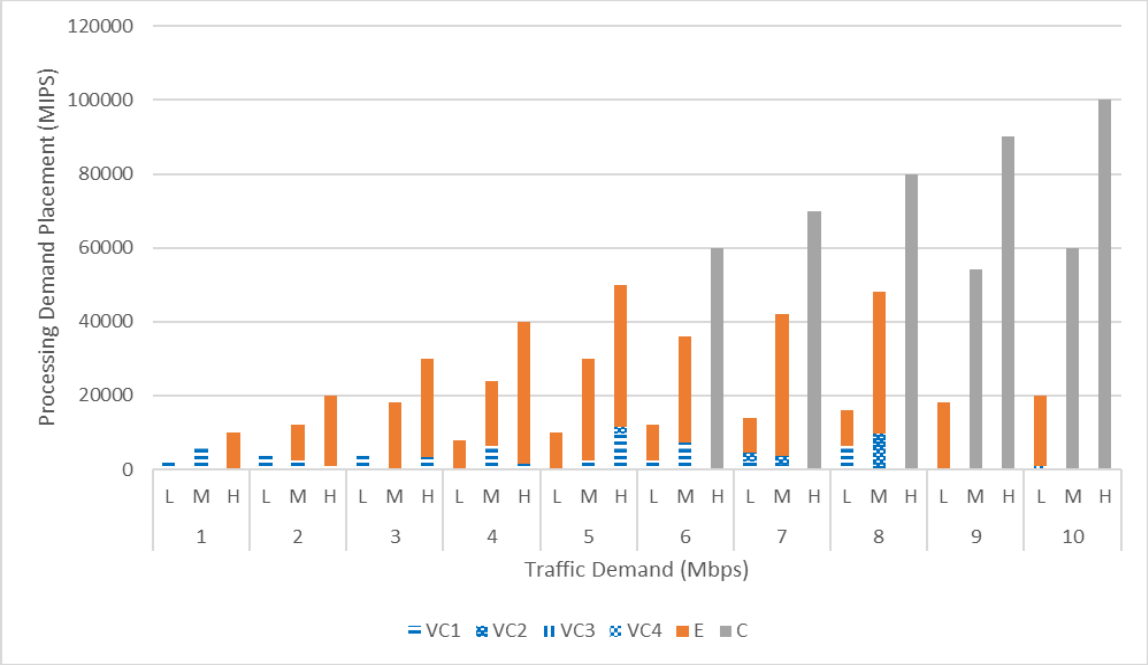

Figure 4.19: Processing demand placement when serving multiple demands of varying sizes considering the VEC processing scenario

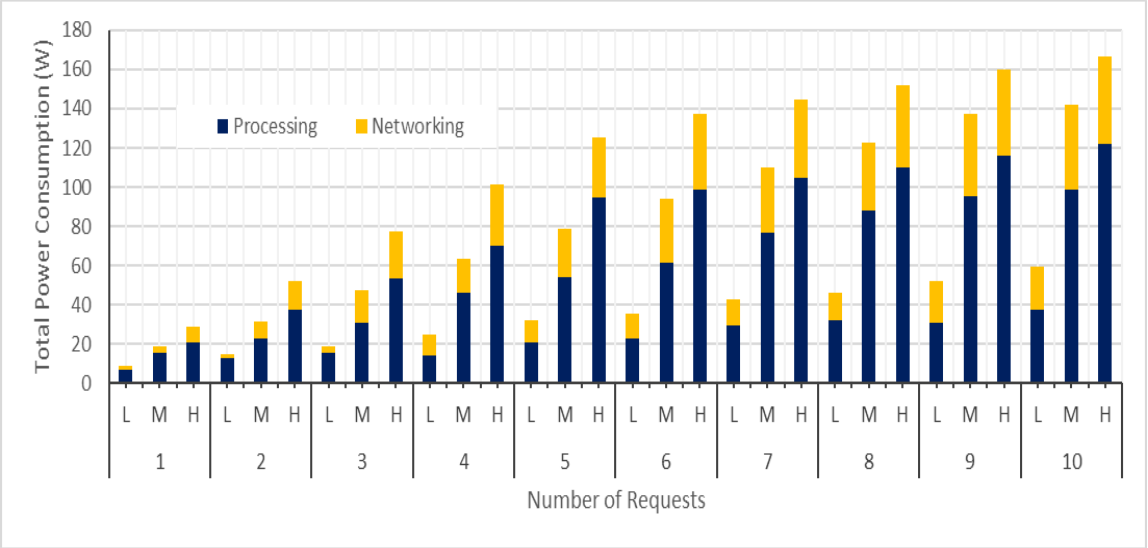

Figure 4.20: Total power consumption when serving multiple demands of varying sizes considering the VEC processing scenario

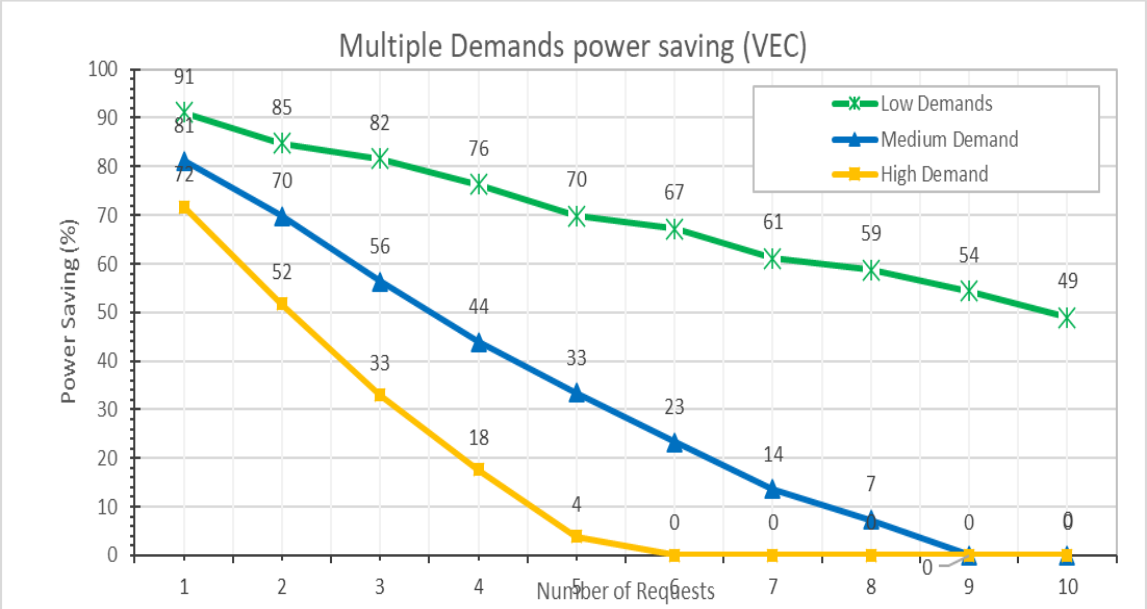

Figure 4.21: Power Saving when serving multiple demands of varying sizes considering the VEC processing scenario

4.4 Energy Efficient Demand Allocation Heuristics in Vehicular Cloud Architecture

A heuristic is developed based on insights obtained from the model to allocate processing demands in real time. The heuristic flowchart is shown in Figure 4.22. The heuristic serves demands in descending order of their processing requirements as the processing power consumption dominates the power consumed to serve demands. For each demand, the processing nodes are sorted based on the criteria defined in equation (4.32), with the *most fit* candidate in the beginning of the list and the *least fit* at the end. Sorting of the processing nodes can be ascending or descending depending on the demand size in comparison with the capacities of the distributed processing nodes. Also, the counter (*Trial)* is set to 1. This variable is set to give each demand two attempts to find a processing destination(s) by going over the complete list of processing nodes candidates and trying to route over them.

$$SortingCriteria_{sd} = NPower_{sd} + PrPower_{sd} + Idle_d PUE_d \quad (4.32)$$

$$\forall s, d \in N$$

where:

$NPower_{sd}$ is the power consumption of routing traffic between source s and processing destination d over the minimum hops route.

$PrPower_{sd}$ is the processing power consumption of serving processing demand of source node s in processing destination d

The heuristic selects the most fit candidate from the sorted list and tries to route the traffic demand over the minimum hop route if the candidate has available processing resources. If the minimum hop is not available due to bandwidth capacity, the heuristic removes the link of limited capacity from valid routes. The heuristics then selects the next node in the sorted list and tries to route the traffic demand on the minimum hop route. The heuristic examines all the nodes in the sorted list until all the demand under consideration is served. If all nodes are examined but the demand is not fully served, the trial counter is incremented by 1 and the heuristic examines nodes in the sorted list again. Note that the availability of the minimum hop routes will change in the second attempt, giving the processing nodes that were skipped before due to bandwidth capacity limitations a new chance in terms of the least hops route, and the processing node might be used to serve. Each demand is allowed only two attempts (more attempts can be allowed, at the cost of increased complexity) of examining the nodes in the ordered list to select a processing destination(s). Also, the number

of processing nodes used is counted to ensure they do not exceed the allowed split limitation. Furthermore, a processing node is packed before moving to the next node.

If the demand is not served after the two attempts, it will be blocked and all the resources that were used to partially serve it are released to be used by other demands. The heuristic then moves to the next demand on the demands list and repeats the above procedure. After completing all the demands list, the total power consumption resulting from routing all demands is calculated.

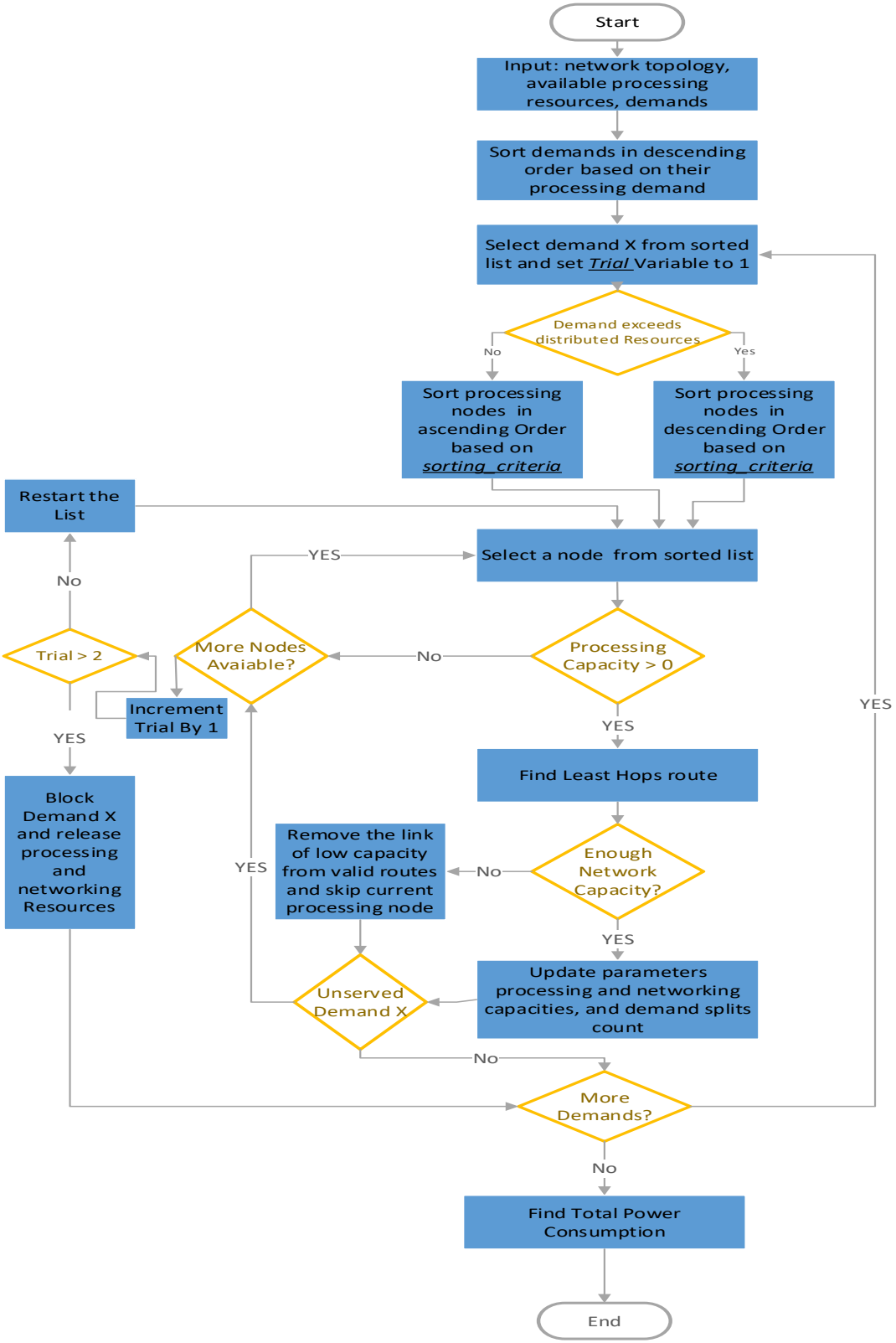

Figure 4.22: Power Optimisation Heuristics Flowchart

4.4.1 Demand Size Variation

Figure 4.23 shows a comparison between the total power consumption of the heuristic and the MILP model results in Section 4.3.1 when considering a single request of varying size in the VEC scenario. Table 4-7 shows the gap in energy efficiency between the two. The heuristic approaches the model with a gap of 0%-15% for most of the demand sizes. This gap is a result of the heuristic sub-optimal selection of processing nodes, as seen in Figure 4.24, resulting from the sequential allocation of the processing based on the current status with no knowledge of the upcoming demands. For the heuristics there is a pattern of depleting the resources of the lower processing layers before allocating demands to higher processing layers. On the other hand, the MILP model allocates demands to edge nodes or a combination of vehicular and edge nodes even when vehicular nodes have more available resources.

Table 4-7: MILP vs Heuristics Power Consumption Difference for Demand Size Variation

Traffic Demand (Mbps)	Diff (%)	Traffic Demand (Mbps)	Diff (%)
2	0	14	11
4	23	16	3
6	16	18	11
8	13	20	3
10	14	30	0
12	0		

The 30 Mbps demand is optimally processed in the cloud, both in the model and the Heuristics. In this case, the traffic demand exceeded the capacity available in the DSRC, which led to the list of candidates to be sorted in descending order, as stated in the flowchart. This arrangement pushed the cloud node to the front of the sorted list, and it was chosen as destination to serve the demand. If the processing nodes were sorted ascendingly, the cloud would have been at the end of the candidate list. As the heuristic would distribute the demand between the vehicles and edge node and when it tries to serve the remainder of the demand in the cloud, it finds the network capacity at the vehicles and edge node layers was already occupied by the traffic of distributed processing in the vehicular and edge layers.

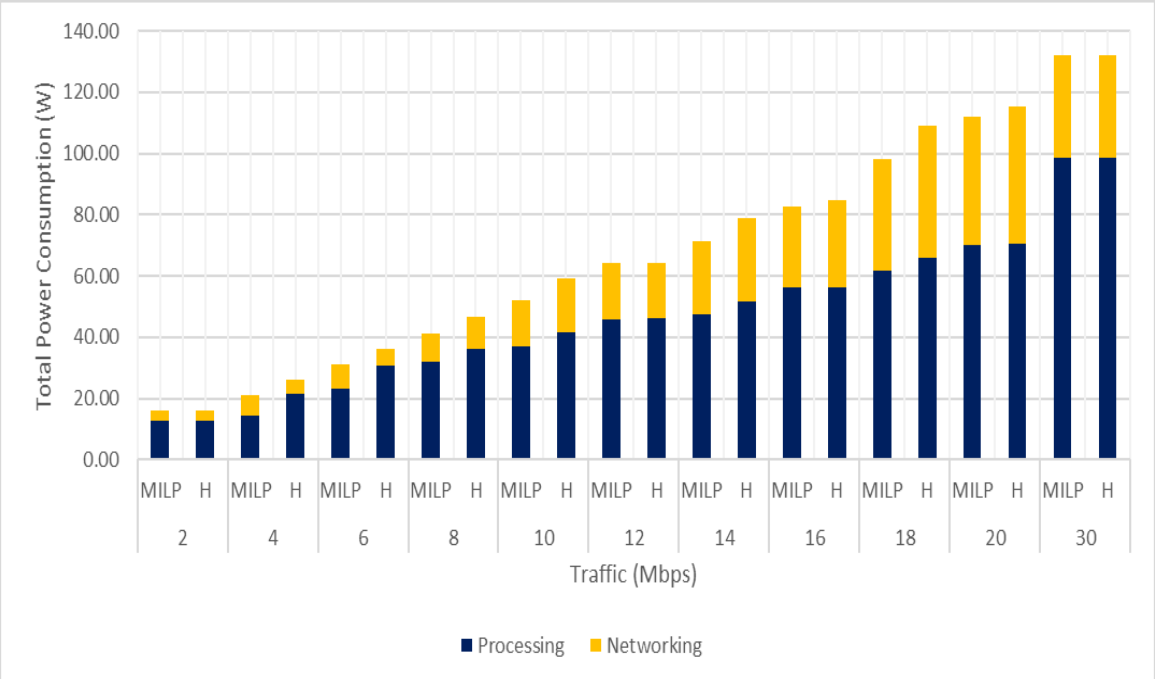

Figure 4.23: Total power consumption when serving a single demand considering the VEC scenario (Heuristics and MILP)

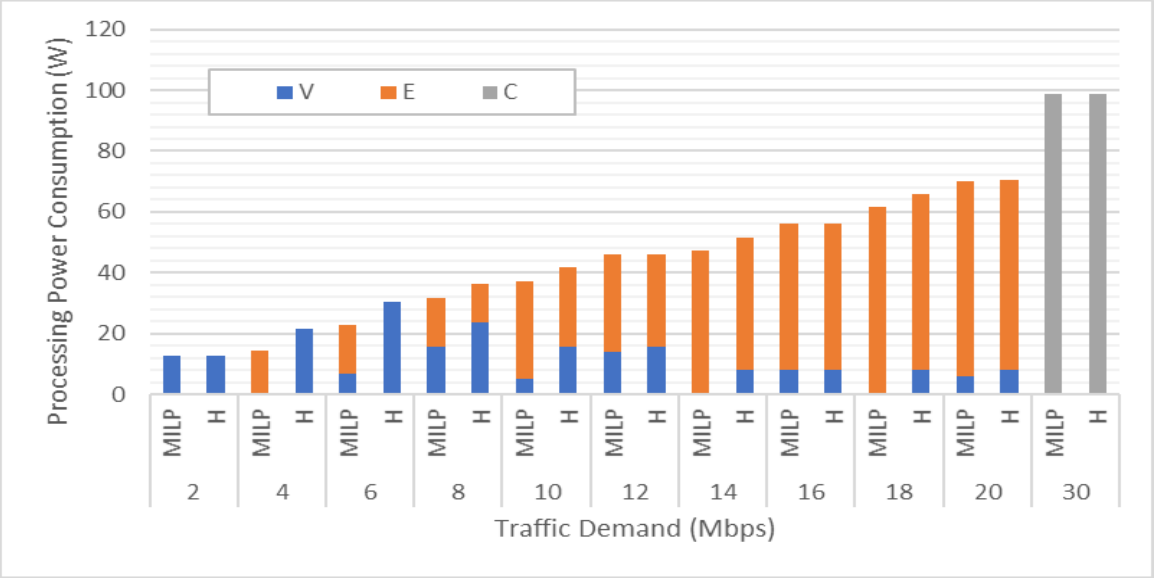

Figure 4.24: Processing power consumption when serving a single demand considering the VEC scenario (Heuristics and MILP)

4.4.2 Processing Demand Splitting Limitations

As part of taking stock of the available resources, the heuristics counts the minimum number of nodes needed to process a specific demand in each processing layer. When it checks the candidate processing capacity, it also checks if the number of nodes required at the candidate layer is within the splitting limitation. If not, then it is seen as insufficient, and the heuristic moves to the next candidate. Figure 4.25 shows a comparison with the results of the model in Section 4.3.2, and Table 4-8 shows the gap between the power consumption values. For smaller demand values, the results are identical for all splits limitations. For demands of 3.5 Mbps and above, the results were identical for smaller number of splits, but as the number of splits increased, the heuristics produced higher power consumption. The reason for this is that as the limits on the number of processing nodes becomes larger, the possibility of serving in the vehicles increases. The heuristics only ensures that the number of processing

nodes does not exceed the limit, while the MILP tends to consolidate the demand in a destination whenever possible, leading to fewer number of active nodes and lower power consumption.

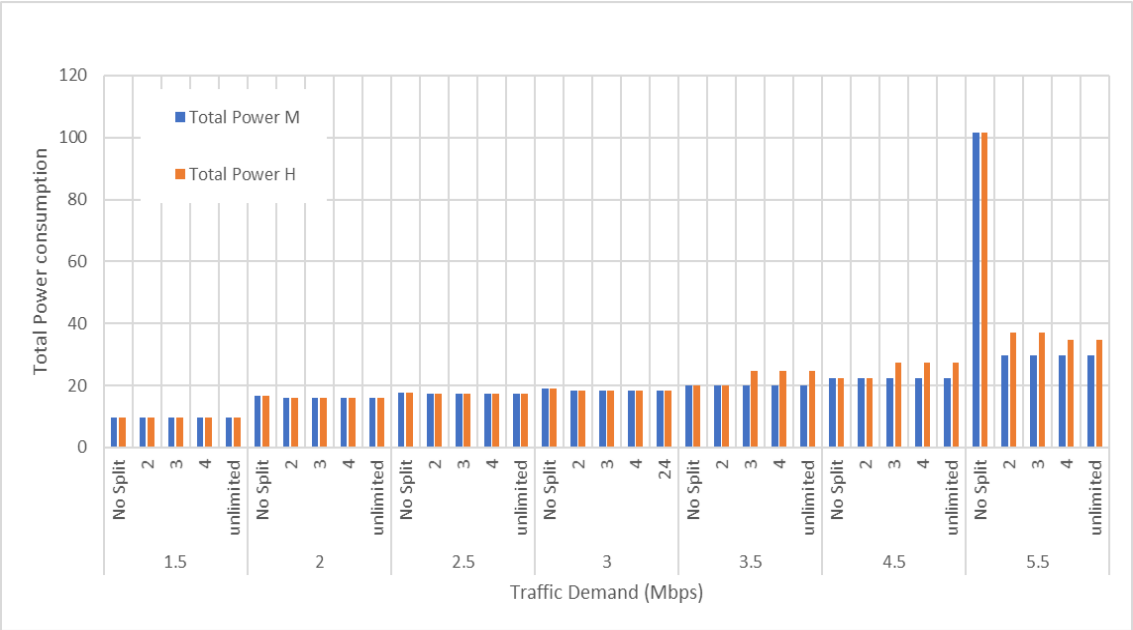

Figure 4.25: Total power consumption with demand splitting limitations (Heuristics and MILP)

Table 4-8: Heuristic vs MILP Power Consumption Difference for Demand Splitting Limitation

Traffic Demand (Mbps)/Split	**1.5**	**2**	**2.5**	**3**	**3.5**	**4.5**	**5.5**
No Split	0	0	0	0	0	0	0
2	0	0	0	0	0	0	23
3	0	0	0	0	23	21	23
4	0	0	0	0	23	21	16
Unlimited	0	0	0	0	23	21	16

4.4.3 Proportional Traffic Assignment

To implement the heuristics in this case, the same change that was made in the model in equation (4.16) was also made in the heuristics so that each destination receives traffic proportional to the processing demand it serves. Figure 4.26 shows the results of the heuristics for the proportional traffic in the V scenario, and it shows a complete match with the results of the MILP model. Figure 4.27 shows the results for the VE scenario and it shows higher power consumption in the heuristic, with the difference shown in Table 4-9. As was explained before, the difference in the power consumption comes from the sequential allocation of the demands in the heuristics.

Even though it is not shown, the VEC scenario had an exact match with the results of the V and VE scenarios in the heuristics, except for the 30 Mbps demand size, for which it made the same choice as the MILP and processed in the cloud.

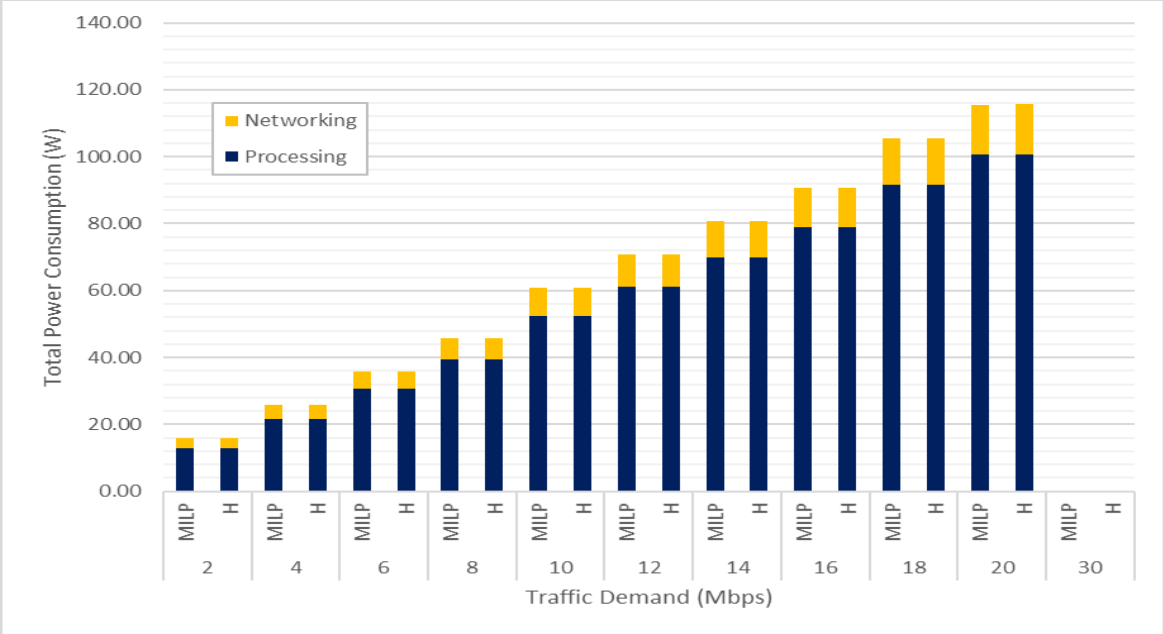

Figure 4.26: Total power consumption of proportional traffic considering the V scenario (Heuristics and MILP)

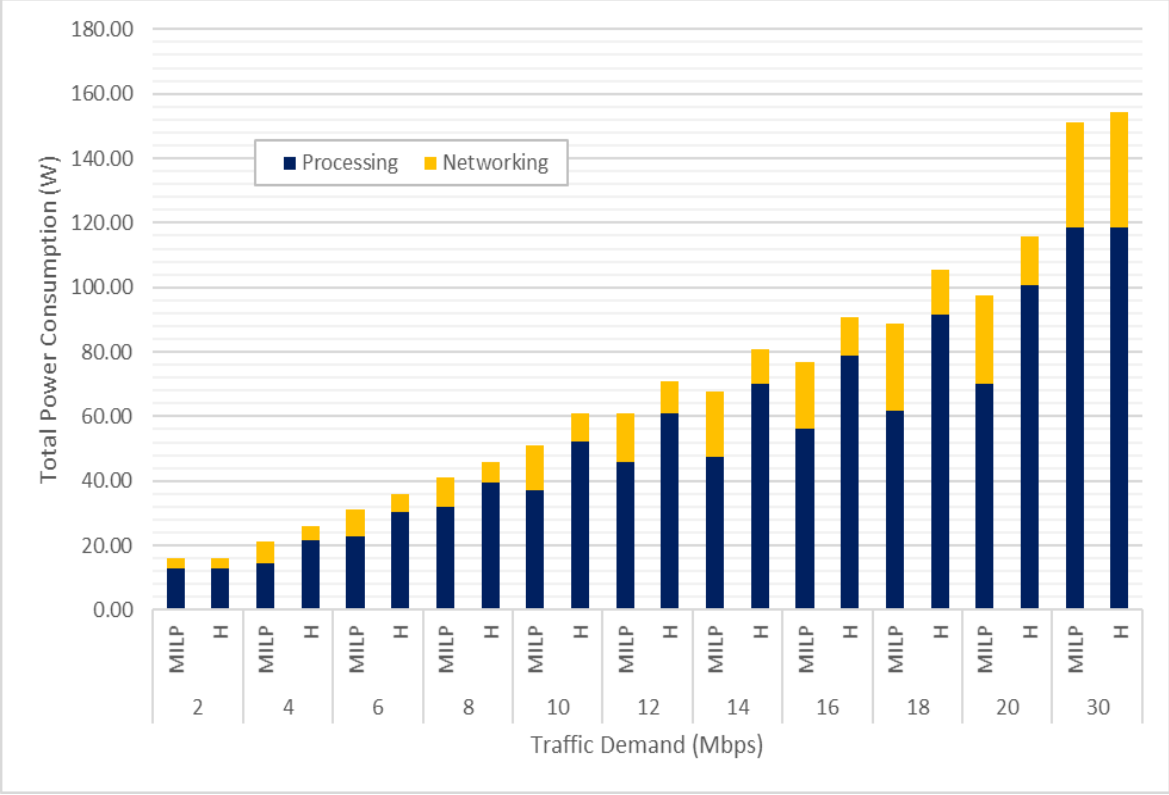

Figure 4.27: Total power consumption of proportional traffic in VE scenario (Heuristics and MILP)

Table 4-9: Heuristic and MILP Power Consumption Difference for Proportional Traffic

Traffic Demand (Mbps)	Diff (%)	Traffic Demand (Mbps)	Diff (%)
2	0	14	19
4	22	16	18
6	16	18	19
8	12	20	19
10	19	30	2
12	16		

4.4.4 Multiple Demands Services

Figure 4.28 - Figure 4.30 show a comparison of the power consumption for multiple demands as obtained from the model and the heuristics, all using VEC scenario. As before, the heuristics sequential approach gave priority for processing in the vehicles before moving to the edge and then the cloud. The distribution over more processing nodes and the choice of first sufficient route with least hops increases the power consumption in the heuristics. The behaviour is similar for low, medium, and high demand sizes. However, for the high demand sizes with the number of requests above 5, the demand requirements exceeded the capacities of the vehicles and edge layers. The heuristics sorted the candidates in descending order, therefore, the demands for these cases were served in the cloud, similar to the model. The higher power consumption of these

cases is due mainly to the routing decisions, which are based on the minimum hop route available at the time with sufficient bandwidth. Table 4-10 shows the difference in the power consumption between the optimal solution (MILP) and the heuristic.

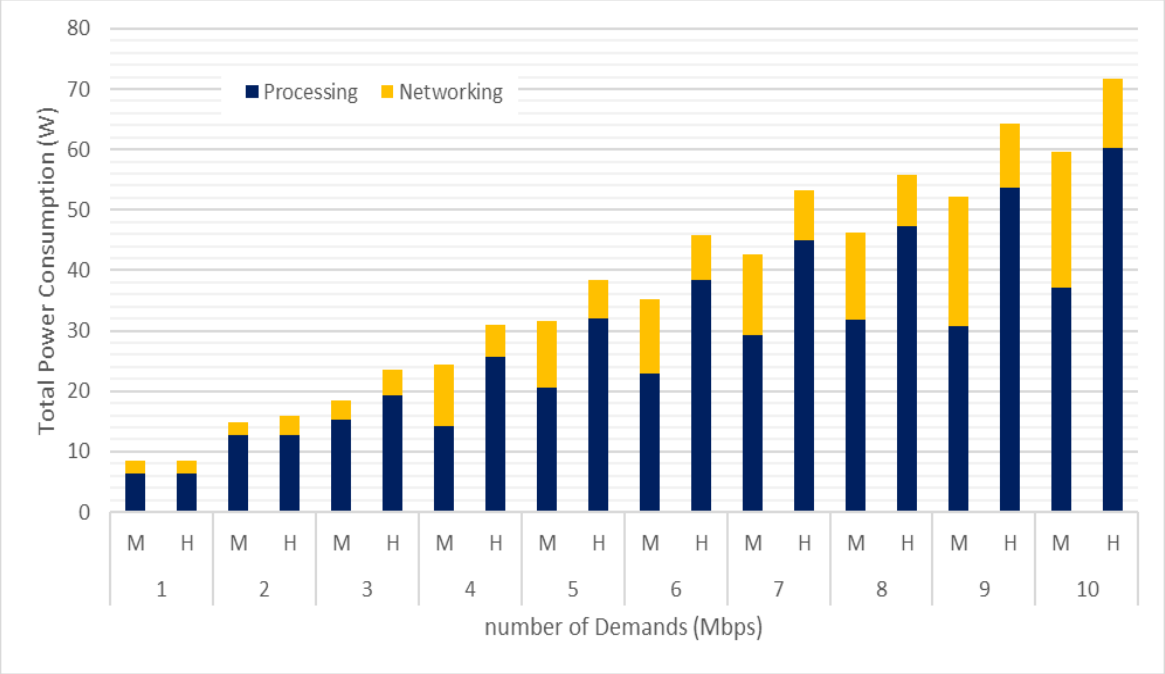

Figure 4.28: Total power consumption for low requirements demands (Heuristics and MILP)

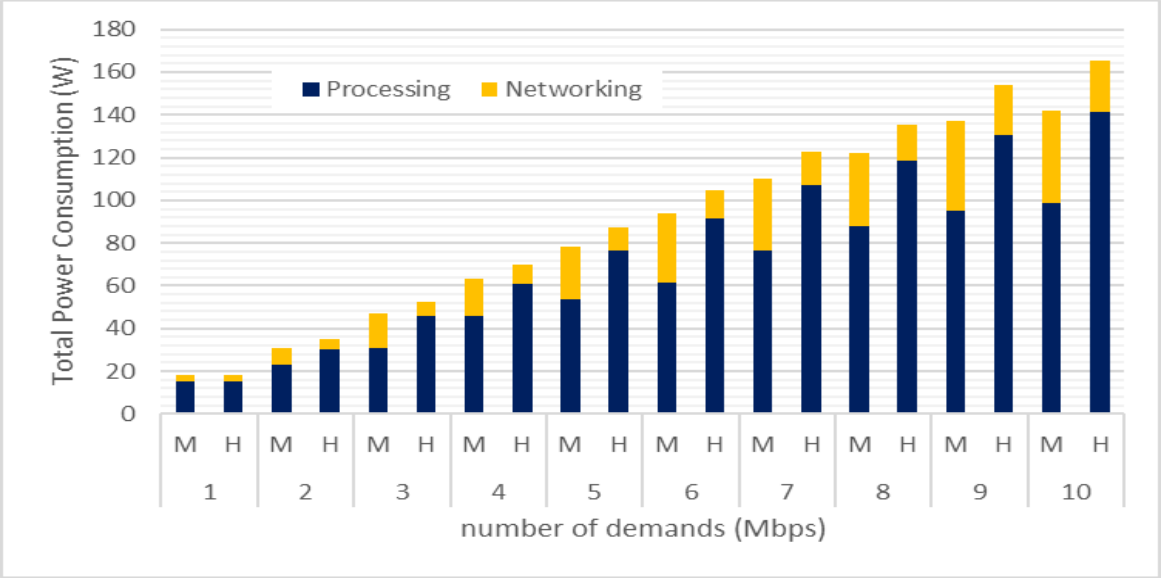

Figure 4.29: Total power consumption for medium requirements demands (Heuristics and MILP)

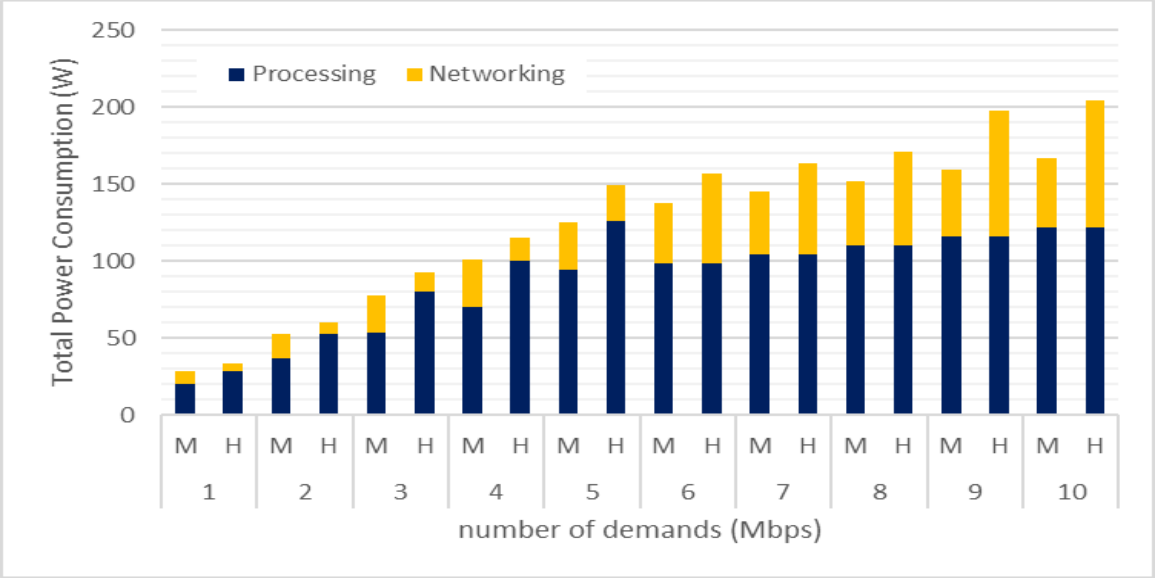

Figure 4.30: Total power consumption for high requirements demands total power consumption (Heuristics and MILP)

Table 4-10: Heuristics vs MILP for Multiple Demands

Traffic Demand (Mbps)	Diff (%)
Low	0-25
Medium	0-16
High	12-25

4.5 Summary

This chapter has investigated the use of underutilised computing resources in modern vehicles to create a processing layer, referred to as the vehicular cloud, in proximity of end users. The vehicular cloud complements conventional cloud computing and fixed edge computing in a distributed processing architecture. The architecture was modelled using a MILP model with the objective of minimising the total power consumption. The results of the MILP model show that the energy efficiency of processing in vehicles compared to the cloud decreases as the size of the demand increases. Processing in a combination of vehicles and edge nodes results in average power savings of 6% compared to processing in the cloud for demands of traffic as high as 18 Mbps. The limited data rate of the vehicle wireless interfaces cannot support distributed processing in vehicles and edge nodes as the traffic is replicated to all processing destinations. Therefore, vehicular communication interfaces of higher data rate are essential to improve the utilisation of vehicular clouds. The results also illustrate that splitting a processing demand improves the energy efficiency of processing in the vehicles

and edge nodes by 71%. Furthermore, the results show applications which require proportional traffic splitting among the processing destinations serving the demand. These applications can be more efficiently processed by vehicles and edge nodes, thus increasing the average power savings to 3%-16% compared to cloud processing, even for demands up to 20 Mbps. A real-time heuristic for allocating processing demands is developed based on insights from the model. The results show that the heuristic has comparable performance to the MILP model.

Chapter 5
Joint Minimisation of Energy and Delay in a Vehicular Cloud Architecture

5.1 Introduction

One motivation behind the research on alternatives to cloud computing is the long distance data needs to travel over the network affecting the power consumption and latency of the service. Vehicular networks are at closer proximity to end users but how they contribute to delay is an issue to be addressed. Some research efforts have studied delay performance in vehicular networks. The work in [174] presents an SDN-enabled delay efficient vehicular networks. A model for delay sensitive communications in distributed processing systems is presented in [175]. A multi-objective technique for task scheduling and allocation in vehicular networks is introduced in [48]. The authors of [176] provide a model for latency and job allocation that considers mobility in vehicular networks.

In the previous chapter, we introduced an energy efficient vehicular cloud architecture to be used in distributed processing. In addition to vehicular processing, the architecture provided processing at edge nodes and the conventional cloud. We evaluated the performance of the architecture considering smart city applications continuously supported by the cloud, subject to SLA data rate and processing speed requirements. The results showed that vehicular cloud, supported by processing in edge nodes, is the optimal processing destination, as long processing and networking capacities allows it.

In this chapter we develop a framework for studying the energy efficiency and delay performance of on-demand smart city applications where users request a one-off demand. As opposed to the energy efficiency of continuously running applications which can be studied considering the power consumption of the application, the study of the energy efficiency of on-demand applications needs to consider the energy consumption of the applications as the time required to serve the application, i.e., the time to send traffic to processing nodes and the processing time, varies with the processing and data rate requirements. In this chapter, we develop a MILP model to minimise the energy consumption and delay experienced by on-demand smart city applications and study the trade-off between energy consumption and delay.

5.2 MILP Model

In this section we present the MILP model developed to optimally allocate processing demands and route traffic between source nodes and processing destinations in the proposed architecture, so the energy consumption and/or end-to-end delay is minimised. Table 5-1 and

Table 5-2 list the parameters and variables of the model. Note that the units of the processing and networking demands are million instructions (MI) and kilobits (kb), respectively, to reflect the on-demand applications type, instead of MIPS and kbps used in the previous chapter. The units of processing and networking capacities of the different nodes are also presented accordingly.

Table 5-1: Model Parameters

ND	Set of vehicles
ED	Set of edge nodes
SD	Set of cloud servers
OLT	Set of OLT devices
MD	Set of metro nodes
CD	Set of core nodes
N	Set of all nodes in the architecture
Nm_n	Set of neighbouring nodes of node n, $n \in N$
$\mathcal{U}_s$	Processing demand generated by node s (MI), $s \in N$
$\mathcal{V}_s$	Traffic demand generated by node s (kb), $s \in N$
C_n	Processing capacity of node n (MIPS), $n \in N$
$\mathcal{K}_n$	Processing efficiency of node n (J/MI), $n \in N$
S	Maximum number of processing nodes to process a demand
B_{nm}	Maximum data rate on the link between n and m (Mbps), $n \in N, m \in N$
B_n	Maximum data rate node *n* can support (Mbps), $n \in N$

B^{VE}	Maximum data rate the WiFi interface of a vehicle (Mbps) (This parameter is defined in addition to $B_n \ \ n \in ND$ to account for the two communication interfaces of vehicles).
B^{ONU}	Data rate of ONU at an edge node (Mbps) (This parameter is defined in addition to $B_n \ \ n \in ED$ to account for the two communication interfaces of edge nodes).
D_{nm}	Distance between node pair (m) $n, m \ \in N$
NM_n	Maximum power consumption of networking at node $n \in N$ (W)
NI_n	Idle power consumption of networking at node $n \in N$ (W)
PM_n	Maximum power consumption of processing at node $n \in N$ (W)
PI_n	Idle power consumption of processing at node $n \in N$ (W)
NM^{ONU}	Maximum power consumption of ONU at an edge node (W)
NI^{ONU}	Idle power consumption of ONU at an edge node (W)

TX	The maximum transmission power consumption of wireless interface (W)
$\mathcal{T}_{nm}$	Wireless transmission energy per bit over link(n,m) $where\ n,m\ \in ND \cup ED$ (j/b)
RX	Receiver sensitivity of wireless interface (W)
$\mathcal{R}_{mn}$	Wireless reception energy per bit at node n from over link(m,n), $n,m \in ND \cup ED$ (j/b)
ϵ	Power amplifier factor for wireless communication (j/b.m$^{2)}$
$\mathcal{E}_n$	Energy per bit of networking at node n, $n \in OLT \cup MD \cup CD \cup ED$ (j/b)
PUE_n	Power usage effectiveness $n \in N$
A	A large Constant
Delay Sets and Parameters	
TA	Set of arrival rates (packets/s) (representing different communication interfaces)
SR	Set of service rates of nodes (packets/s) (representing different communication interfaces)
Dr_{nm}	Data rate over link (m,n), (packets/s), $m,n \in N$

$Pckt$	Packet size in kb
$PcktC_s$	Number of packets per demand of source node $s \in N$
$ArrQ_{rj}$	Queueing delay in s/packet for arrival value $j \in TA$, and service rate r , $r \in SR$
Sr_{mnr}	Sr_{mnr} = 1 if node n uses service rate r to handle arrivals from node m, otherwise $Sr_{mnr} = 0$, $n, m \in N$
VT_n	Transmission delay of vehicular node n through the DSRC interface (ms), $n \in ND$
SoL_{nm}	The propagation delay of link (n, m). $n, m \in N$
γ, δ	Objective function weighting factors
y	Time period equal to 1 second
Z	Maximum queueing delay expected (s/packet)
μ_n	Service rate at node n (Mbps)

Table 5-2: Model Variables

JED	Sum of energy consumption and delay
$\mathcal{W}_n$	Total energy consumption at node $n \in N$ (J)
$\mathcal{N}_n$	Networking energy consumption at node $n \in N$ (J)
$\mathcal{P}_n$	Processing energy consumption at node $n \in N$ (J)

u_{sd}	The amount of processing demand of source node s served by processing node d, $s, d \in N$ (MI)
v_{sd}	Traffic demand in bits from source node s to processing node d, $s, d \in N$ (bits)
f_{nm}^{sd}	The amount of traffic demand in bits from source node s to processing node d traversing link(n, m) where $s, d, n \in N, m \in Nm_n$ (bits)
α_{sd}	$\alpha_{sd} = 1$ if demand of source node s is served by processing destination d, otherwise $\alpha_{sd} = 0$. $s, d \in N$
Q_s	Total number of processing nodes serving demand of source s, $s \in N$
β_n^{Net}	$\beta_n^{Net} = 1$ if node n is used for networking, $n \in N$, otherwise $\beta_n^{Net} = 0$
β_n^{Pr}	$\beta_n^{Pr} = 1$ if node n is used for processing, $n \in N$, otherwise $\beta_n^{Pr} 0$
β_n^{ONU}	$\beta_n^{ONU} = 1$ if ONU at node n is used, $n \in ED$, otherwise $\beta_n^{ONU} = 0$

Delay Variables

$\mathcal{L}_s$	Maximum delay of serving demand of node s in all destinations $s \in N$ (ms)
Del_{sd}	End-to-end delay between any pair of source and destination $s, d \in N$ (ms)
TD_{sd}	Transmission delay between pair (s, d), $s, d \in N$ (ms)
PrD_{sd}	Processing delay between pair (s, d), $s, d \in N$ (ms)
PD_{sd}	Propagation delay between pair (s, d), $s, d \in N$ (ms)
QD_{sd}	Queueing delay between pair (s, d), $s, d \in N$ (ms)
td_{nm}^{sd}	Portion of Transmission delay between pair (s, d) that come from link(n, m), $s, d, n \in N$, $m \in Nm_n$ (ms)
pd_{nm}^{sd}	Portion of propagation delay between pair (s, d) that come from link (n, m). $s, d, n \in N$, $m \in Nm_n$ (ms)
qd_{mn}^{sd}	Portion of queueing delay between pair (s, d) that come from link (n, m) . $s, d, n \in N$, $m \in Nm_n$ (ms)
Arr_{nr}	Total number of packets arriving at node $n$to be handled with service rate r. $n \in N$, $r \in SR$
X_{nrj}	X_{nrj} = 1 if arrival rate at node n handled with service rate r equal to j, otherwise $X_{nrj} = 0$. $n \in N$, $r \in$ SR, , $j \in TA$

QU_{nr}	Queueing delay at node n with service rate r. $n \in N$, $r \in SR$ (s/packet)
LK_{nm}	LK_{nm} = 1 if the traffic is routed over link (n, m), otherwise $LK_{nm} = 0$. $n \in N$, $m \in Nm_n$
L_{nm}^{sd}	$L_{nm}^{sd} = 1$ if the traffic demand of pair (s, d) is routed over link (n, m), otherwise $L_{nm}^{sd} = 0$. $s, d, n \in N, m \in Nm_n$
λ_{m1n}^{s1d1}	The amount of traffic demand in packets/s from source node $s1$ to processing node $d1$ traversing link$(n, m1)$ where $s, d, n \in N, m1 \in Nm_n$

The objective of the model is to minimise a weighted sum of energy consumption and delay (JED) given as

$$JED = \gamma \sum_{n \in N} \mathcal{W}_n + \delta \sum_{n \in N} \mathcal{L}_n \quad (5.1)$$

where

$$\mathcal{W}_n = PUE_n(\mathcal{N}_n + \mathcal{P}_n) \quad \forall n \in N \quad (5.2)$$

and

$$\mathcal{L}_s \geq Del_{sd} \quad \forall s, d \in N \quad (5.3)$$

where

$$Del_{sd} = TD_{sd} + PD_{sd} + \ PrD_{sd} + QD_{sd} \quad \forall\, s, d \ \in N, s \neq d \quad (5.4)$$

The MILP model objective, as given in equation (5.1), is to minimise the energy consumption and delay; γ and δ are used as weighting factors to scale the power and delay values and to prioritise one over the other. Equation (5.2) gives the energy consumption of a node as the sum of the networking power consumption and the processing power consumption multiplied by the power usage effectiveness (PUE) of the node. The PUE is the ratio of the total energy consumed by a networking or computing node to the energy consumed by the IT equipment only. It is an important measure of efficiency as modern computing and networking nodes require non-computing components for their operation, such as cooling and ventilation systems. A PUE equal to 1, means that all power is consumed in performing IT operations.

Inequality (5.3) finds the maximum end-to-end delay experienced by a source node demand. As the processing demand can be split between more than one destination, the maximum delay accounts for the job that finishes last. The objective of the model is to minimise the latency of the longest job. The end-to-end delay between a demand source and a processing destination is the sum of the different delay components as given in equation (5.4).

As explained in Chapter 4, the processing and networking devices are assumed to follow a linear profile where the power consumption is composed of an idle power consumption which is power consumed to activate the device, and load dependent power consumption, obtained by multiplying the device load by the energy per bit. In the following we present a detailed model of the network energy consumption and processing energy consumption of the different nodes in the network, followed by delay components.

Energy consumption of vehicular nodes:

$$\mathcal{N}_n = NI_n VT_n + NI_n MaxQ_n + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n} \mathbb{f}_{nm}^{sd}\ \mathcal{T}_{nm} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n} \mathbb{f}_{mn}^{sd}\ \mathcal{R}_{mn} \qquad \forall\, n \in ND \tag{5.5}$$

Equation (5.5) gives the networking energy consumption of a vehicular node as the sum of the energy consumed to activate the OBU, the traffic-dependent, distance-dependent transmission energy consumption and the traffic-dependent reception energy consumption. The activation time of the OBU is given as the maximum queuing delay experienced by traffic relayed by the node and the transmission delay of the traffic originating at the node. Note that other delay components overlap with these delay components. The transmission delay, given by parameter VT_n is conversely based on the DSRC interface of the lower data rate.

Energy consumption of edge nodes:

$$\mathcal{N}_n = NI_n \sum_{r \in SR, r=B_n} TotalQ_{rn} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in Nm_n \cap (ND \cup ED)} \not{\ell}_{nm}^{sd} \mathcal{T}_{nm} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in Nm_n \cap (ND \cup ED)} \not{\ell}_{mn}^{sd} \mathcal{R}_{mn} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in Nm_n \cap (OLT)} (\not{\ell}_{nm}^{sd} + \not{\ell}_{mn}^{sd}) \mathcal{E}_n + NI_n^{ONU} \sum_{r \in SR, r=B^{ONU}} TotalQ_{rn} \quad \forall n \in ED \tag{5.6}$$

Equation (5.6) gives the energy consumption of the edge node's two communication interfaces: the WiFi interface through its AP, and the PON interface through the ONU. The AP (WiFi interface) energy consumption is calculated similar to the vehicular nodes. The PON interface energy consumption is found considering the traffic routed between the edge node ONU and the OLT. The energy consumed to activate the AP and the ONU is calculated considering the queueing delay experienced by traffic routed through each interface.

The traffic-dependent distance-dependent energy per bit for wireless transmission ($\mathcal{T}_{nm}$) is given as

$$\mathcal{T}_{nm} = \frac{TX}{B_{nm}} + \epsilon * D_{nm}^2 \tag{5.7}$$

$$\forall n \in ND \cup ED, \forall m \in Nm_n \cap (ND \cup ED)$$

The first term of equation (5.7) gives the traffic dependent part found by dividing the transmitter maximum power consumption (TX) by the link maximum data rate. The second term gives the distance-dependent power consumption as a function of transmission distance and the power amplifier factor. The reception energy per bit for wireless transmission ($\mathcal{R}_{mn}$) is found by dividing the node receiver sensitivity RX by the link maximum data rate, as in equation (5.8).

$$\mathcal{R}_{mn} = \frac{RX}{B_{mn}} \tag{5.8}$$

$$\forall\, n \in ND \cup ED, \forall\, m \in Nm_n \cap (ND \cup ED)$$

Energy consumption of the OLT, metro, core nodes:

$$\mathcal{N}_n = NI_n \sum_{r \in SR, r=B_n} TotalQ_{rn} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in Nm_n} \not{v}_{mn}^{sd}\ \mathcal{E}_n \tag{5.9}$$

$$\forall\, n \in OLT \cup MD \cup CD$$

$$\mathcal{E}_n = \frac{(NM_n - NI_n)}{B_n} \quad \forall\, n \in OLT \cup MD \cup CD \tag{5.10}$$

$$\mathcal{E}_n = \frac{(NM^{ONU} - NI^{ONU})}{B^{ONU}} \quad \forall\, n \in ED \tag{5.11}$$

The networking power consumption for the OLT, metro and core nodes are calculated in equation (5.9) by summing the energy consumed to activate the

node and the load proportional energy consumption. The energy per bit of a node is given in equations (5.10) - (5.11).

Processing energy consumption:

$$\mathcal{P}_n = PI_n \sum_{s \in N} PrD_{sn} + \sum_{s \in N} \mathcal{u}_{sn} \mathcal{K}_n \quad \forall\, n \in N \quad (5.12)$$

The processing energy consumption of a node is given in equation (5.12) considering the energy consumed to activate the servers and the processing load dependent energy consumption which is a function of the node processing efficiency (the energy consumed per instruction) as shown in equation (5.13)

$$\mathcal{K}_n = \frac{(PM_n - PI_n)}{C_n} \quad \forall\, n \in N \quad (5.13)$$

Transmission Delay:

$$TD_{sd} = \sum_{n \in N} \sum_{m \in N m_n} td_{nm}^{sd} \quad \forall\, s, d \in N \quad (5.14)$$

$$td_{nm}^{sd} = \frac{\mathcal{f}_{nm}^{sd}}{(Dr_{nm} * Pckt)} \quad \forall\, s, d, n \in N, \forall\, m \in Nm_n \quad (5.15)$$

The transmission delay experienced by traffic between a demand source node and a processing destination is calculated by summing the transmission delay at nodes of the traffic route as given equation (5.14). Equation (5.15) calculates the

transmission delay experienced by a traffic demand at a node on its route. Note that we assume that traffic bifurcation is not allowed, i.e., a relay node in the route of a traffic demand will be retransmitted through a single link.

Propagation delay:

$$PD_{sd} = \sum_{n \in N} \sum_{m \in Nm_n} pd_{nm}^{sd} \quad \forall s, d \in N \quad (5.16)$$

$$pd_{nm}^{sd} = L_{nm}^{sd} SoL_{nm} \quad \forall s, d, n \in N, \forall m \in Nm_n \quad (5.17)$$

The propagation delay experienced by traffic between a demand source node and a processing destination is calculated in equation (5.16) as the sum of the propagation delay of each link in the route traversed by the traffic.

Propagation delay, as shown in equations (5.17)-(5.19), is a function of the distance and the medium-specific propagation speed. For wireless medium, the propagation speed is equal to the speed of light c. For optical fibres, the propagation speed is the speed of light divided by the refraction index of fibre.

For wireless medium:

$$SoL_{nm} = \frac{D_{nm}}{c} \quad \forall n, m \in (ND \cup ED) \quad (5.18)$$

For Optical Fibres:

$$SoL_{nm} = \frac{D_{nm}}{(\frac{C}{v})}, \qquad v = \frac{3}{2}, \qquad \forall\, n, m \in (ED \cup OLT \cup MD \cup CD) \quad (5.19)$$

Processing Delay:

$$PrD_{sd} = \frac{u_{sd}}{C_d} \quad \forall\, s, d \in N \quad (5.20)$$

The processing delay of a demand at a processing destination is given in equation (5.20) considering the part of the demand processed by the destination and the destination processing capacity.

Queueing Delay:

The queuing delay calculations are based on a M/M/1 model, where each node has only one server, a Poisson arrival at λ packets per second and exponentially distributed service time at μ packets per second. The delay under the M/M/1 model is given as $\frac{1}{\mu - \lambda}$. For each node, we are assuming the queue is independent of previously traversed queues.

In the model, the queuing delay experienced by a traffic demand at a node along its route is given in equation (5.21). The arrival rate at a node is calculated by summing the traffic traversing the links ending at this node.

$$qd_{nm}^{sd} = L_{mn}^{sd} \frac{1}{\mu_n - \sum_{s1 \in N} \sum_{d1 \in N} \sum_{m1 \in Nm_n} \lambda_{m1n}^{s1d1}} \quad (5.21)$$

$$\forall\, s, d, n \in N, \forall\, m \in Nm_n$$

where:

$$\mu_n = \frac{B_n}{Pckt} \quad \forall\, n \in N \quad (5.22)$$

and:

$$\lambda_{m1n}^{s1d1} = L_{m1n}^{s1d1} Dr_{m1n} \quad \forall\, s1, d1, n \in N, \forall\, m1 \in Nm_n \quad (5.23)$$

Equation (5.21) is nonlinear. The model is linearised using equations (5.24) - (5.27) which represent a lookup table to find the queuing delay that corresponds to the arrival rate of the node.

$$Arr_{nr} = \sum_{s \in N} \sum_{d \in N} \sum_{m \in Nm_n} Sr_{mnr} LK_{mn} Dr_{mn} \quad \forall\, r \in SR, \forall\, n \in N \quad (5.24)$$

$$Arr_{nr} = \sum_{j \in TA} X_{nrj}\ j \quad \forall\, r \in SR, \forall\, n \in N \quad (5.25)$$

$$\sum_{j \in TA} X_{nrj} \leq 1 \quad \forall\, r \in SR, \forall\, n \in N \quad (5.26)$$

$$QU_{nr} = \sum_{j \in TA} X_{nrj}\, ArrQ_{rj} \quad \forall\, r \in SR, \forall\, n \in N \quad (5.27)$$

Equation (5.24) finds the arrival rate of a node through an interface by summing traffic arrivals from all links of the node. The value calculated in equation (5.24) is one of the predefined arrival rates in the set TA. Equation (5.25) sets a binary variable to 1 to indicate the arrival rate calculated in equation (5.24). Equation (5.26) ensures that only one arrival value per service rate per node is found. Equation (5.27) uses this binary variable to obtain the corresponding pre-calculated queueing delay from the lookup table given by $ArrQ_{rj}$.

$$qd_{mn}^{sd}\ Sr_{mnr} = \ L_{mn}^{sd}\, QU_{nr} \quad \forall\, r \ \in \ SR, \forall\, s, d, n \ \in N, \forall\, m \ \in Nm_n \quad (5.28)$$

Equation (5.28) finds the queuing delay experienced by a traffic demand at a node on its route considering the communication interface used by the demand. Equation (5.28) is non-linear as it includes the multiplication of two variables. Therefore, it is replaced by the linear inequalities (5.29)-(5.31) which collectively achieve the multiplication of the variables. Table 5-3 illustrates those inequalities (5.29) -(5.31) hold for values of $\boldsymbol{qd_{nm}^{sd}}$ calculated by (5.28).

$$qd_{mn}^{sd} \leq \ L_{mn}^{sd} Z \quad \forall\, s, d, n \ \in N, \forall\, m \ \in Nm_n \quad (5.29)$$

$$Sr_{mnr} qd_{mn}^{sd} \leq QU_{nr} \quad \forall r \in SR, \forall s, d, n \in N, \forall m \in Nm_n \quad (5.30)$$

$$Sr_{mnr} qd_{mn}^{sd} \geq QU_{nr} - Z(1 - Sr_{mnr} L_{mn}^{sd}) \quad (5.31)$$
$$\forall r \in SR, \forall s, d, n \in N, \forall m \in Nm_n$$

Table 5-3: Truth table of inequalities (5.29-5.31)

Binary Variable	Equations	Conclusions	eq (5.28)
$L_{mn}^{sd} = 0$	*(5.29)* $qd_{mn}^{sd} \leq 0$	qd_{mn}^{sd} is a positive number. For 5.29 and 5.30 to hold, it needs to be 0. 5.31 is a negative number, to hold qd_{mn}^{sd} must be 0	$qd_{mn}^{sd} = 0$
	(5.30) $qd_{mn}^{sd} \leq QU_{nr}$		
	(5.31) $qd_{mn}^{sd} \geq QU_{nr} - Z$		
$L_{mn}^{sd} = 1$	*(5.29)* $qd_{mn}^{sd} \leq Z$	qd_{mn}^{sd} is a positive number For (5.20) and (5.30) to hold qd_{mn}^{sd} must be less than or equal to $\mathrm{QU_{nr}}$	$qd_{mn}^{sd} = QU_{nr}$
	(5.30) $qd_{mn}^{sd} \leq QU_{nr}$		

	(5.31) $qd_{mn}^{sd} \geq QU_{nr}$	For (5.30) and (5.31) to hold qd_{mn}^{sd} must be equal to QU_{nr}	

$$TotalQ_{rn} = \sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n} Sr_{mnr}\, QU_{nr} PcktC_s \quad (5.32)$$

$$\forall\, r \in SR, \forall\, n \in ND$$

$$MaxQ_n \geq TotalQ_{rn} \quad \forall\, r \in SR, \forall\, n \in ND \quad (5.33)$$

The queueing delay at each node is used to calculate the energy consumed in activating the node. Equation (5.32) finds the queueing delay for the total arrival (total traffic) at each node at a given service rate. If a node has more than one service rate. The maximum active time is the highest of the queueing delays per service rate, which is found in inequality (5.33).

$$QD_{sd} = PcktC_s \sum_{n \in N} \sum_{m \in N m_n} qd_{mn}^{sd} \quad \forall\, s, d \in N \quad (5.34)$$

Equation (5.34) calculates the queueing delay experienced by the demand for all relay nodes in its a route

The model is subject to the following constraints:

$$\mathcal{U}_s = \sum_{\substack{d \in N \\ d \neq s}} u_{sd} \qquad \forall\, s \in N \qquad (5.35)$$

Constraint (5.35) states that the processing demand for a source node must be fully served by the processing destinations.

$$\sum_{\substack{s \in N \\ s \neq d}} \frac{u_{sd}}{C_d} \leq y \quad \forall\, d \in N \qquad (5.36)$$

Constraint (5.36) gives the processing nodes capacity constraints. To simplify the capacity constraint, we assume demands do not queue for processing, i.e., the processing allocated to a node is less than or equal its processing capacity in one second, $y = 1$.

$$u_{sd} \geq \alpha_{sd} \qquad \forall\, s, d \in N, s \neq d \quad (5.37)$$

$$u_{sd} \leq A\, \alpha_{sd} \qquad \forall\, s, d \in N, s \neq d \quad (5.38)$$

A binary variable is set to indicate that a node is selected as a processing destination for a demand source in constraints (5.37) and (5.38).

$$\mathcal{F}_{sd} = \mathcal{V}_s \alpha_{sd} \qquad \forall\, s, d \in N\,, s \neq d \quad (5.39)$$

The processing demand can be served in several processing nodes. Each processing node receives the full traffic demand from the source, as stated by constraint (5.39).

$$\sum_{m \in Nm_n} \mathbb{f}_{nm}^{sd} - \sum_{m \in Nm_n} \mathbb{f}_{mn}^{sd} = \begin{cases} \mathcal{F}_{sd} & if\ n = s \\ -\mathcal{F}_{sd} & if\ n = d \\ 0 & otherwise \end{cases} \quad (5.40)$$

$$\forall\, s, d, n \in N\ , s \neq d$$

Constraint (5.40) is the flow conservation constraint. It ensures that the amount of traffic received by a relay node is equal to the amount re-transmitted.

$$\mathbb{f}_{nm}^{sd} \geq L_{nm}^{sd} \quad \forall\, s, d, n \in N, \forall\, m \in Nm \quad (5.41)$$

$$\mathbb{f}_{nm}^{sd} \leq A\, L_{nm}^{sd} \quad \forall\, s, d, n \in N, \forall\, m \in Nm \quad (5.42)$$

$$\sum_{m \in Nm_n} L_{nm}^{sd} \leq 1 \quad \forall\, s, d, n \in N \quad (5.43)$$

Constraints (5.41) - (5.42) set a binary variable to 1 to indicate links traversed by a traffic demand. Traffic between a source-destination pair is not allowed to split among multiple routes as enforced by constraint (5.43).

$$\sum_{s\in N}\sum_{d\in N} f_{nm}^{sd} \geq LK_{nm} \quad (5.44)$$

$$\forall\, n \in N, \forall\, m \in Nm_n$$

$$\sum_{s\in N}\sum_{d\in N} f_{nm}^{sd} \leq ALK_{nm} \quad (5.45)$$

$$\forall\, n \in N, \forall\, m \in Nm_n$$

Another binary variable is set to 1 if any traffic traverses a link in constraint (5.44) -(5.45), which is used for the queueing delay lookup table search.

$$\sum_{m\in Nm_n} LK_{nm}Dr_{nm} + \sum_{m\in Nm_n} LK_{mn}Dr_{mn} \leq B_n/Pckt \quad (5.46)$$

$$\forall\, n \in OLT \cup MD \cup CD$$

$$\sum_{m\in Nm_n\cap ND} LK_{nm}\ Dr_{nm} + \sum_{m\in Nm_n\cap ND} LK_{mn}\ Dr_{mn} \leq B_n/Pckt \quad (5.47)$$

$$\forall\, n \in ND$$

$$\sum_{m \in Nm_n \cap ED} LK_{nm}\ Dr_{nm} + \sum_{m \in Nm_n \cap ED} LK_{mn}\ Dr_{mn} \leq B^{VE}/Pckt \quad (5.48)$$

$$\forall\, n \in ND$$

$$\sum_{m \in Nm_n \cap (ND \cup ED)} LK_{nm}\ Dr_{nm} + \sum_{m \in Nm_n \cap (ND \cup ED)} LK_{mn}\ Dr_{mn} \leq B_n/Pckt \quad (5.49)$$

$$\forall\, n \in ED$$

$$\sum_{m \in Nm_n \cap OLT} LK_{nm}\ Dr_{nm} + \sum_{m \in Nm_n \cap OLT} LK_{mn}\ Dr_{mn} \leq B^{ONU}/Pckt \quad (5.50)$$

$$\forall\, n \in ED$$

Constraint (5.46) ensures that the maximum data rate of the OLT, metro, and core nodes is always preserved. For vehicles, constraint (5.47) preserves the DSRC interface data rate, which is used to communicate with vehicles. Similarly, the data rate limits of the WiFi interface between vehicles and edge node is separately implemented in constraint (5.48) - (5.49). Similarly, for the edge node when using the optical link through ONU, the data rate is limits are enforced through constraint (5.50).

$$Q_s = \sum_{\substack{d \in N \\ d \neq s}} \alpha_{sd} \qquad \forall\, s \in N \quad (5.51)$$

$$Q_s \leq S \qquad \forall s \in N \quad (5.52)$$

To benefit from the distributed processing resources, a processing demand can be split among multiple processing destinations. Constraint (5.51) and (5.52) state the number of splits allowed.

5.3 Scenarios Studied and Results

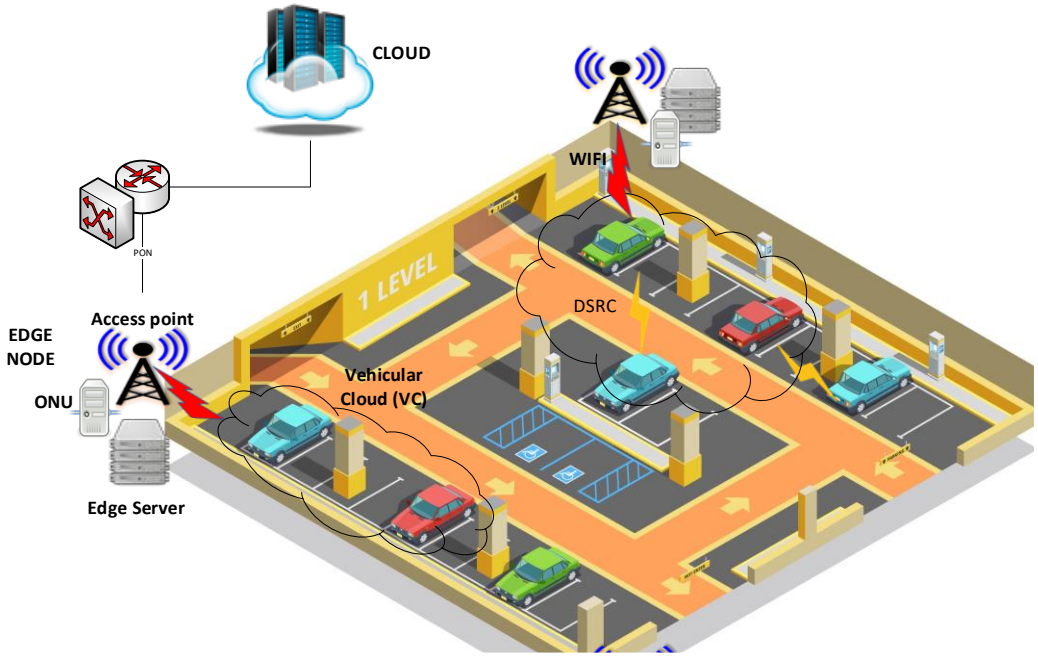

Figure 5.1: Parking lot setting

We evaluate the energy efficiency and delay performance of the proposed architecture considering a small parking lot of 45x45 meters, as illustrated in Figure 5.1, with 8 parked vehicles. The distance between vehicles ranges between 2 meters and 24 meters. The parking lot is surrounded by 2 edge nodes, placed at an average distance of 30 meters from the vehicles. Each edge node

forms a vehicular cloud of 4 vehicles. DSRC and WiFi have communication rages of several hundreds of meters [16, 158]. For the network from the edge node to the cloud, we assume the following distances:

1. From the edge node to the OLT, a typical distance of 20 km is assumed.
2. OLT is usually located at proximity of the metro node (1-10 km). We chose 5 km for our calculations.
3. To emphasise the large geographical distance of the conventional cloud, we consider a cloud located in another city, around 300 km from the city in which the demands are generated

Figure 5.2 shows the end-to-end architecture with the specified distances.

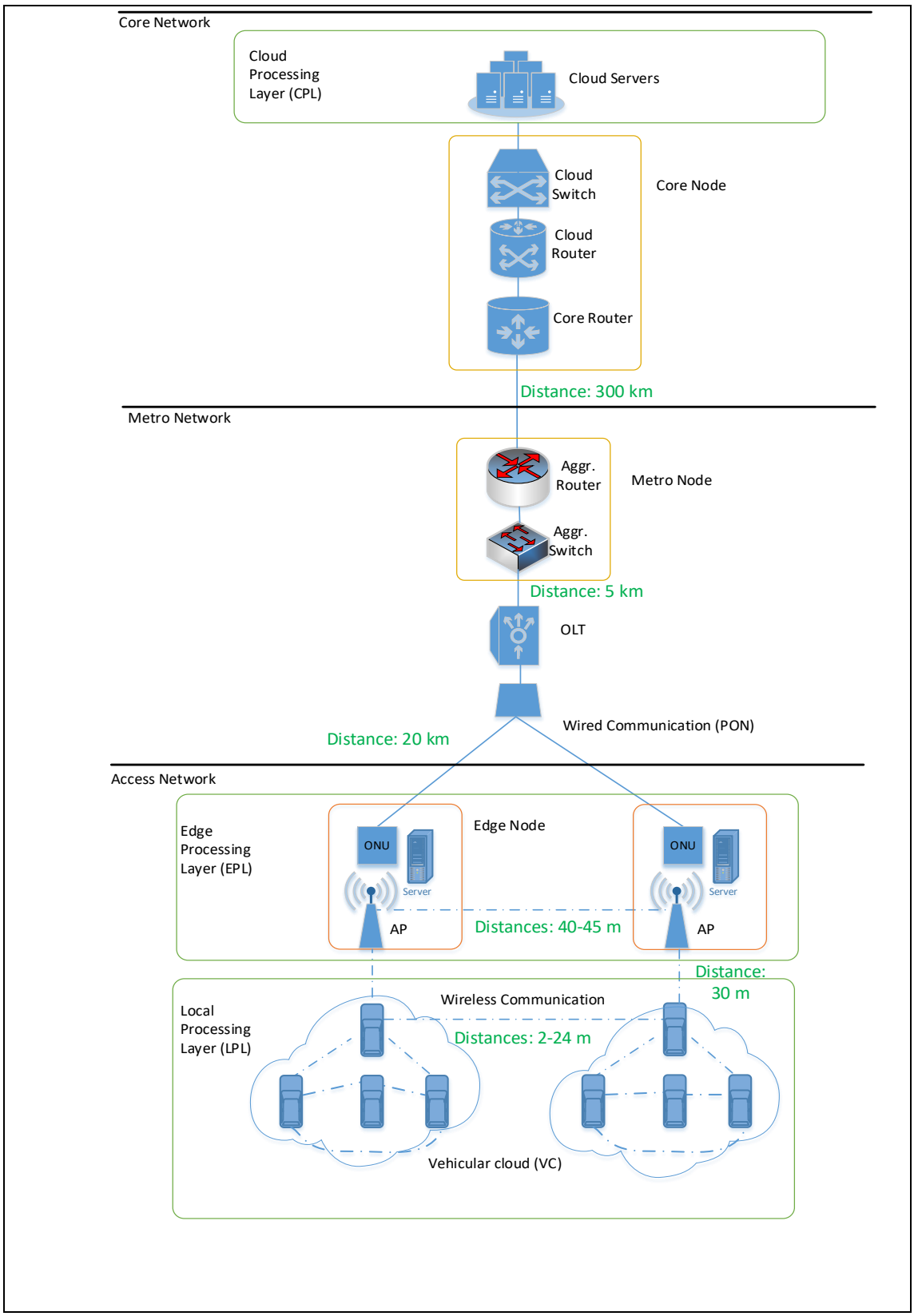

Figure 5.2: End-to-End Vehicular Cloud Architecture

The parameter values for vehicles, edge nodes, cloud, and network devices are the same as Chapter 4 and for completeness are shown again in Table 5-4,Table 5-5,Table 5-6, and Table 5-7 respectively. For the vehicles PUE, the OBU is assumed to be small and efficient enough not to require a ventilation system of any significant power consumption. For the edge node, its three devices are assumed to be collocated in an open area with natural cooling and ventilation. Table 5-7 presents the portion of the idle power consumption attributed to the connected city applications. According to [111], the connected city applications are assumed to account for 13% of the M2M traffic, where the latter accounts for 7% of the global traffic, i.e. $\mathbf{0.07} \times \mathbf{0.13}$. The traffic and processing requirements of demand are based on the estimation in [153] for smart environment applications. For each 2000 MI of processing, 1 Mb of traffic is sent. The packet size used is the Ethernet maximum packet size of 1500 bytes (12 kb).

Table 5-4: Vehicular node parameters

Notation	Value
C_n	2 instructions/cycle × 2 cores × 800 MHz= 3200 MIPS [160]
NM_n	2.712 W [160] [162]
NI_n	1.05007 W [160] [162]
PM_n	OBU = 7.9 W [160]
PI_n	OBU = 3.95 W [160]
B_n	27 Mbps [161]
B_{nm}	DSRC = 27 Mbps, WiFi = 150 Mbps
B_n^{VE}	150 Mbps [163]

K_n	(7.9-3.95)/3200=0.00123 W/MIPS
ϵ	100 pj/bit.m^2 [164]
PUE_n	1

Table 5-5: Edge node parameters

Notation	Value	
C_n	Rasberry Pi	2 instructions/cycle × 4 cores × 1200 MHz = 9600 MIPS [166]
PM_n	Rasberry Pi	12.5 W [166]
PI_n	Rasberry Pi	2 W [166]
K_n	(12.5-2) /9600 = 0.0011 W/MIPS	
NM_n	Access Point	25 W [163]
NI_n	Access Point	5.5 W [163]
B_n	150 Mbps [163]	
B_{nm}	150 Mbps	
NM_n^{ONU}	15 W [167]	
NI_n^{ONU}	13.5 W [167]	
B_n^{ONU}	10 Gbps[167]	
PUE_n	1	

Table 5-6: Cloud parameters

Notation	Value
C_n	4 instructions/cycle × 10 cores × 2.8 GHz = 112000 MIPS [168]
PM_n	115 W
PI_n	57 W (50% of maximum)
K_n	(115--57)/112000=0.000518 W/MIPS
PUE_n	1.1 [116]

Table 5-7:Metro node and core node parameters

Device Type	NM_n (W)	NI_n (W)	NI_n $(\%)$	B_n $(Gbps)$	E_n $(\frac{nj}{bit})$	PUE_n
OLT [171, 172]	1940	60	0.546	8600	0.219	1.5 [173]
Agg. Switch	210	189	1.72	6*40 = 240	0.088	1.5
Agg. Router port	5.25	4.725	0.043	10	0.053	1.5
Core Router port	30	27	0.246	40	0.075	1.5
Cloud router port	30	27	0.246	40	0.075	1.5
Cloud Switch	470	423	3.85	6*100=600	0.078	1.5

Table 5-8: Table of Data Rates

Notation	value
$Dr_{nm}\ \forall n, m\ \in ND$	417 packet/s = 5 Mbps, SR = 27 Mbps
$Dr_{nm}\ \forall n\ \in ND, m\ \in ED$	2,250 packet/s = 27 Mbps, SR = 150 Mbps
$Dr_{nm}\ \forall n\ \in ED, m\ \in (ED\ \cup ND)$	2,250 packet/s = 27 Mbps, SR = 150 Mbps
$Dr_{nm}\ \forall n\ \in ED, m\ \in OLT$	

	7500 packet/s = 90 Mbps, SR = 10 Gbps
$Dr_{nm}\ \forall\ n\ \in OLT, m\ \in MD$	7500 packet/s = 90 Mbps, SR = 10 Gbps
$Dr_{nm}\ \forall\ n\ \in MD, m\ \in CD$	7500 packet/s = 90 Mbps, SR = 10 Gbps

Table 5-8 shows the data rates used in the evaluation of the model. The choices where made based on the following:

1. The WiFi interface between edge nodes and vehicles, and in edge nodes can be used to communicate to 4 vehicles and a neighbouring edge node. Dividing the maximum data rate of 150 Mbps by 5 leads to 30 Mbps per connection. However, due to the use of M/M/1 queueing model, 27 Mbps is chosen as the maximum input traffic rate to prevent the queueing delay from increasing to infinity.
2. For the DSRC, each of the 8 vehicles in the setup can communicate in a P2P fashion with the others. The 5 Mbps is the maximum data rate per connection that can be used to deploy V2V connections, without increasing queueing delay to unbounded levels.
3. For the optical fibres from ONU to the core, we estimated that the data rate assigned to the type of applications being tested is the same as stated in [3], i.e., $(0.07\ of\ M2M \times 0.13\ connected\ city\ traffic)$ of the global traffic.

The following subsections study the allocation of the proposed architecture resources to a single demand of varying size. Figure 5.3 summarises the scenarios evaluated.

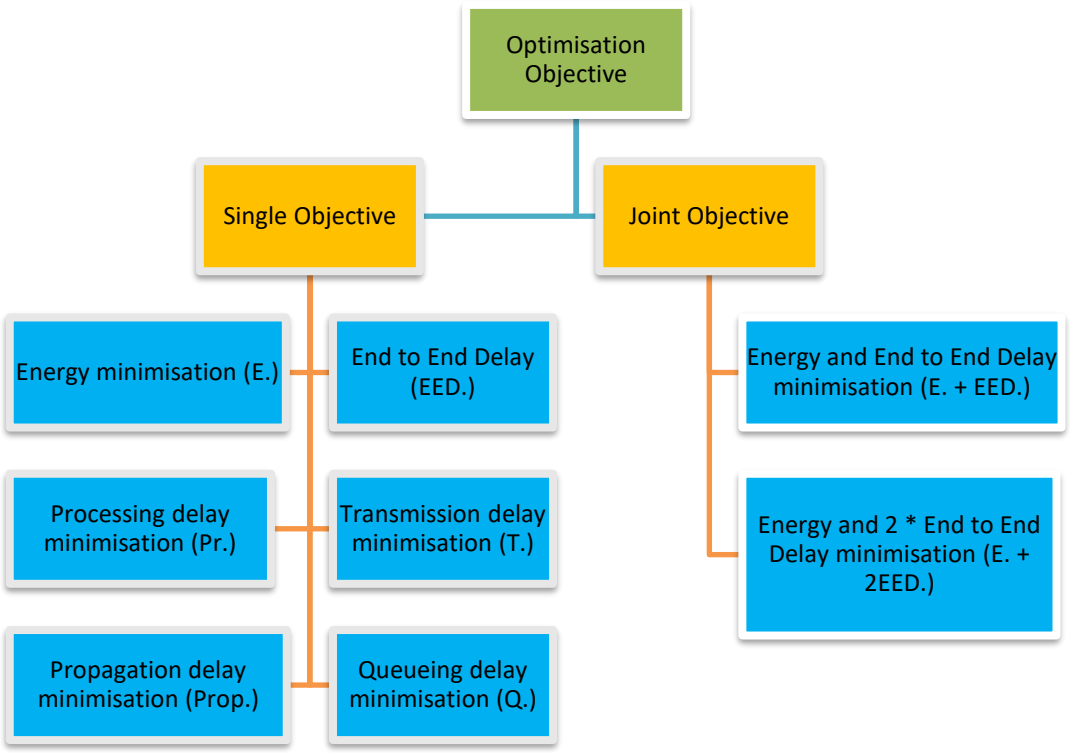

Figure 5.3: Evaluation Scenarios

5.3.1 Single objective optimisation

In this subsection we study the impact of minimising energy consumption and end-to-end delay on the delay performance and energy efficiency of the proposed architecture. We also study scenarios where we minimise delay components individually to obtain further insights into the delay in the architecture.

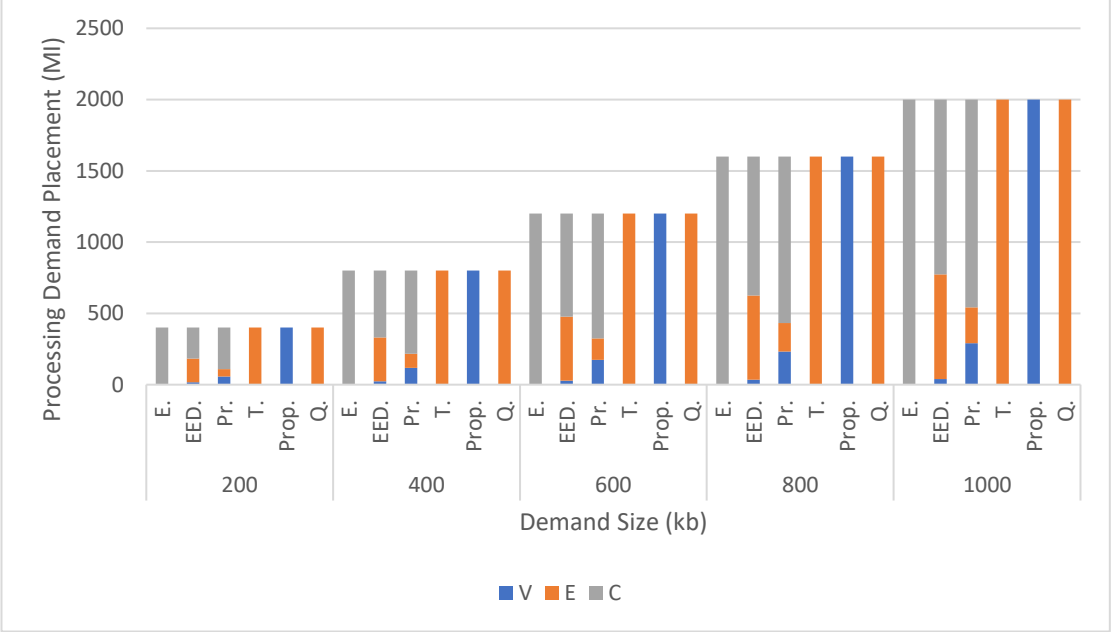

Figure 5.4: Processing demand placement of single objective optimisation scenarios

Figure 5.4 shows the processing demand placement and Figure 5.5 shows the energy consumption of the energy minimisation and delay minimisation scenarios. Figure 5.5 shows that the processing energy dominates in all the scenarios with significant increases for the end-to-end delay minimisation scenario and the processing delay minimisation scenario. The end-to-end delay minimisation has resulted in an average increase of 50% in energy consumption compared to the energy minimisation scenario.

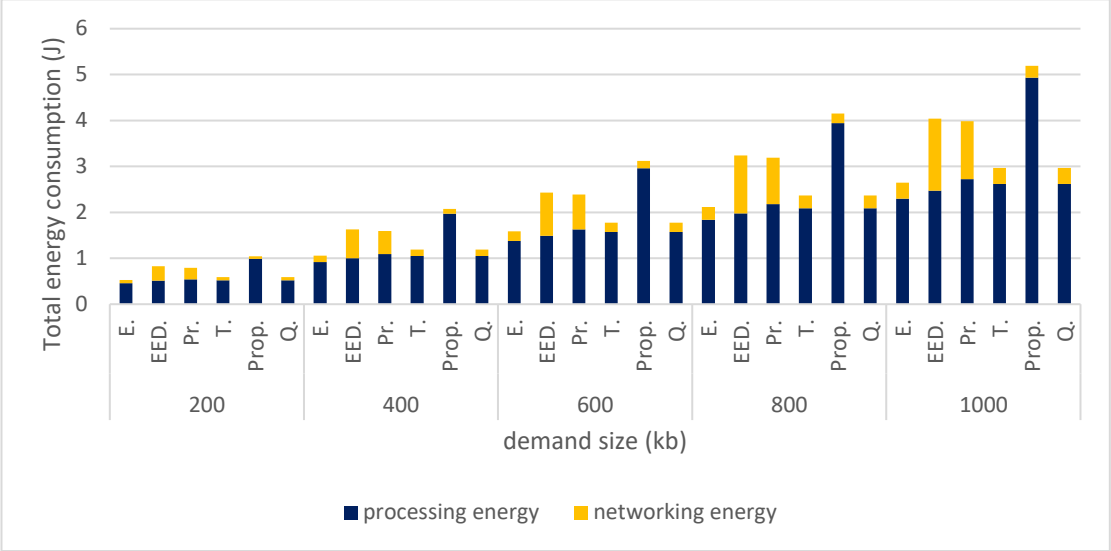

Figure 5.5: Total energy consumption of single objective optimisation scenarios

For further insights into the energy efficiency of the different scenarios, Figure 5.6 and Figure 5.7 show the distribution of processing energy consumption and the networking power consumption, respectively over the architecture layers.

Figure 5.4 and Figure 5.6 show that processing in the cloud is the optimal solution when minimising energy consumption due to the cloud high processing efficiency and processing speed and the low energy consumption of the PON, metro and core networks connecting demand sources to the cloud. The high processing speed of the cloud and high data rate of the network reduce the activation time of servers and network devices and therefore reduces the energy consumption.

To minimise the end-to-end delay, processing is optimally distributed among the three processing layers as seen in Figure 5.6 with on average 59% of the processing performed in the cloud, 38% in the two edge nodes, and 3% in six vehicular nodes. Figure 5.7 shows that distributed processing has resulted in

higher networking energy consumption as traffic is replicated to all processing destinations.

To understand the allocation of processing resources in the end-to-end delay minimisation scenario, we investigate the performance of the scenarios minimising the delay components individually.

Figure 5.6 shows that the processing delay is minimised by a combination of parallel processing in multiple nodes of the three processing layers. The majority of the processing (83.5%) is performed in the cloud, the layer with the highest processing speed layer, followed by processing in the vehicular nodes as there are more vehicular nodes than edge nodes, with 12% of processing in the two edge nodes and 14.5% processing in all the vehicular nodes available for processing.

Figure 5.6 also shows that edge nodes are the optimal processing destinations to minimise transmission delay and queueing delay as these are located within a single hop of the source nodes (vehicular nodes), i.e., retransmission at relay nodes is avoided. To minimise propagation delay, processing is performed in the vehicular nodes as they are the closest to the demand source.

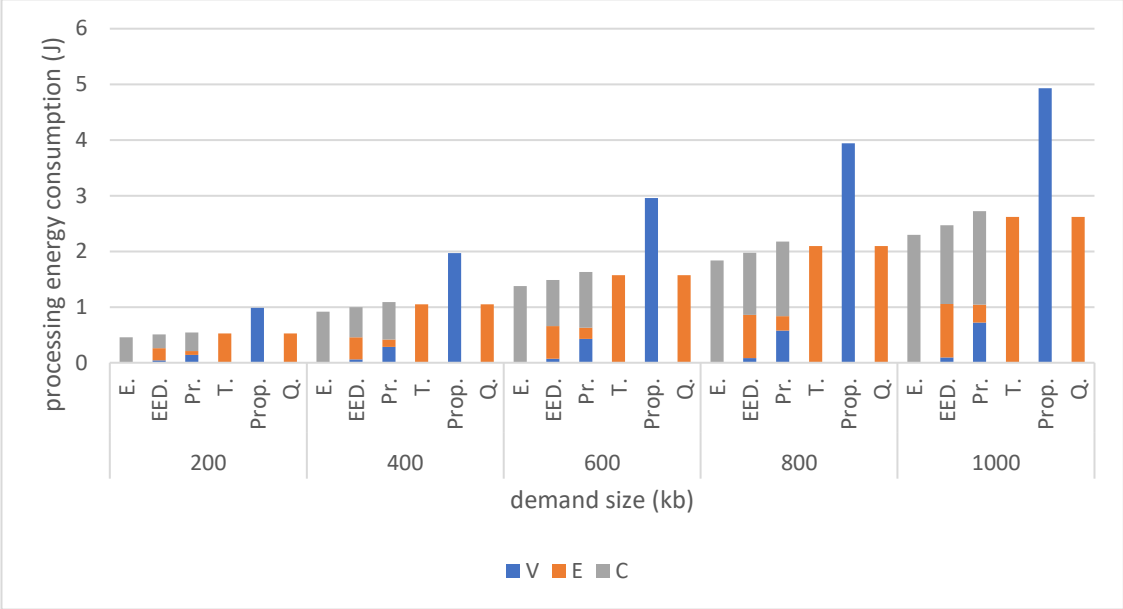

Figure 5.6: Processing energy distribution for single objective optimisation

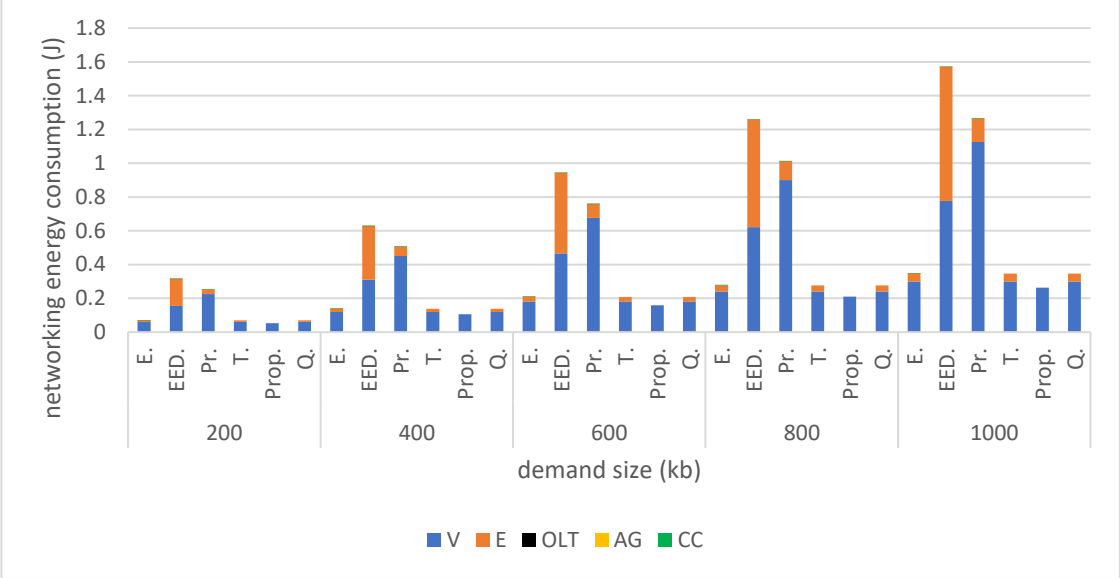

Figure 5.7: Networking energy distribution for single objective optimisation

Figure 5.8 shows the end-to-end delay performance of the different scenarios. As explained in Section 5.2, the end-to-end delay is given as the maximum end-to end delay experienced by the demand splits. Therefore, the components of the end-to-end delay in Figure 5.8 represent the components of the maximum end-

to end delay. The transmission delay and the processing delay, as seen in Figure 5.8, dominate over the other delay components. Therefore, the end-to-end delay is minimised by parallel processing to reduce processing delay with the majority of the processing performed in the edge nodes to reduce transmission delay, and cloud to reduce processing delay, while vehicular nodes received the smallest share of processing as seen in Figure 5.6.

Figure 5.8 also shows that the end-to end delay of the energy minimisation scenario where processing is performed in the cloud is comparable to that of the end-to-end minimisation scenario where processing is distributed over the three processing layers. The increase in end-to end delay is limited to 7%.

This is due to the high processing speed of the cloud and high data rate of the PON, metro and core networks connecting the cloud to the demand sources.

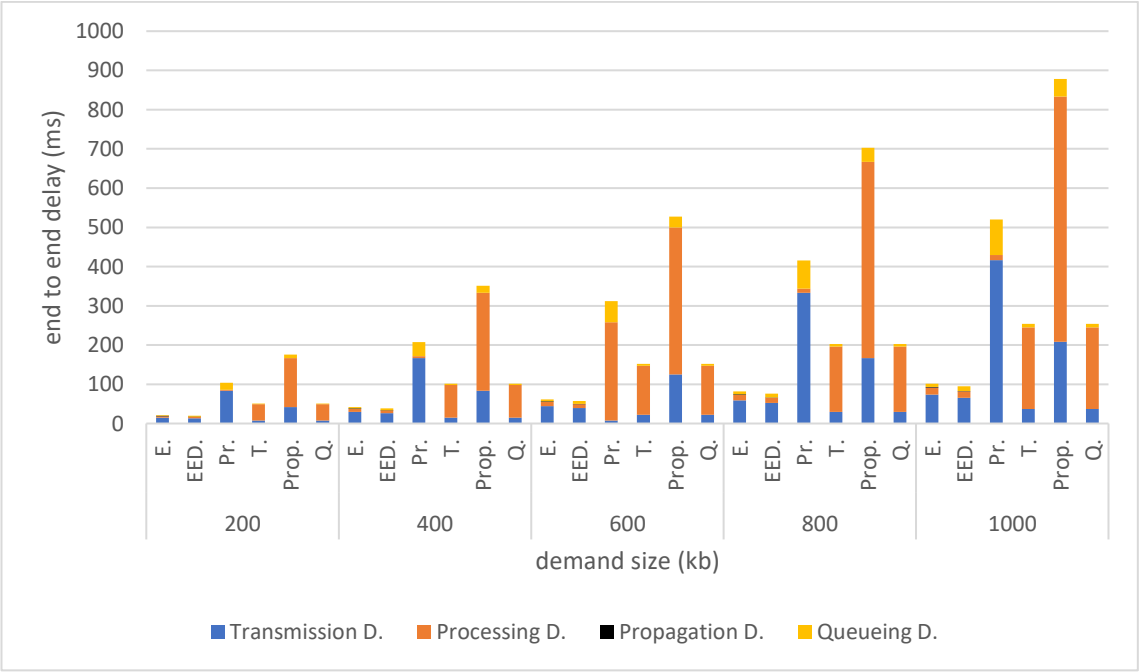

Figure 5.8: End to End Delay for single objective optimisation

5.3.2 Joint Minimisation of Energy and End to End Delay

In the previous subsection, we studied the impact of minimising energy consumption and minimising end-to-end delay on the delay performance and energy efficiency, respectively. In this subsection, we study the joint minimisation of energy consumption and end-to-end delay. We start by investigating a scenario where energy consumption and delay are of equal importance. To set the values of the weighting factors:

1. We obtain the total energy consumption of the energy consumption minimisation scenario, $\sum_{n\in N} W_n$. For example, from Figure 5.5, minimising energy for demand size 200 kb consumed 0.53 J and for demand size 1000 kb the energy consumed is 2.65 J.
2. We obtain the total delay of the end-to-end delay minimisation scenario, $\sum_{n\in N} \mathcal{L}_n$, As an example, from Figure 5.8, the minimisation of EED resulted in 20 ms for 200 kb demand and 94.8 ms for the 1000 kb demand, respectively.
3. We set γ to 1 (unitless) and find the value of δ (J/ms) that satisfies $\delta\,\mathcal{L}_n = \mathrm{W_n}$. This gives energy consumption and end-to-end delay equal weights in the objective function. So, for demand 200 kb, $\delta\,(20) = 0.53$, so $\delta = \frac{0.53}{20} = 0.0265\,J/ms$. For demand 1000 kb, $\delta\,(94.8) = 2.65$, so $\delta = \frac{2.65}{94.8} = 0.0279\,J/ms$.

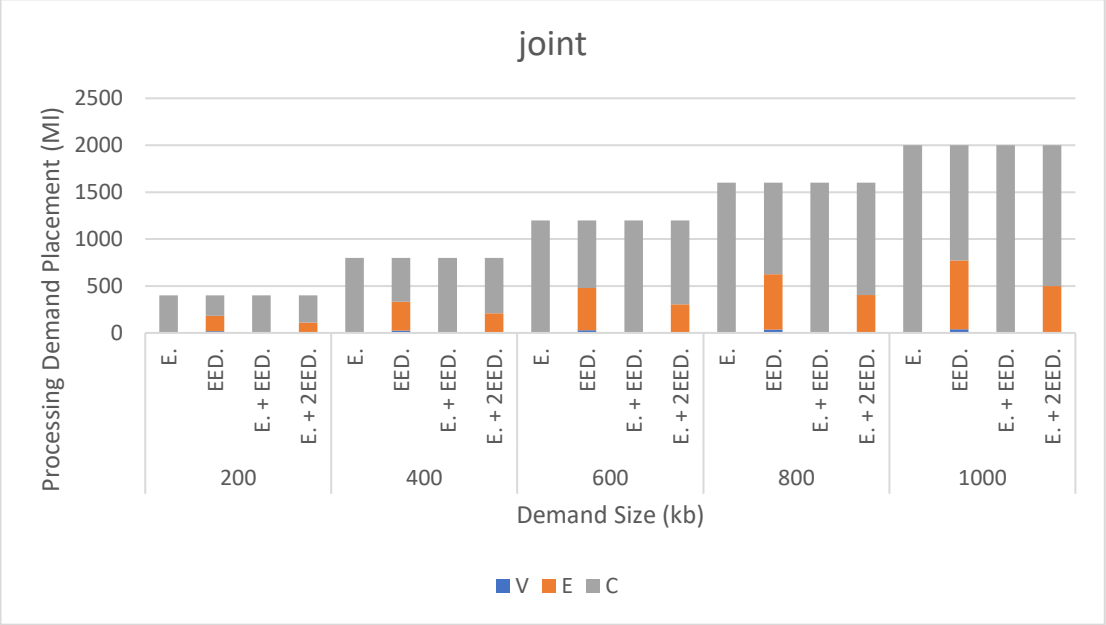

Figure 5.9: Processing demand placement of joint objective optimisation scenarios

Figure 5.9 - Figure 5.13 show that the joint Optimisation with equal weights for energy consumption and delay (E+EED) has yielded the same processing allocation, processing in the cloud, as the energy consumption minimisation scenario. This is because moving part of the processing to the edge nodes and vehicular nodes will lead to power consumption increase that outweighs the reduction in delay. The end-to-end delay of the joint optimisation scenario increased by 7% compared to the delay minimised scenario.

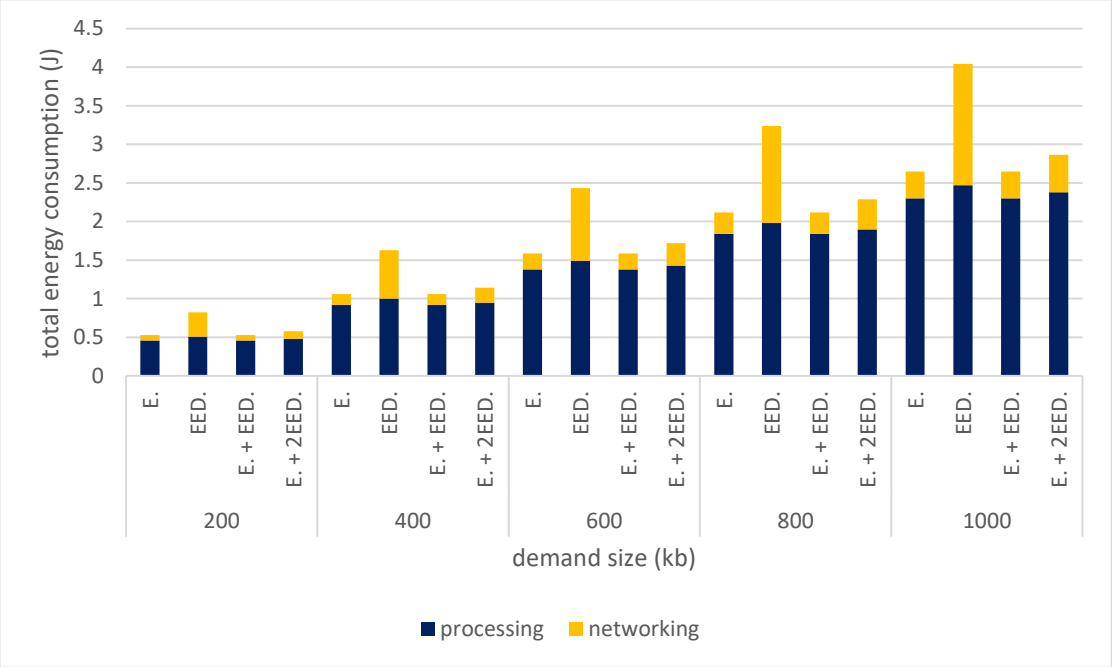

Figure 5.10: Total energy consumption of joint objective optimisation scenarios

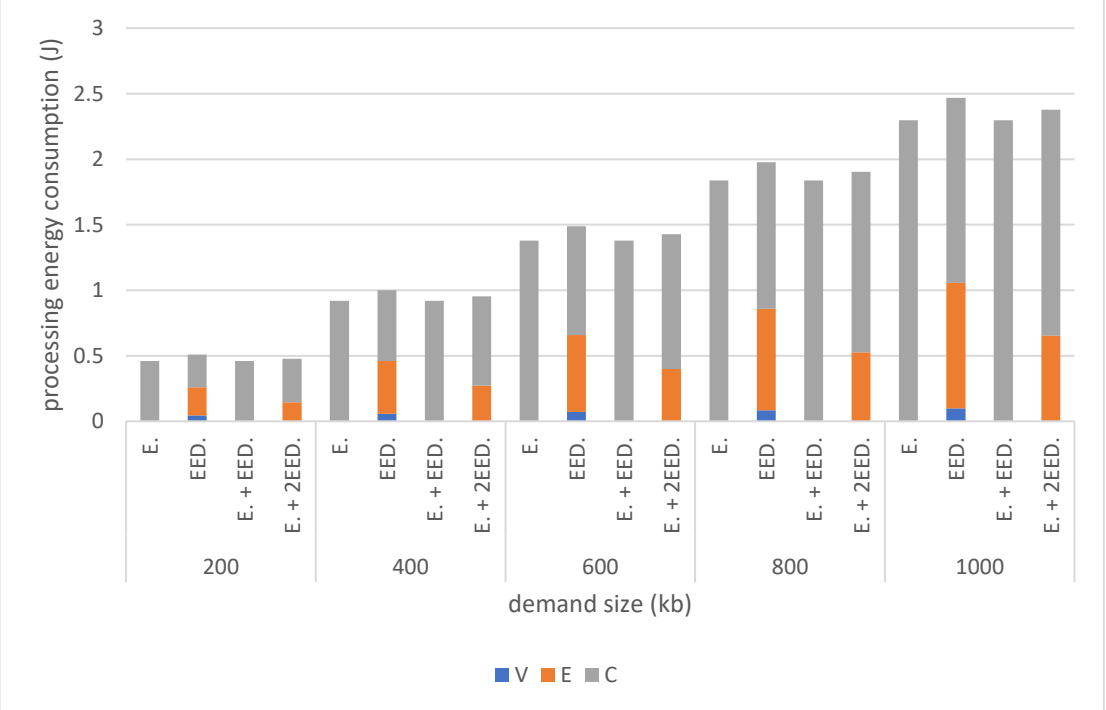

Figure 5.11: Processing energy distribution with joint objective Optimisation

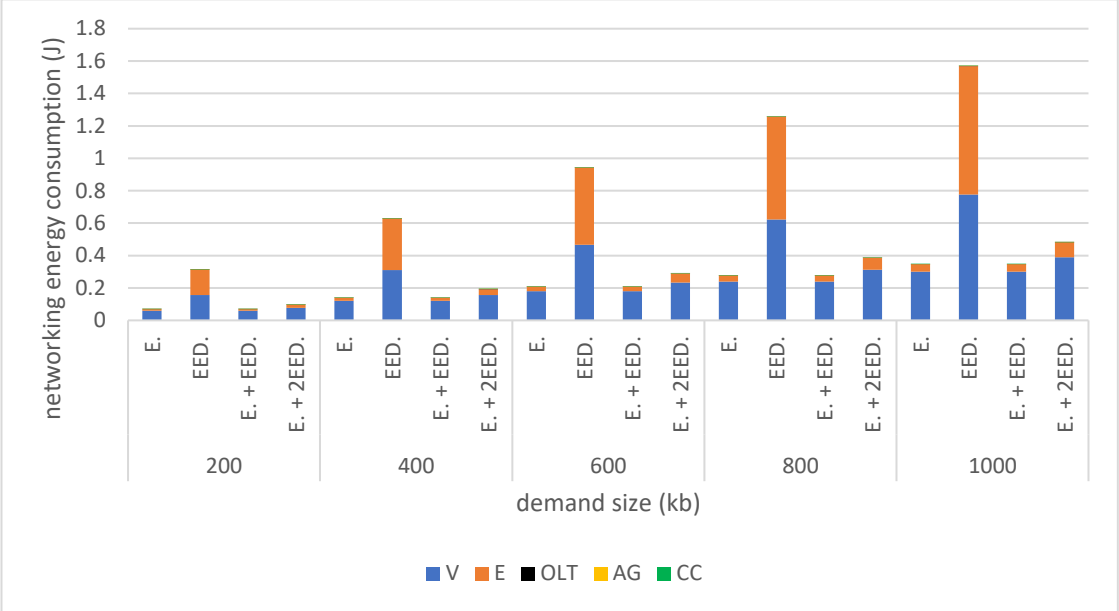

Figure 5.12: Networking energy distribution with joint objective Optimisation

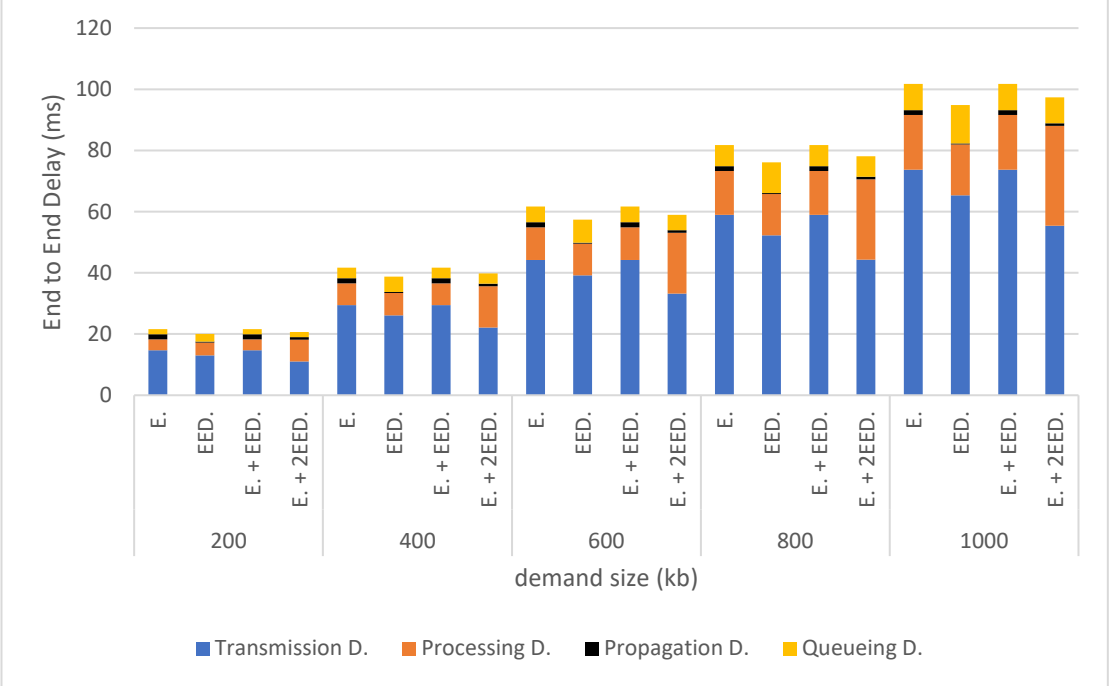

Figure 5.13: End to End Delay with joint objective Optimisation

To understand more the relation between the energy and the delay in the joint optimisation, we considered another case to investigate the joint minimisation of energy consumption and delay of 'delay sensitive applications' by prioritising delay over energy consumption in the model by setting the weighting factors to satisfy $\delta \mathcal{L}_n = 2\mathrm{W_n}$. Using the same examples as before, in this case for demand 200 kb, $\delta = 2 * \frac{0.053}{20} = 0.0053\, J/ms$, and for demand 1000 kb, $\delta = 2 * \frac{2.65}{94.} = 0.056\, J/ms$

Increasing the significance of delay (E+2EED) has resulted in moving some of the processing to the edge nodes, as seen in Figure 5.9, to reduce transmission delay, the dominating delay component as seen in Figure 5.13. This comes at the cost of increasing processing delay but yields 4% reduction in end-to-end delay. The increase in energy consumption is limited to 8% as seen in Figure 5.10.

5.3.2.1 Comparison with State-of-art model

Optimisation of latency in VC is an active research topic. However, the comparison between different results is not a straightforward process as the paradigm is not standardised and allows for open assumptions and interpretations in terms of architecture and processing and networking resources. The works of [139, 174, 176, 177] can be viewed in the same light as the work of this chapter since they share the objective of latency minimisation. However, most of them were concerned with the VC/fog level without inclusion of conventional cloud, or they defined the delay differently. We find the work of [48] the closest in its handling of the resources allocation problem to our work. An

algorithm was developed to optimally allocate demands in a three-layer architecture. The optimality has been evaluated by running a fitness model on each layer and then choosing the fittest of the three. We summarise the similarities as:

1. Having three layers architecture: Vehicular Cloud, Roadside cloudlet (edge level), conventional cloud level.
2. Having the objective of delay and energy minimisation
3. Similar mathematical formulations specially in the calculation of the delay (propagation, transmission, processing, queueing) and energy (networking, processing).

The differences can also be summarised as follows:

1. Even though the demand can be split; all parts of the demand must be served in the same cloud layer (VC or cloudlet or CC).
2. Only a single hop route from the client to the appointed processing destination.
3. The vehicles are mobile.
4. The actual formulation of the multi-objective function is different.

In Table 5-9, we present a comparison between our values and the state-of-art for one demand size (we choose 1000 kb, which is the one that had approximately similar number of packets to one demand in [48]).

Table 5-9: Comparison of State-of-the-Art results for joint minimisation

	This thesis works (demand: 833 packets)		State-of-art (demand: 781 packets)	
Objective	E. + EED.	E. + 2EED.	Multi-objective (placement in any of the three layers)	Multi-objective (placement in vehicular layer)
Energy (J)	26	28.5	40	100
EED. (ms)	1000	960	200	720
EED. Per packet (ms/packet)	1.2	1.15	0.256	0.921

5.3.3 MILP Results Verification

In Chapter 4, the heuristic results verified the results of the MILP model. In this chapter we verify the results by conducting validation test cases and show that the MILP model generated an optimal solution. We chose 3 checkpoints for three different objectives, for which the optimal solution can by manually verified, energy minimisation, propagation delay minimisation, and transmission delay minimisation. Figure 5.14 and Figure 5.15 show the calculated energy and delay for the optimal solution and the results obtained from the MILP.

The three checkpoints are:

1. checkpoint # 1: Energy minimisation with a demand of 1 Mb and 2000 MI
    - Optimal Solution: Allocation of the Whole demand in the cloud
2. checkpoint # 2: Propagation delay minimisation while allocating a demand of 200 kb and 400 MI.
    - Optimal Solution: Allocation in the nearest vehicle
3. checkpoint # 3: Transmission delay minimisation while allocating a demand of 1 Mb and 2000 MI.
    - Optimal Solution: Allocation in the edge node associated with the VC

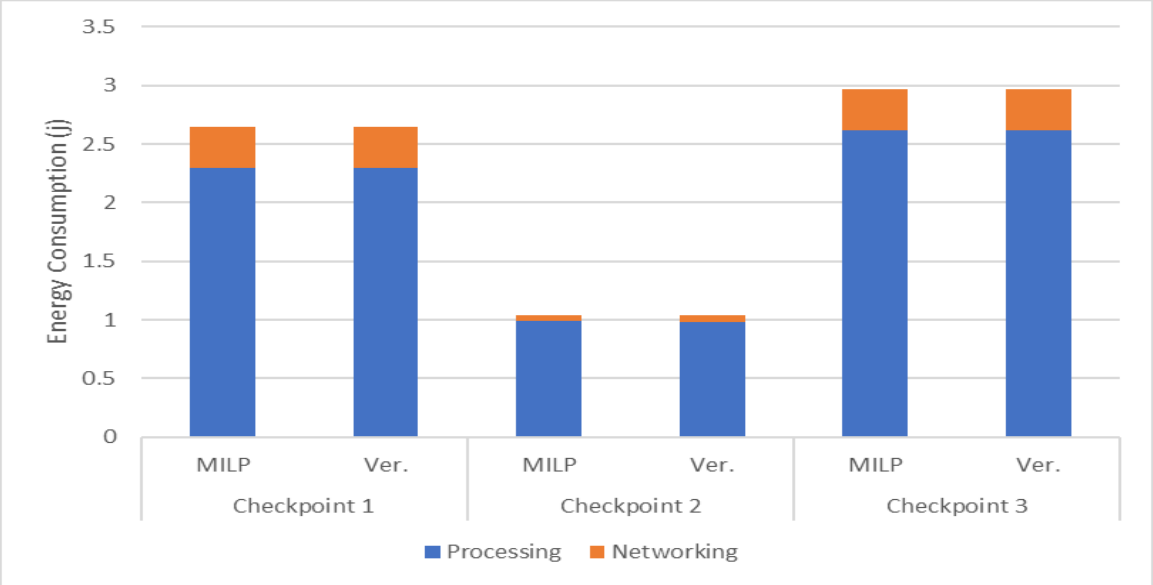

Figure 5.14: Energy consumption (MILP vs. Analytic Verification)

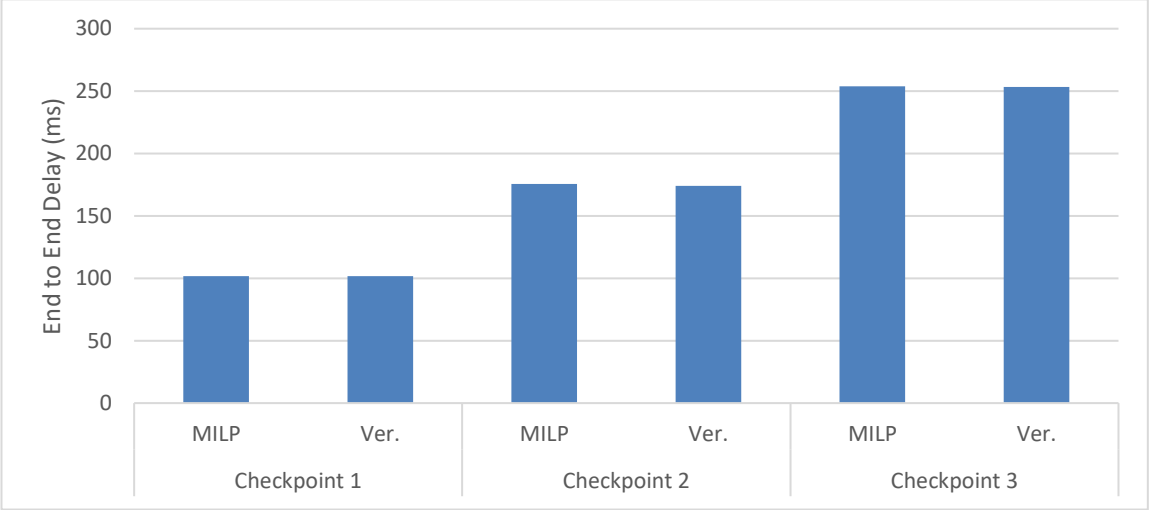

Figure 5.15: End to End Delay (MILP vs. Analytic Verification)

5.4 Summary

This chapter presented a study of the energy consumption and delay performance of on-demand smart city applications. A MILP model was developed to minimise the energy consumption and end-to-end delay.

The end-to-end delay accounted for transmission delay, processing delay, propagation delay and queueing delay. The results showed that processing in the cloud is optimal when minimising energy consumption of on-demand applications which yielded only 7% increase in end-to-end delay compared to the delay minimisation scenario. For delay minimisation, processing is optimally distributed over the three processing levels which increased the processing energy consumption. Joint optimisation of energy consumption and delay maintained the energy efficiency of the energy minimisation scenario while limiting the increase in end-to-end delay to 7% compared to the delay minimisation scenario.

Chapter 6
Resilient Vehicular Cloud Architecture

6.1 Introduction

The ad hoc topologies of vehicular clouds are ever changing as vehicles come and go, raising questions regarding the availability, QoS and reliability of such networks. In the previous chapters, vehicles were assumed to be available for the whole demand service period, i.e., no service disruption or failure. However, to truly make vehicular clouds usable and desirable, mechanisms are needed to make the architecture resilient and able to overcome intermission of service due to vehicles leaving vehicular clouds.

Resilience can be implemented based on two main approaches, *reactive resilience*, and *proactive resilience*. In the reactive resilience, measures are taken to ensure undisrupted services *as an event occurs* (i.e., a vehicle leaving). This is the more challenging mechanism due to the need to find alternative processing resources in real time. Live virtual machine migration can be viewed as an implementation of this type of resilience. In the context of vehicular clouds, the topic is still in its infancy but is attracting more interest from researchers [66, 68-70]. For proactive resilience, mechanisms are implemented to provide advanced backup for every demand being served in vehicles, so whenever a vehicle leaves a replacement is readily available. Resilience is tackled for vehicular networks to ensure reliability and QoS but it remains long way from being thoroughly investigated. The authors in [178] provided a strategy for fault tolerance

redundant job assignment to vehicles in a parking lot. Theoretical analysis was compared to simulation results for job completion time. The work did not consider a vehicular cloud architecture or optimised the allocation of the jobs. The quality of service of an infrastructure-less vehicular cloud architecture was studied in [179]. In the paper, an optimisation model was developed that minimised the response time to complete a job and maximised the probability of success, which is measured by the percentage of completed jobs before a vehicle leaves the cloud. The work does not concern the power consumption and the reliability is provided through the strategy developed for job assignment. In [180], a scheme for reliable real-time streaming in vehicular environment was developed. The approach maximises the utilisation of cloud and fog computation resources for real-time content delivery to vehicles. Different from our work, in the architecture, vehicles are receivers of the services rather than providers. Recent work in [181] produced a resilient clustering algorithm for the formation of more stable vehicular clouds. It used metrics such as delay and throughput to evaluate in comparison with other clustering algorithms. The above-mentioned efforts Mostly focused on of the resilient and reliable vehicular clouds in light of delay and minimised failure probability. The relationship of resilience and power consumption of vehicular cloud architecture is not widely investigated.

In this chapter, we extend our model introduced in Chapter 4 to have a resilient architecture in the context of processing destinations redundancy. We explore how this view of resilience would impact the power consumption of the architecture.

6.2 Resilient Vehicular Cloud Architecture

As discussed before, a demand generated by a vehicle has three levels of processing. Edge nodes and clouds are assumed to be able to finish the job allocated to them with high probability due to the resilience inherited from their fixed nature. The same cannot be said about the vehicular nodes, which are mainly under the control of the vehicle owner. A vehicle can leave the area, or the owner can simply decide to stop sharing the vehicle computational resources. Therefore, measures should be introduced to handle such intermittent service. In this chapter we extend the MILP model in Chapter 4 to improve the resilience of the vehicular cloud using a proactive resilience approach. To this end, redundancy schemes are introduced to have multiple destinations assigned to a demand instead of single destination. A main (primary) destination is chosen in addition to a redundant (backup) destination to support the primary and replace it, if the primary choice is no longer available. As stated before, the edge and cloud nodes are fixed entities that can complete an assigned job, therefore, no backup is required if any of them is chosen. The vehicles availability is questionable. Therefore, for every vehicle serving a processing demand (or portion of processing demand), a *backup* processing node is allocated. The choice of backup node can be an edge node or a cloud. Choosing another vehicle as a backup is a poor choice as it suffers the same concerns of reliability and availability of the primary processing vehicle, admittedly with lower power consumption than an edge or a cloud. The backup node receives the full traffic demand.

Even though having a resilient architecture is always desirable, it is expected to increase power consumption due to the activation of backup nodes. In an effort to reduce the impact of resilience measures on the architecture power consumption, we propose two mechanisms:

1. **Active Processing Resilience (APR)**

In this mechanism, the backup node performs the computation in parallel with the primary node (vehicle), i.e., the backup node does not wait for the primary node (vehicle) to leave to proceed with processing. If the associated primary node (vehicle) leaves, the backup node will carry on with processing and deliver the results without delay. This approach results in minimum level of service disruption [70]. It is, however, the most power consuming as the job is ran concurrently on two nodes.

2. **Idle Processing Resilience (IPR)**

In this mechanism, a backup node receives the demand (both processing and traffic). However, the backup node only performs processing if the primary node (vehicle) leaves the network. In addition to the networking power consumption induced by the traffic delivered from the source node, the backup node server consumes idle power when receiving and safe keeping the demand traffic. This mechanism results in service delays as the backup node starts processing after the primary node (vehicle) leaves.

6.3 MILP Model

The model in Chapter 4 is extended to introduce resilience. Table 6-1 and Table 6-2 contain the additional parameters and variables needed.

Table 6-1: Additional parameters defined to model a resilient architecture

Act	Act = 1 if active resilience is supported, otherwise Act = 0

Table 6-2: Additional variables defined to model a resilient architecture

$bg\Omega_{sgd}$	The processing demand of source node s, primary processed in vehicle g, and backed up in node d, $s, d \in N$, $g \in ND$ (MIPS)
$b\Omega_{sd}$	The amount of processing demand of source node s backed up in node d, $s, d \in N$ (MIPS)
bF_{sd}	Traffic from source node s delivered to backup node d. $s, d \in N$ (Mbps)
$b\lambda_{nm}^{sd}$	The traffic demand between source node s to backup node d passing through link (n, m) where $s, d, n, m \in N$ (Mbps)
$bg\alpha_{sgd}$	$Rg\alpha_{sgd}$ = 1 if node d is back up for primary vehicle g which serves source node s, otherwise $Rg\alpha_{sgd}$ = 0. $s, d \in N$, $g \in ND$
$b\alpha_{sd}$	$b\alpha_{sd}$ = 1 if node d is back up for node s, otherwise $b\alpha_{sd}$ = 0. $s, d \in N$

The objective of the model is to minimise the architecture power consumption by optimising the allocation of processing resources to primary and backup processing of demands and routing of traffic. The total power consumption (TP) of the architecture is given as:

$$TP = \sum_{n \in N} W_n \quad (6.1)$$

where

$$W_n = PUE_n(WN_n + WP_n) \quad \forall\, n \in N \quad (6.2)$$

In the following we give the power consumption of the different nodes in the resilient architecture.

For vehicular nodes:

$$WN_n = \beta_n^{NET}\ NI_n + \sum_{s \in N} \sum_{d \in N} \sum_{m \in Nm_n} (\lambda_{nm}^{sd} + b\lambda_{nm}^{sd})\ T_{nm} + \sum_{s \in N} \sum_{d \in N} \sum_{m \in Nm_n} (\lambda_{mn}^{sd} + b\lambda_{mn}^{sd})\ R_{mn} \quad (6.3)$$

$$\forall\, n \in ND$$

For edge nodes:

$$WN_n = \beta_n^{NET}\, NI_n + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n \cap (ND\cup ED)} (\lambda_{nm}^{sd} + b\lambda_{nm}^{sd})\, T_{nm} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n \cap (ND\cup ED)} (\lambda_{mn}^{sd} + b\lambda_{mn}^{sd})\, R_{mn} + \beta_n^{ONU}\, NI_n^{ONU} + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n \cap (OLT)} (\lambda_{nm}^{sd} + \lambda_{mn}^{sd} + b\lambda_{nm}^{sd} + b\lambda_{mn}^{sd}) E_n \quad (6.4)$$

$$\forall\, n \in ED$$

For OLT, metro, core nodes:

$$WN_n = \beta_n^{NET}\, NI_n + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n} (\lambda_{mn}^{sd} + b\lambda_{mn}^{sd})\quad E_n \quad (6.5)$$

$$\forall\, n \in OLT \cup MD \cup CD$$

Equations (6.3)-(6.5) give the networking power consumption of each layer in the architecture. The traffic traversing a node consists of primary traffic meant for primary processors and backup traffic meant for the backup processors.

Processing power consumption:

The processing power consumption of a node in the architecture is given in equation (6.6), considering the processing idle power consumption and the processing load dependent power consumption. This equation distinguishes

between the APR and the IPR mechanisms. For APR (Act = 1), the power consumption of primary and backup processing load is included in the calculation. For IPR (Act = 0), only the processing power consumption of the primary load is accounted for, and the backup impact would only be reflected in the idle power as will be seen in the constraints.

$$WP_n = \beta_n^{Pr} PI_n + \sum_{s \in N} (\Omega_{sn} + Actb\Omega_{sn})\, K_n \; \forall\, n \in N \quad (6.6)$$

Some constrains from Chapter 4 are modified to represent a resilient architecture as follows:

$$\sum_{\substack{s \in N \\ s \neq d}} \Omega_{sd} + b\Omega_{sd} \leq C_d \quad \forall\, d \in N \quad (6.7)$$

Constraint (6.7) replaces constraint (4.13) to ensure that the processing capacity of a node is not exceeded. This should hold for both APR and IPR, as idle backup resources should be on standby.

$$\sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n} (\lambda_{nm}^{sd} + b\lambda_{nm}^{sd}) + \sum_{s \in N} \sum_{d \in N} \sum_{m \in N m_n} (\lambda_{mn}^{sd} + b\lambda_{mn}^{sd}) \leq B_n \quad (6.8)$$

$$\forall\, n \in OLT \cup MD \cup CD$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap ND}(\lambda_{nm}^{sd}+\ b\lambda_{nm}^{sd})\ +\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap ND}(\lambda_{mn}^{sd}+\ b\lambda_{mn}^{sd}) \tag{6.9}$$

$$\leq\ B_n\quad \forall\ n\ \in ND$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap ED}(\lambda_{nm}^{sd}+\ b\lambda_{nm}^{sd})$$

$$+\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap ED}(\lambda_{mn}^{sd}+\ b\lambda_{mn}^{sd})\ \leq\ B^{VE} \tag{6.10}$$

$$\forall\ n\ \in ND$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap (ND\cup ED)}(\lambda_{nm}^{sd}+\ b\lambda_{nm}^{sd})\ +\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap (ND\cup ED)}(\lambda_{mn}^{sd} \tag{6.11}$$

$$+\ b\lambda_{mn}^{sd})\ \leq\ B_n\quad \forall\ n\ \in ED$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}(\lambda_{nm}^{sd}+\ b\lambda_{nm}^{sd})$$

$$+\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}(\lambda_{mn}^{sd}\ +\ b\lambda_{mn}^{sd})\ \leq\ B^{ONU} \tag{6.12}$$

$$\forall\ n\ \in ED$$

Constraints (4.18) – (4.22) are modified to Constraints (6.8) - (6.12) which ensure that the primary and backup traffic are always within the networking capacity of each node.

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n}(\lambda_{nm}^{sd} + b\lambda_{nm}^{sd}) + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n}(\lambda_{mn}^{sd} + b\lambda_{mn}^{sd}) \geq \beta_n^{NET} \quad \forall\, n \in N \tag{6.13}$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n}(\lambda_{nm}^{sd} + b\lambda_{nm}^{sd}) + \sum_{s\in N}\sum_{d\in N}\sum_{m\in Nm_n}(\lambda_{mn}^{sd} + b\lambda_{mn}^{sd}) \leq A\beta_n^{NET} \quad \forall\, n \in N \tag{6.14}$$

Instead of equations (4.25) and (4.26), equations (6.13) and (6.14) set a binary variable to 1 for nodes used in networking, i.e. transmit, receive, or relay primary or backup traffic.

$$\sum_{\substack{s\in N\\ s\neq d}}\Omega_{sd} + \sum_{\substack{s\in N\\ s\neq d}} b\Omega_{sd} \leq \beta_d^{PR} \quad \forall\, d \in N \tag{6.15}$$

$$\sum_{\substack{s\in N\\ s\neq d}}\Omega_{sd} + \sum_{\substack{s\in N\\ s\neq d}} b\Omega_{sd} \geq A\,\beta_d^{PR} \quad \forall\, d \in N \tag{6.16}$$

Another binary variable is set to 1 in (6.15) and (6.16), modified from (4.27) and (4.28), to indicate nodes used for primary and/or backup processing.

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}(\lambda_{nm}^{sd} + b\lambda_{nm}^{sd}) + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}(\lambda_{mn}^{sd} + b\lambda_{mn}^{sd}) \geq \beta_n^{ONU} \quad \forall n \in ED \tag{6.17}$$

$$\sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}(\lambda_{nm}^{sd} + b\lambda_{nm}^{sd}) + \sum_{s\in N}\sum_{d\in N}\sum_{m\in N m_n\cap OLT}(\lambda_{mn}^{sd} + b\lambda_{mn}^{sd}) \leq A\,\beta_n^{ONU} \quad \forall n \in ED \tag{6.18}$$

A binary variable is set to 1 in (6.17) and (6.18), modified from (4.29) and (4.30), to indicate the use of the ONU in an edge to relay primary or backup traffic.

The following constraints are added for a resilient architecture:

$$\sum_{d\in N} bg\Omega_{sgd} = \Omega_{sg} \quad \forall s \in N, \forall g \in ND \tag{6.19}$$

Constraint (6.19) ensures that portions of a processing demand primarily processed in a vehicle are backed up.

$$b\Omega_{sd} = \sum_{g\in ND} bg\Omega_{sgd} \quad \forall s,d \in N \tag{6.20}$$

Constraint (6.20) calculates the amount of backup processing at a node coming from a certain source.

$$bg\Omega_{sgd} \geq bg\alpha_{sgd} \qquad \forall s,d \in N, \forall g \in ND, s \neq d \quad (6.21)$$

$$bg\Omega_{sgd} \leq A\, bg\alpha_{sgd} \qquad \forall s,d \in N, \forall g \in ND, \qquad s \neq d \quad (6.22)$$

Constraints (6.21) and (6.22) set a binary variable to indicate the primary and back up nodes of a demand.

$$\sum_{d \in N} bg\alpha_{sgd} = 1 \qquad \forall s \in N, g \in ND \quad (6.23)$$

To preserve resources, constraint (6.23) indicates that demands primarily processed in a vehicle need a single back up.

$$b\Omega_{sd} \geq b\alpha_{sd} \qquad \forall s,d \in N, s \neq d \quad (6.24)$$

$$b\Omega_{sd} \leq A\, b\alpha_{sd} \qquad \forall s,d \in N, s \neq d \quad (6.25)$$

In constrains (6.24) and (6.25), a binary variable is set to indicate the backup node of a demand.

$$b\alpha_{sd} + \alpha_{sd} = 1 \ \forall s,d \in ND, s \neq d \quad (6.26)$$

Constraint (6.26) states that a vehicle cannot be primary and backup for the same demand.

$$bF_{sd} = V_s b\alpha_{sd} \ \forall s,d \in N \quad (6.27)$$

Constraint (6.27) ensures that backup nodes receive the full traffic demand from source nodes

$$\sum_{m \in Nm_n} b\lambda_{nm}^{sd} - \sum_{m \in Nm_n} b\lambda_{mn}^{sd} = \begin{cases} bF_{sd} & if\ n = s \\ -bF_{sd} & if\ n = d \\ 0 & otherwise \end{cases} \quad (6.28)$$

$$\forall s,d,n \in N$$

Constraint (6.28) gives the flow conservation constraint. It ensures that the amount of back up traffic received by a relay a node is equal to the amount re-transmitted.

6.4 Scenarios Studied and Results

To evaluate the energy efficiency of the resilient vehicular cloud architecture, we consider a scenario similar to that of Chapter 4. We restate the details of the evaluation scenario for the reader's convenience. We consider 16 vehicles parked in a small parking area. The distance between two vehicles ranges from 2 meters to 24 meters. The parking lot is surrounded by 4 edge nodes, placed at

an average distance of 30 meters away from vehicles. Both DSRC and WiFi have communication ranges of several hundreds of meters [16, 158]. Each edge node serves a vehicular cloud of 4 vehicles.

The demands in this work are generated by the vehicles and the traffic demand (Mbps) and processing demand (MIPS) of a request are related, based on the estimation in [153] for smart environments applications. On average, one Mbps is needed for each 2000 MIPS of processing demand. The model is run to optimise processing placement over the three processing layers (VEC).

In total, there are nine cases to depict different primary and backup choices, as follows:

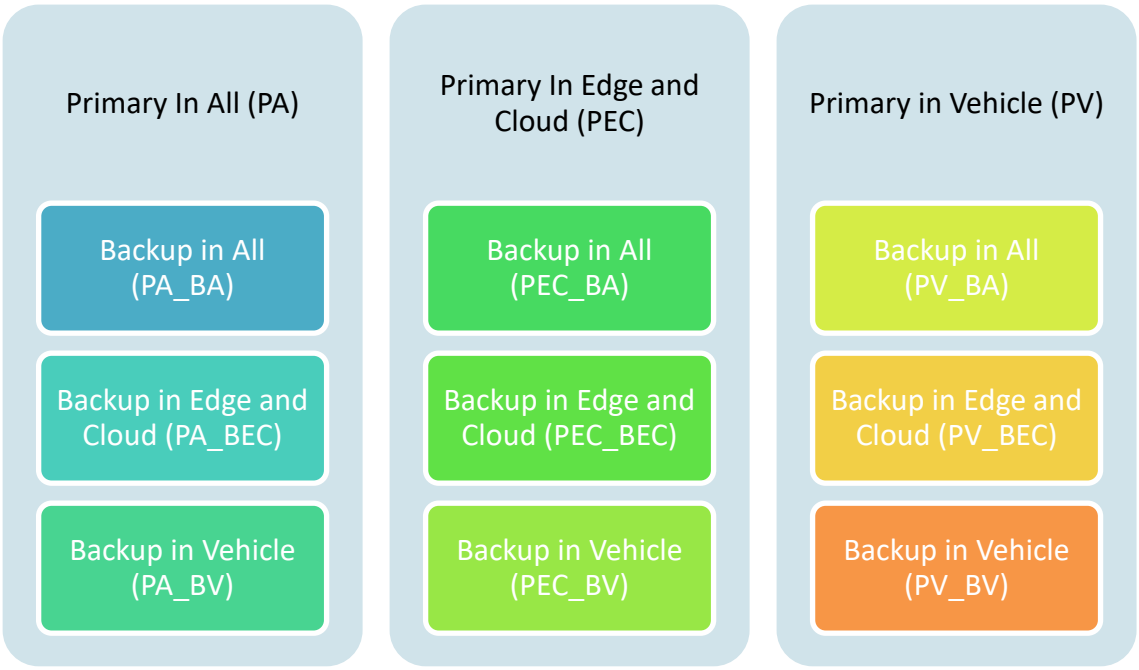

Figure 6.1: All possible scenarios for the choice of primary and backup nodes in the resilient vehicular cloud model

From Figure 6.1, we make the following observations:

1. For the middle column in which the primary processing node is an edge node or a cloud (PEC), the cases are irrelevant as no backup is needed.
2. The cases with backup specified in vehicles, are not considered as we stated vehicles are a poor choice for backup. Therefore, the cases where they are explicitly chosen is excluded.
3. In the right column, primary processing is done in vehicles (PV). This is the case in which resilience measures are crucial. As a backup choice, we are left with two options, backup in All (PV_BA) and backup in edge and cloud (PV_BEC). The later provides the highest level of resilience and the former provides a comparison in terms of processing energy efficiency appeal for backup.
4. The cases in the left column with primary in all levels (PA) are also included in our evaluation to study the impact of the choices made for the primary and backup nodes on the power consumption in comparison with the non-resilient model from Chapter 4. The model must make the decision if it is more efficient to use the lower power consuming vehicles as primary, or to avoid using vehicles to eliminate the need for backup.

From the above, four cases are considered in the MILP model evaluation, as summarised in Figure 6.2.

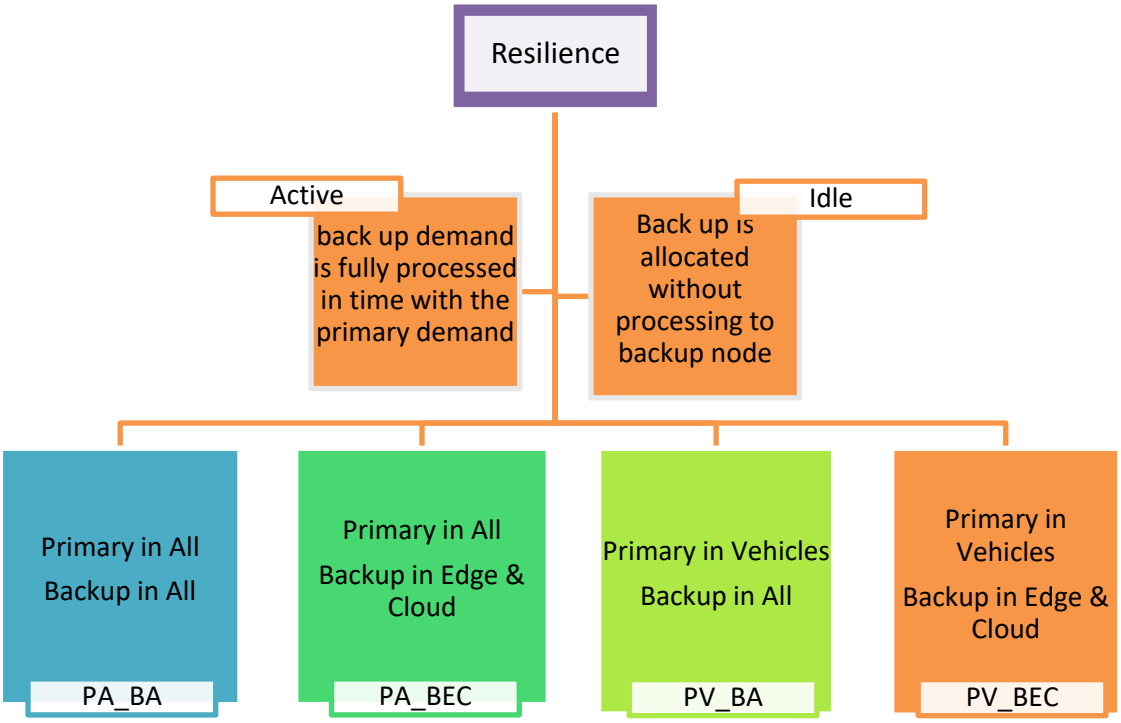

Figure 6.2: Resilient Vehicular Cloud Architecture Scenarios

The results evaluate the merits of the four described scenarios and compare them to the VEC scenario with no resilience mechanism, which were obtained in Section 4.3.1 for a single demand case and in Section 4.3.4 for the multiple demands case. Each scenario is tested under two cases in terms of number of requests and their sizes. The first case is a single demand of varying size, and the second case is multiple demands with three levels of demand size, low, medium, and high.

The parameters for each device are the same as the ones used in Chapter 4

6.4.1 Single Demand

6.4.1.1 Active Processing Resilience (APR)

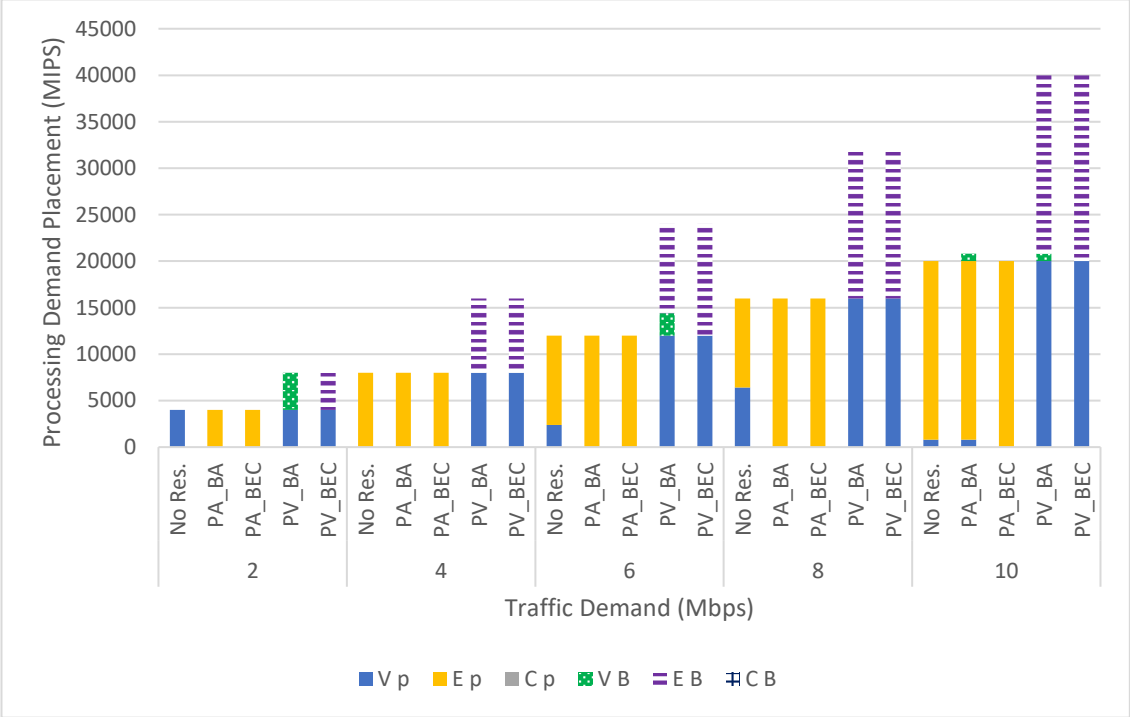

Figure 6.3: Processing demand placement of APR scenarios for single demand

Figure 6.3 shows the processing demand placement in each of the resilience scenarios with active mechanism. The use of vehicles as primary nodes has retreated when resilience is considered. When the use of the vehicles is forced, the edge nodes are the preferred backup nodes.

Figure 6.4 shows the total power consumption in each resilience scenario with the active mechanism. It shows that limiting primary allocation to vehicles caused the highest increase in power consumption, when compared with the *no resilience* (No_Res) scenario. It also shows that for the PA_BA and PA_BEC, the networking power consumption was the dominating factor in the power rise.

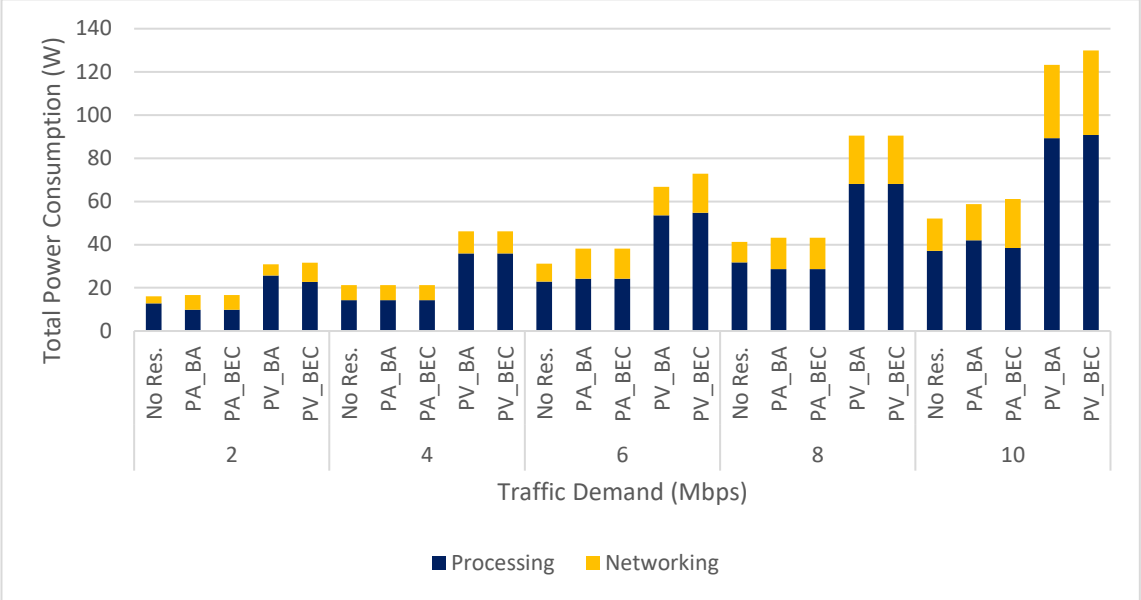

Figure 6.4: Total power consumption of APR scenarios for single demand

Figure 6.5 details the processing power consumption. In the PA_BA and PA_BEC scenarios processing is optimally allocated in edge node for most demand sizes. This way, the need for back up is avoided. The processing power consumption does not show much increase over the No_Res scenario, and for some (2 Mbps and 8 Mbps), the processing power consumption is lower than the No_Res. As the demand grows (10 Mbps), the use of vehicles became preferable for lower power consumption, even with added power used in the backup in PA_BA.

For the PV_BA and PV_BEC, forcing the use of the vehicles for primary processing increases the processing power consumption for two reasons. First, the need for more vehicles to serve the demand, i.e., activating more nodes. Second, the unavoidable need for back up in these scenarios. The slight difference between PV_BA and PV_BEC is due to the possibility in PV_BA to use

some of the vehicles in the backup, which has lower power consumption than edge and cloud.

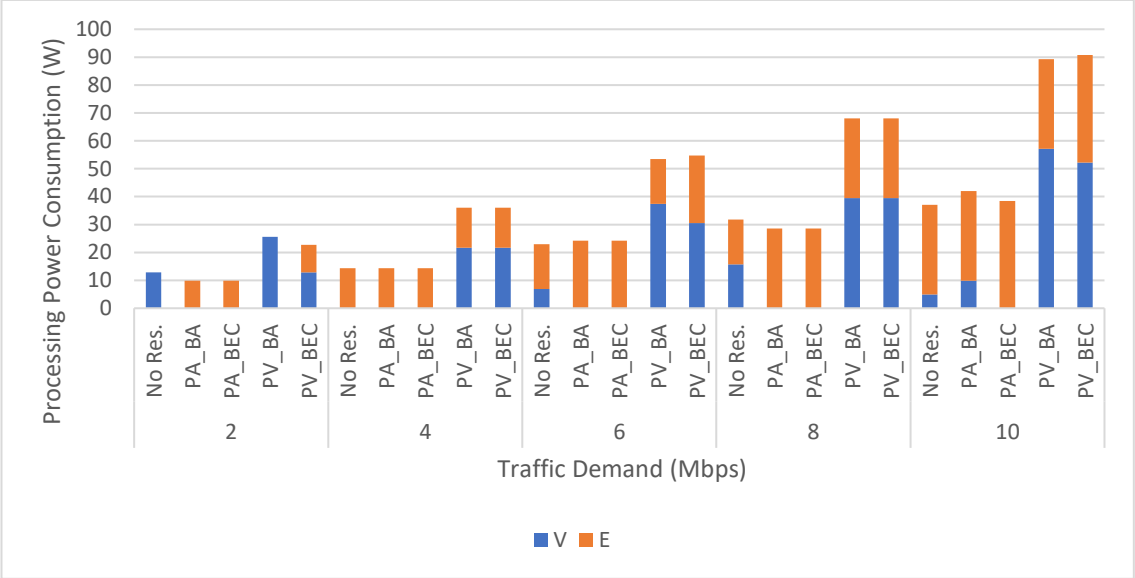

Figure 6.5: Processing power consumption of APR scenarios for single demand

Figure 6.6 shows the networking power consumption. The figure shows increase in the networking power consumption of the edge node in PA_BA and PA_BEC scenarios compared to the No-Res scenario, as the edge node is used to avoid the need for back up, and the load-dependent distance-dependent networking power consumption per bit is higher for the edge layer than the vehicles layer. For the PV_BA and PV_BEC scenarios, the networking power consumption is affected in many ways. The sole use of vehicles as primary destinations requires activating more vehicles to accommodate the demand size. Hence, the traffic demand would have more replicas to be sent to the assigned primary destinations, increasing the networking power. Also, the backup traffic would increase the networking power consumption further. Furthermore, the increased

traffic would exhaust networking capacity and force the vehicles to use both the Wi-Fi and DSRC interfaces.

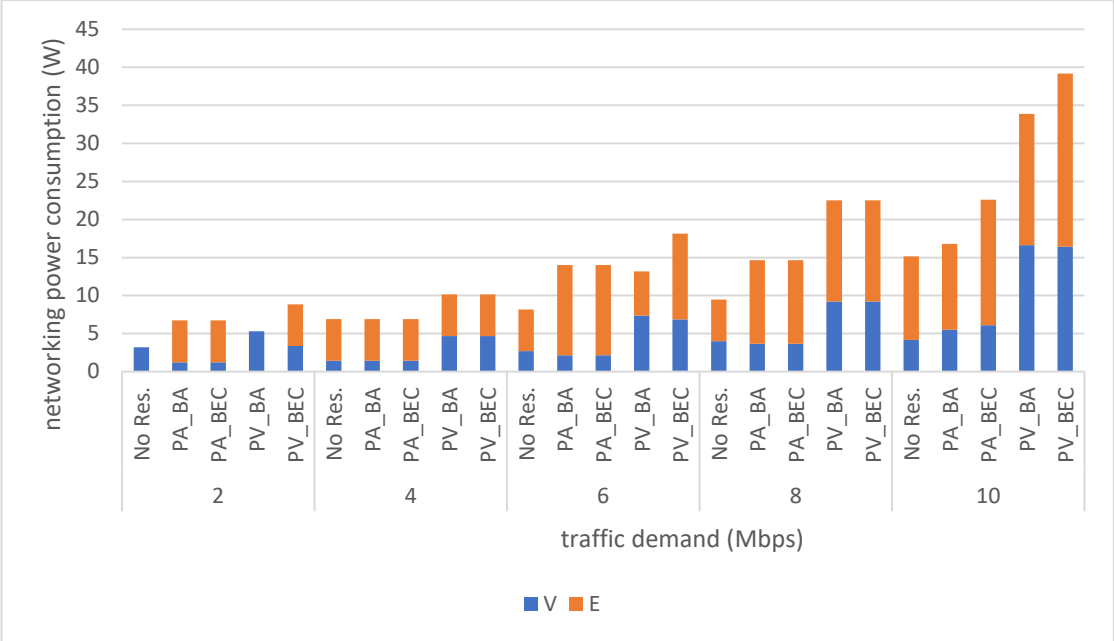

Figure 6.6: Networking power consumption of APR scenarios for single demand

Figure 6.7 shows the increase in power consumption of the active resilience scenarios in comparison with the No_Res scenario. The PA_BA and PA_BEC scenarios have resulted in a maximum increase of 23% as the processing is moved to the edge nodes to avoid the need for backup as opposed to the No-Res scenario where processing is distributed among vehicle and edge nodes

In the PV_BA and PV_BEC scenarios where backup is needed, power consumption is increased by a maximum of 149% compared to the No_Res scenario. The increase in power consumption is a function of the number of additional nodes to be activated and the communication interface used for the backup traffic. Note that the improved resilience of the PV_BEC scenario is

obtained at a cost of 3.4% average increase in power consumption compared to the PV_BA scenario.

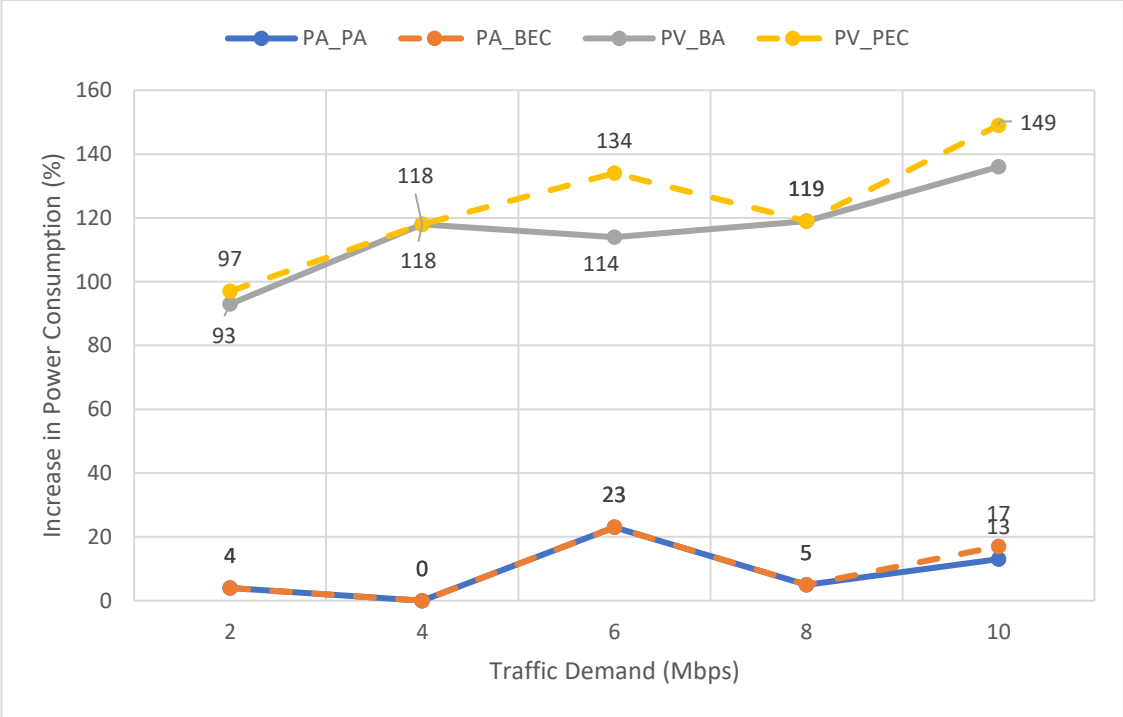

Figure 6.7: increase in power consumption of APR scenarios for single demand

6.4.1.2 Idle Processing Resilience (IPR)

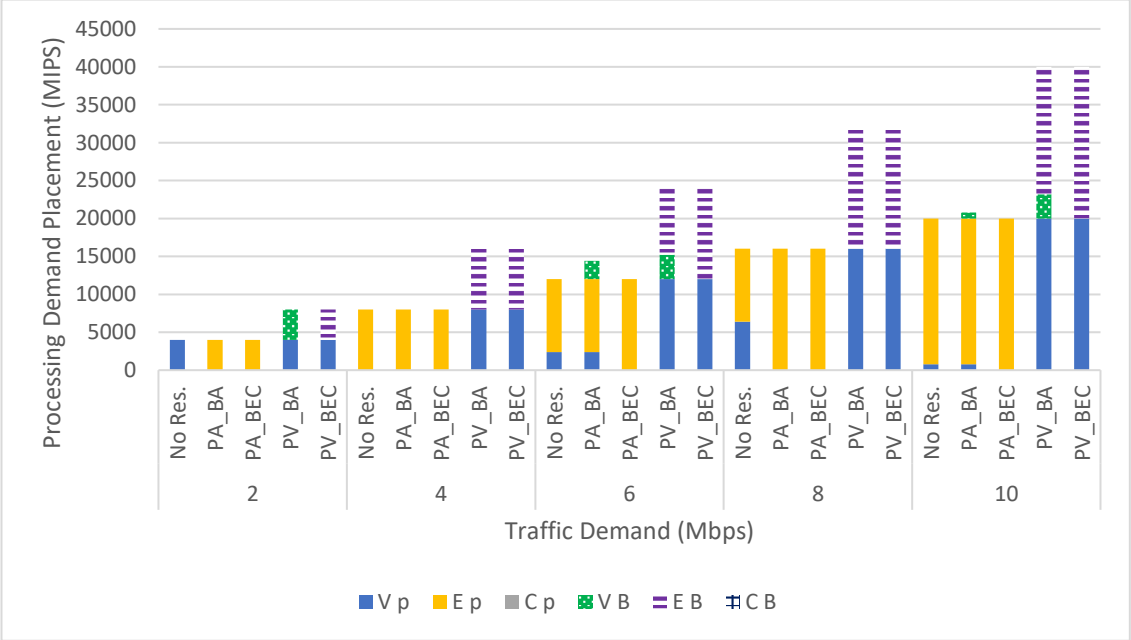

Figure 6.8: Processing demand placement of IPR scenarios for single demand

Figure 6.8 shows the processing demand placement using the IPR mechanism. The distribution is mostly like the APR except for the 6 Mbps and 10 Mbps demands, in which more vehicles have been assigned as backup nodes.

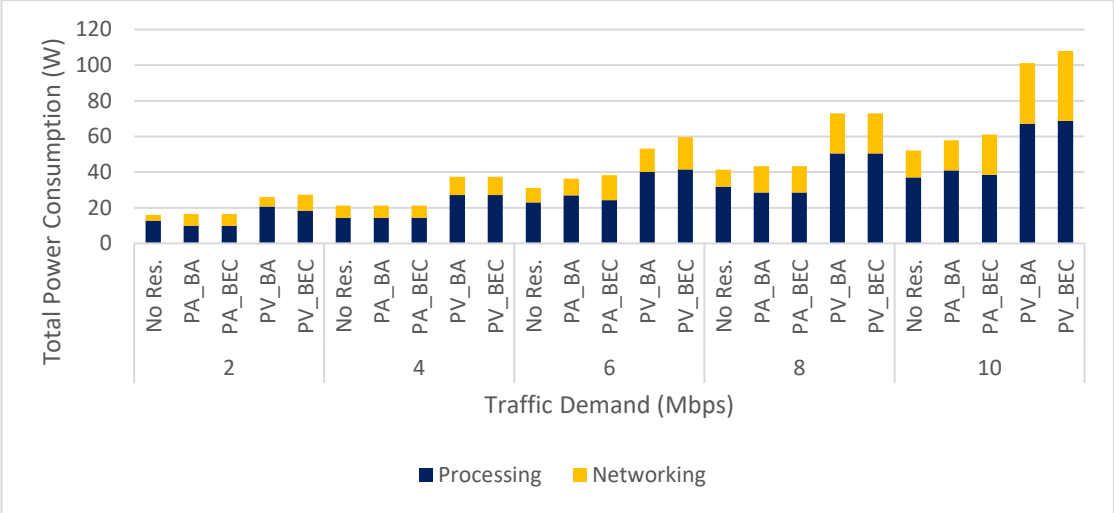

Figure 6.9: Total power consumption of IPR scenarios for single demand

Figure 6.9 shows the total power consumption for each IPR scenario. The IPR has been beneficial mainly for the PV_BA and PV_BEC scenarios where primarily processing is limited to vehicles. Figure 6.10 and Figure 6.11 show respectively the processing and networking power consumption. Figure 6.10 shows similar trends to those observed for the active scheme in Figure 6.5. The reduction in backup power consumptions in the idle scenario has not made vehicles attractive as processing destinations for the PA-BA and PA-BEC scenarios. Processing in these scenarios with the idle scheme for most of the traffic sizes is performed in the edge to avoid the need for backup.

For the 6 Mbps demand with the PA_BA scenario, a slight increase is seen in processing power consumption for the idle scheme compared to the active scheme, as well as the use of vehicles in processing as opposed to using only the edge node in the active case. Comparing the networking power consumption for this specific case (Figure 6.6 vs Figure 6.11) shows reduction in the networking power consumption for the active case with the choices made for processing allocation. The PV_BA and PV_PEC scenarios maintain the same trends as in the active scheme. However, the processing power consumption is significantly reduced by keeping backup processing in idle state, and comparison with active case (Figure 6.5 vs Figure 6.10), shows more contribution of vehicles in the processing power consumption in the idle scheme.

Figure 6.11 shows that the networking power consumption was not affected by the switch to the idle resilience, apart from the PA-BA scenario for 6 Mbps, as the processing destinations have not changed compared to the active resilience.

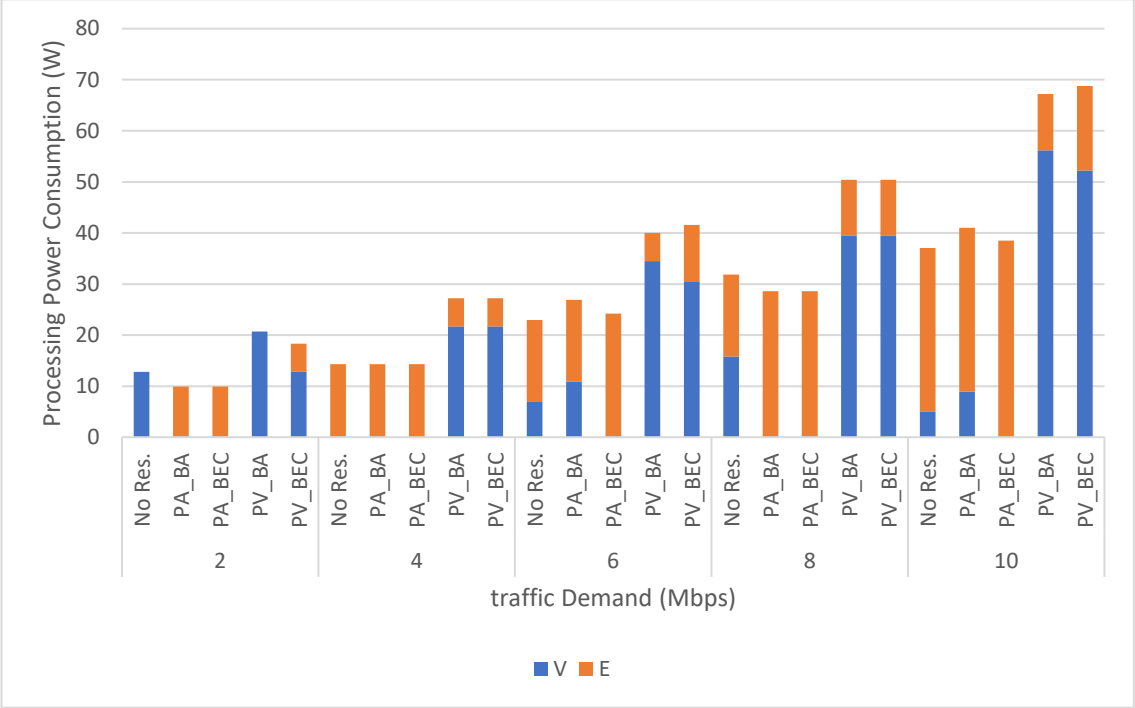

Figure 6.10: Processing power consumption of IPR scenarios for single demand

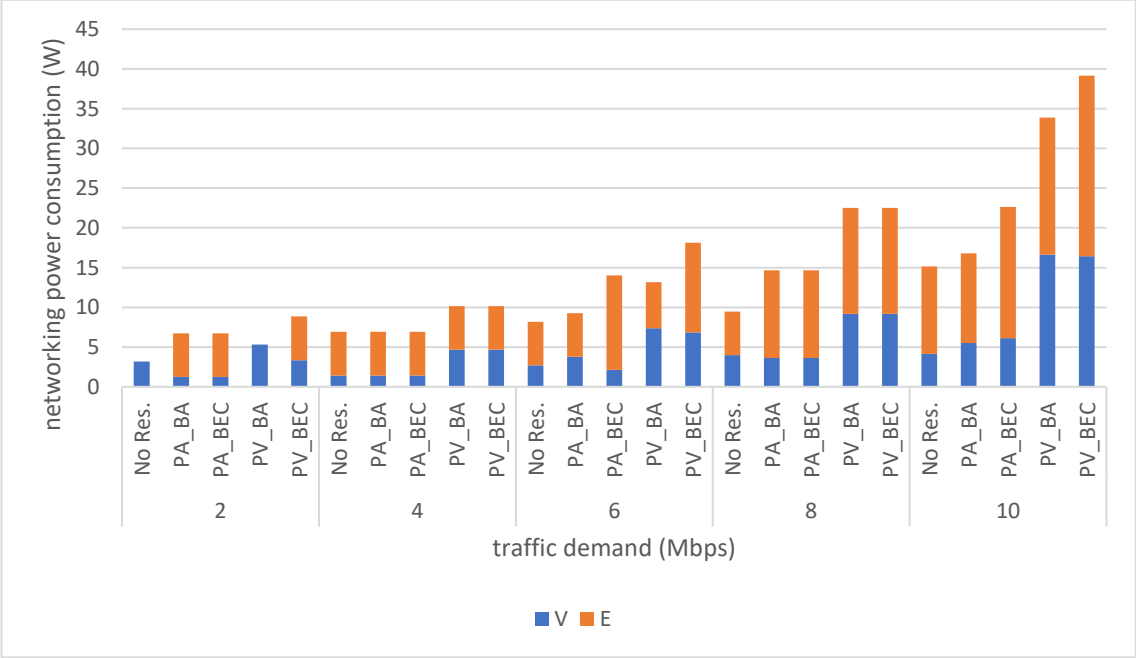

Figure 6.11: Networking power consumption of IPR scenarios for single demand

The increase in power consumption of the idle resilience scenarios compared to No_Res scenario is shown in Figure 6.12. As expected, putting backup processing resources in idle state reduces the increase in power consumption compared to the power consumption increase resulting from the active resilience. For the PV-BEC scenario the increase in power consumption is reduced by an average of 32%.

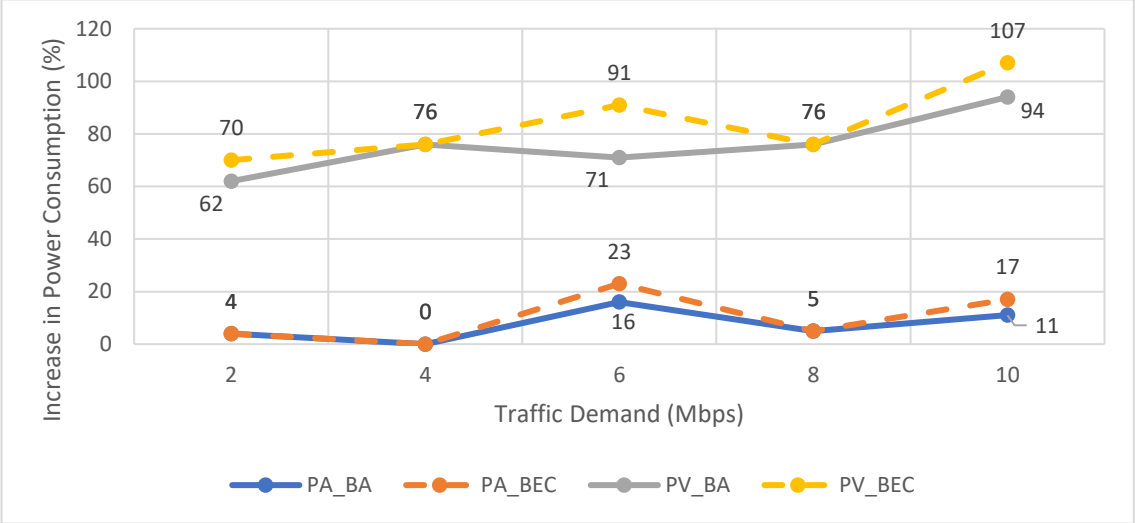

Figure 6.12: Increase in power consumption of IPR scenarios for single demand

6.4.1.3 Impact of vehicles leaving

Even though mobility is not in the scope of our work, stationary VC can still change its topology and resources capacity by vehicles leaving. As distributed processing greatly depends on the available resources, a new test case where the number of vehicles is changing has been used to evaluate the impact of that change on the choices on the primary and backup allocation decision and the power consumption. The model was run with reduced number of vehicles with APR and using PV_BA and PV_BEC scenario, as those two are the ones that

are directly affected by the number of vehicles available. Also, PA_PA was used to show the impact of the change on more flexible scenario. In the previous test case, 16 vehicles were present, with one of them generating the demand and 15 for processing. This number is reduced while trying to allocate a demand of 10 Mbps and 20000 MIPS. Figure 6.13 shows the demand placement for the PV_BA. In this scenario as vehicles start leaving the car park, the model places the primary demand in the next available car, which might be at longer distance and increases the networking power consumption slightly. The placement of the backup in the edge and vehicles was not affected until only 7 vehicles were available (only enough for the primary demand). Hence, the full backup is placed in edge nodes increasing the total power consumption by 8% compared to the 15 vehicles case. The demand was not served with 6 vehicles as it exceeds the capacity

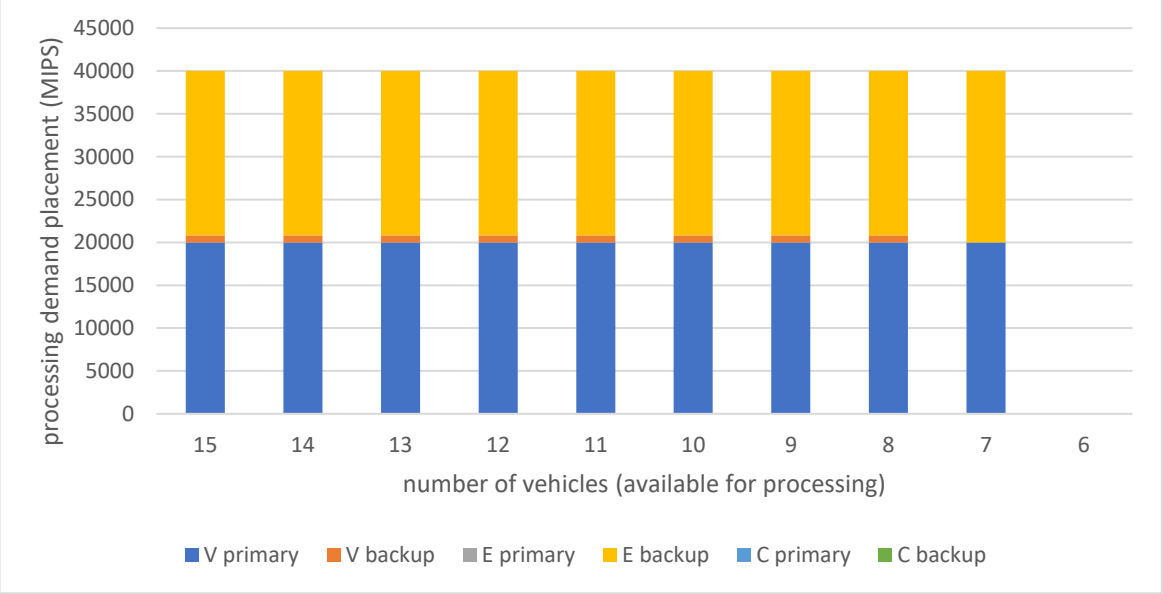

Figure 6.13: Processing demand placement using PV_BA with APR and changing vehicles number

For the PV_BEC, Figure 6.14 shows an increase in networking power consumption, where the total power consumption increased by 3%. The backup placement was more stable and was not affected by the vehicles leaving as it only uses edge and cloud.

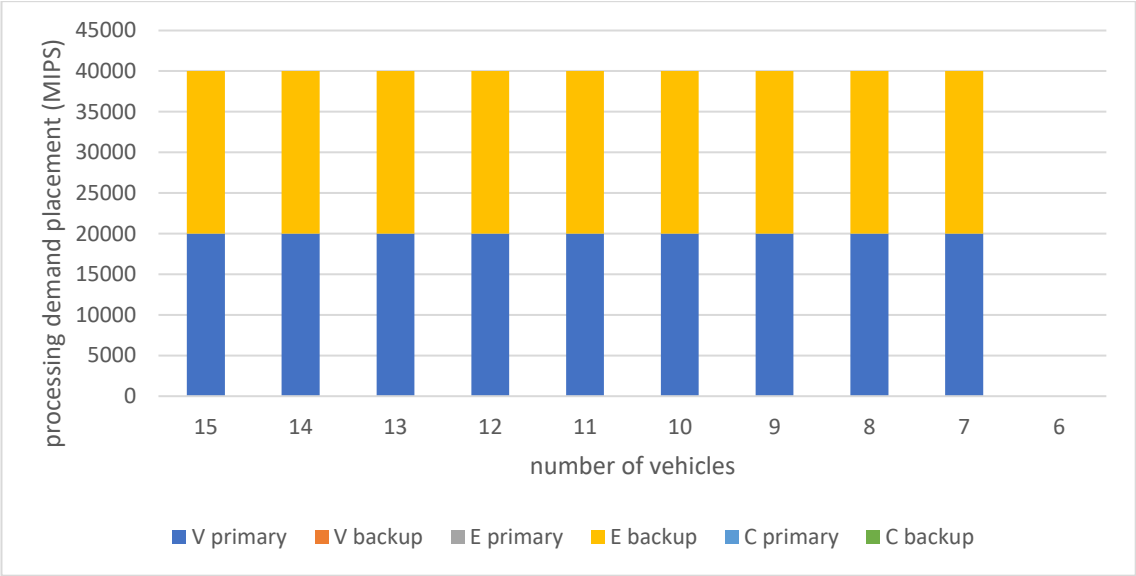

Figure 6.14: Processing demand placement using PV_BEC with APR and changing vehicles number

For the PA_BA scenario, most of the demand is primarily processed in the edge unaffected by vehicles leaving the car park, as shown in Figure 6.15. A small portion of the demand is primarily placed in a vehicle and is backed up in a vehicle as well. As vehicles leave, the networking power consumption increases (very slightly) while maintaining the same placement decision in two vehicles for primary and backup. When only one vehicle is available to do processing, the model placed the full demand in the edge node. The decision avoided the need for backup (which would have needed the activation of an extra edge node), but the power consumption was higher by 4% than before.

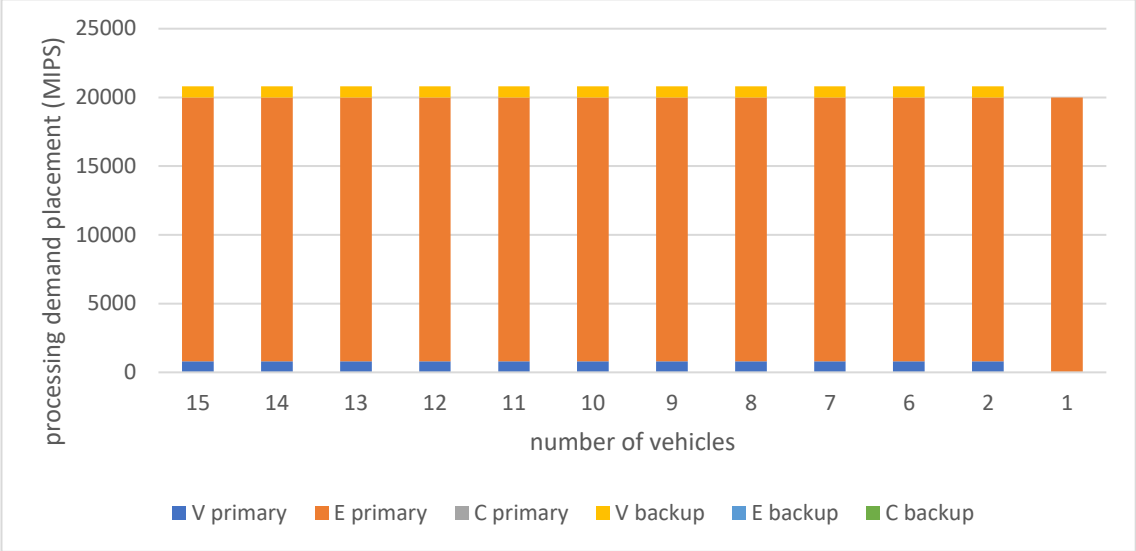

Figure 6.15: Processing demand placement using PV_BA with APR and changing vehicles number

6.4.2 Multiple Demands

In this section we discuss the processing allocation and power consumption increase resulting from serving multiple demands considering active and inactive resilience schemes.

We consider up to 5 demands of three requirement levels: low (traffic 1 Mbps, Processing 2000 MIPS), medium (traffic 3 Mbps, Processing 6000 MIPS), and high (traffic 5 Mbps, Processing 10000 MIPS). We limited the number of demands to 5 demands to ensure that networking and processing resources are sufficient to serve primary and backup processing for all demand volumes, including the scenarios where the primary allocation is in vehicles only (5 demands of high volumes require 50000 MIPS, and the total vehicles processing capacity is 51200 MIPS). In this section we discuss the processing allocation and power consumption increase resulting from serving multiple demands considering active and idle resilience schemes.

6.4.2.1 Active Processing Resilience (APR)

Figure 6.16 shows the preference to use the edge nodes as primary for low demand size. It also shows the use of the vehicles as backup nodes for the (PV_BA) scenario which can be explained by the fact that the vehicles which have a demand are already active so they can be used as back up, albeit a poor choice for reliability. For the medium and high demands, Figure 6.17 and Figure 6.18, when the primary node can be in any level, the placement of the demand is similar to the no resilience scenario, in edge and cloud. However, when the primary processing is in the vehicles, edge and then cloud are the better choices for backup for medium and high demands, respectively.

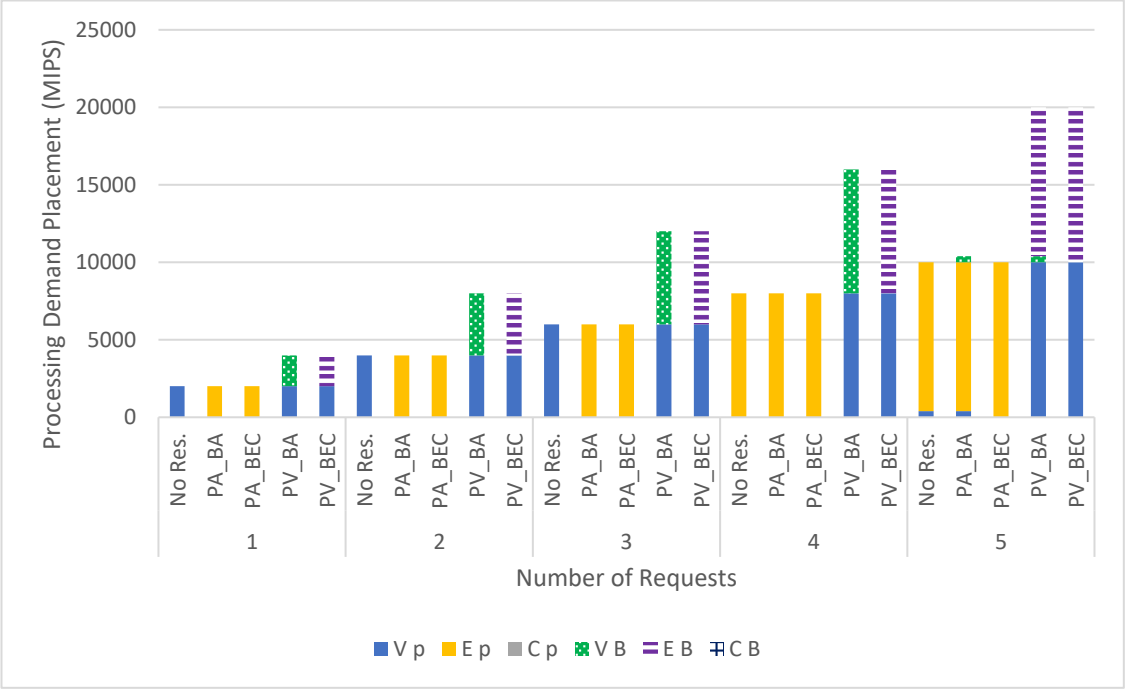

Figure 6.16: Processing demand placement of APR scenarios considering multiple demands of low requirements

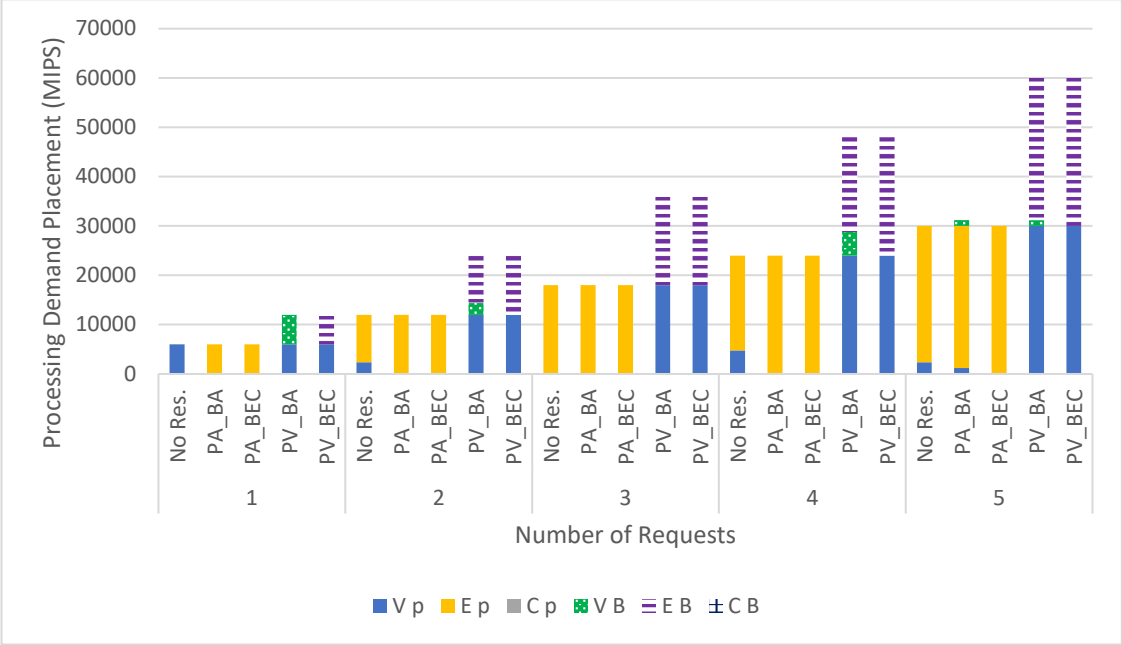

Figure 6.17: Processing demand placement of APR scenarios considering multiple demands of medium requirements

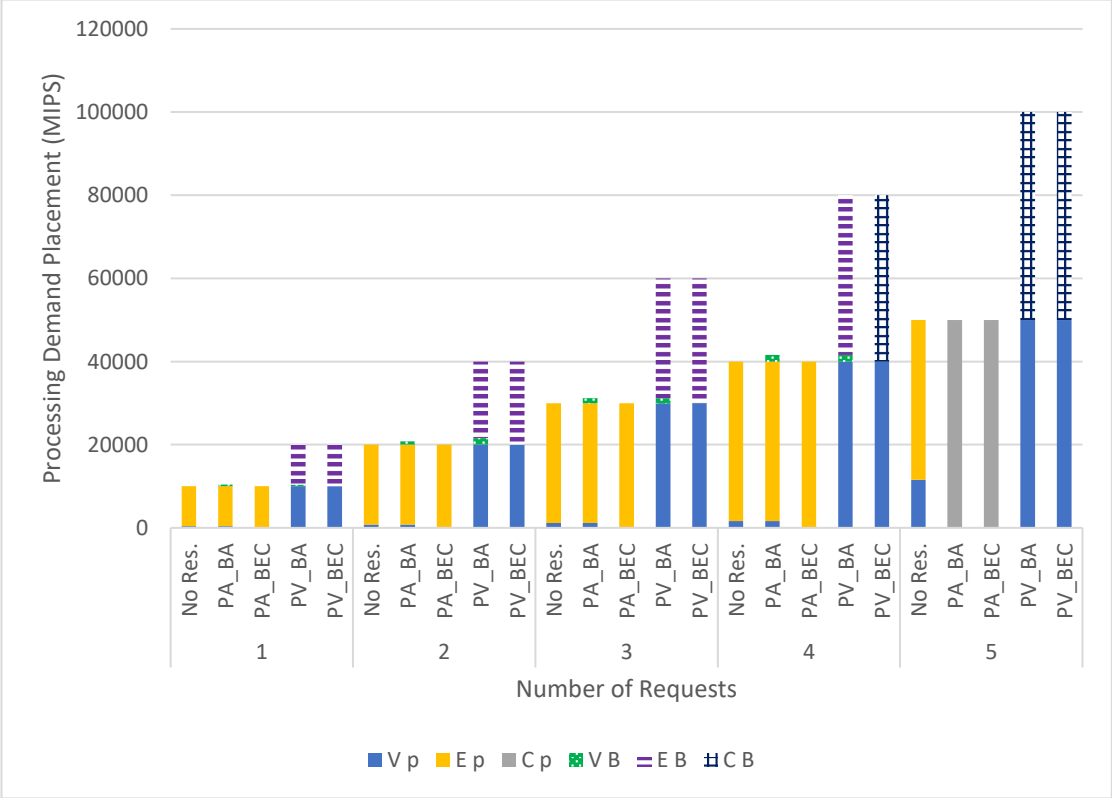

Figure 6.18: Processing demand placement of APR scenarios considering multiple demands of high requirements

Figure 6.19 shows that under the PA_BA and PA_BEC scenarios up to 4 demands of low volume can be served in the edge nodes to avoid the need for backup. For 5 demands, the PA_BA scenario shows that processing in the vehicles, which are backed up in other vehicles is more energy efficient than processing in the edge nodes as done in PA_BEC, where the limited capacity of the edge forces the activation of an extra node. The edge nodes capacity is not enough to serve 5 demands which are optimally processed in a combination of edge nodes and vehicles. Note that processing in vehicles which is backed up in other vehicles is more energy efficient than processing in the cloud. For the PV_BA scenario, only vehicles were used to serve up to 4 demands as they are already activated to serve primary processing of one demand and can accommodate the backup processing for another demand. Note that the primary and backup processing cannot be allocated to the same node for the same demand. To serve 5 demands, one or more edge nodes are needed to accommodate some of the backup. For the PV_BEC, the edge nodes were chosen to act as backup nodes. Similar behaviour can be seen for medium volume demands in Figure 6.20, except that for PV_BA scenario, where the need for edge nodes for backup begins at lower number of demands due to their bigger size. Figure 6.21 shows a tendency to use the cloud as primary or backup destination as the demands grow in number and size (4 and 5 demands).

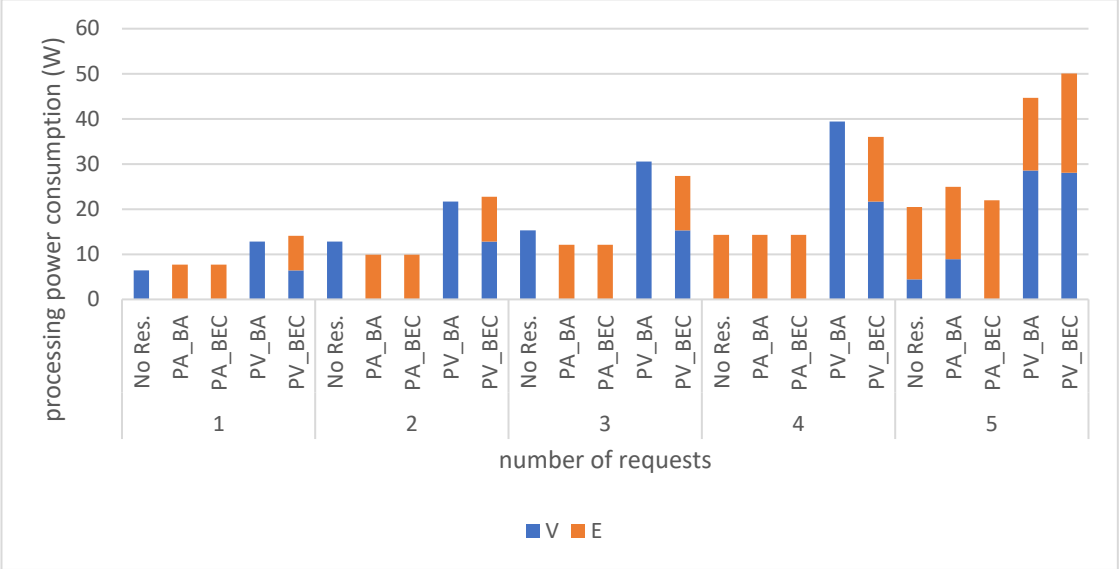

Figure 6.19: Processing power consumption of APR scenarios considering multiple demands of low requirements

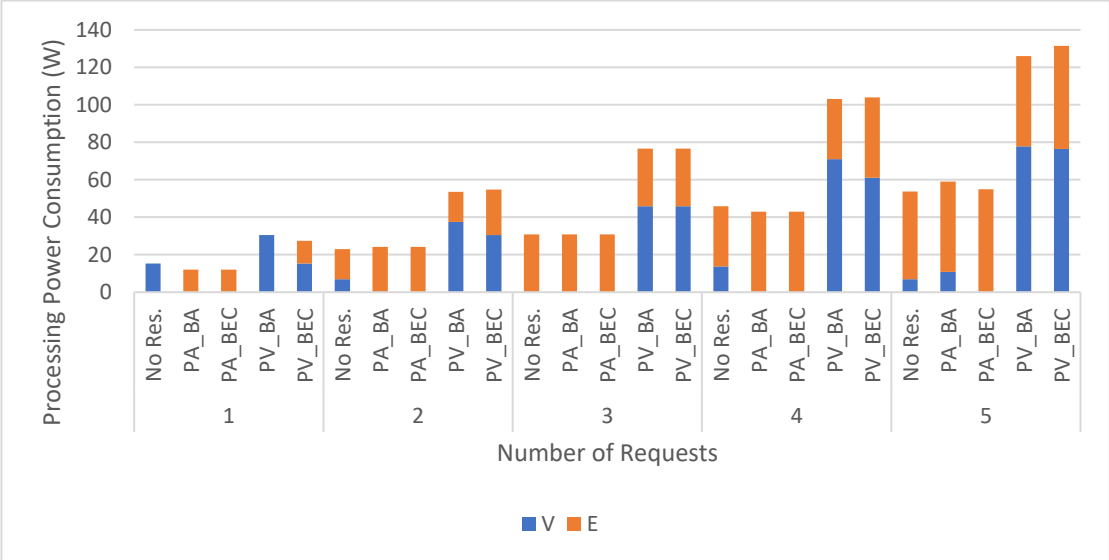

Figure 6.20: Processing power consumption of APR scenarios considering multiple demands of medium requirements

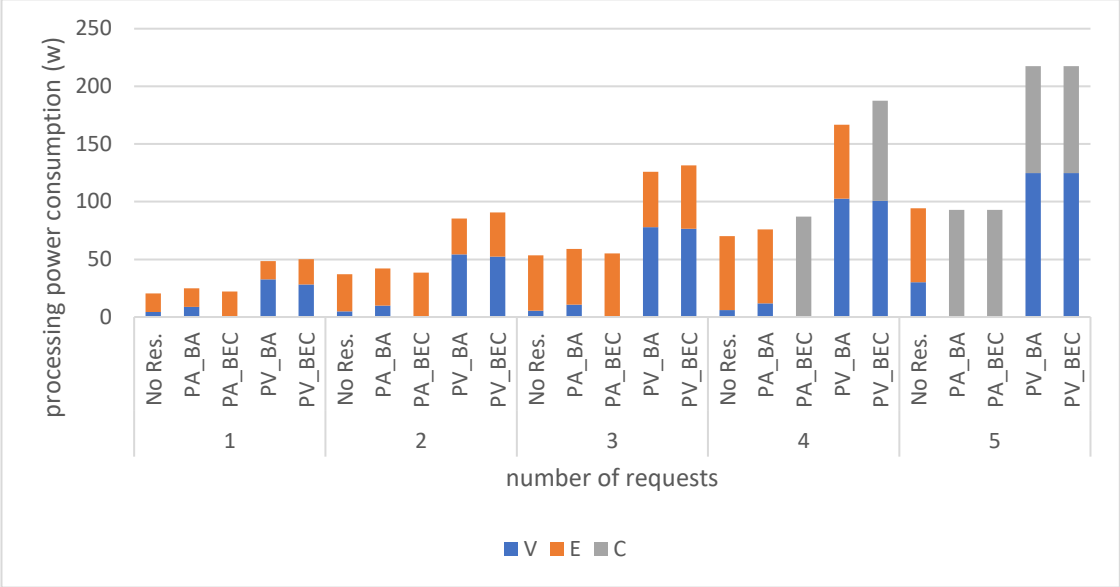

Figure 6.21: Processing power consumption of APR scenarios considering multiple demands of high requirements

Figure 6.22- Figure 6.24 show the increase in power consumption of the active resilience scheme scenarios in comparison with the No-Res scenario for the low, medium, and high demands. The power consumption increase varies with the additional nodes activated to serve as backup.

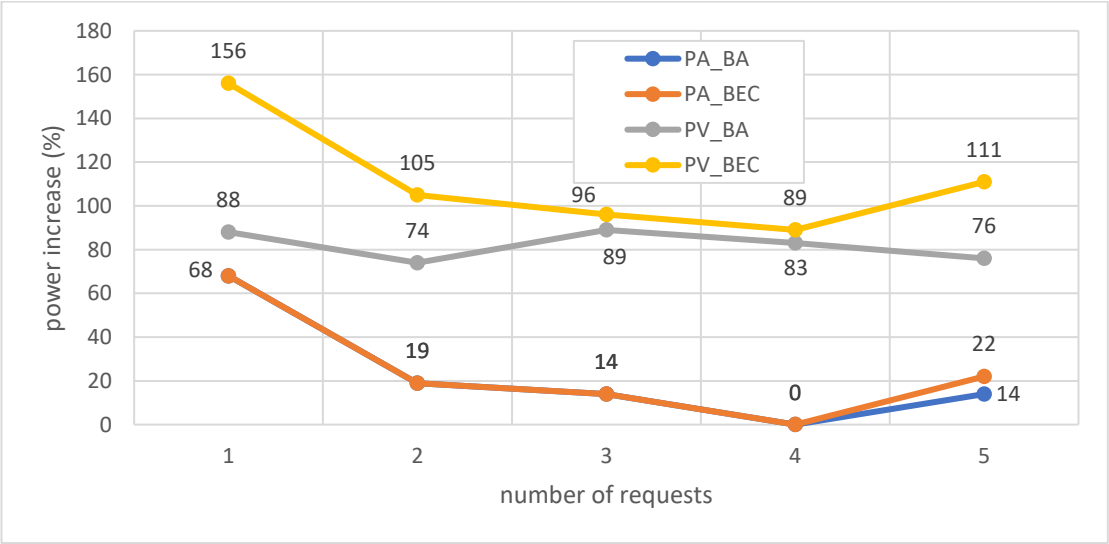

Figure 6.22: Increase in power consumption of APR scenarios considering multiple demands of low requirements

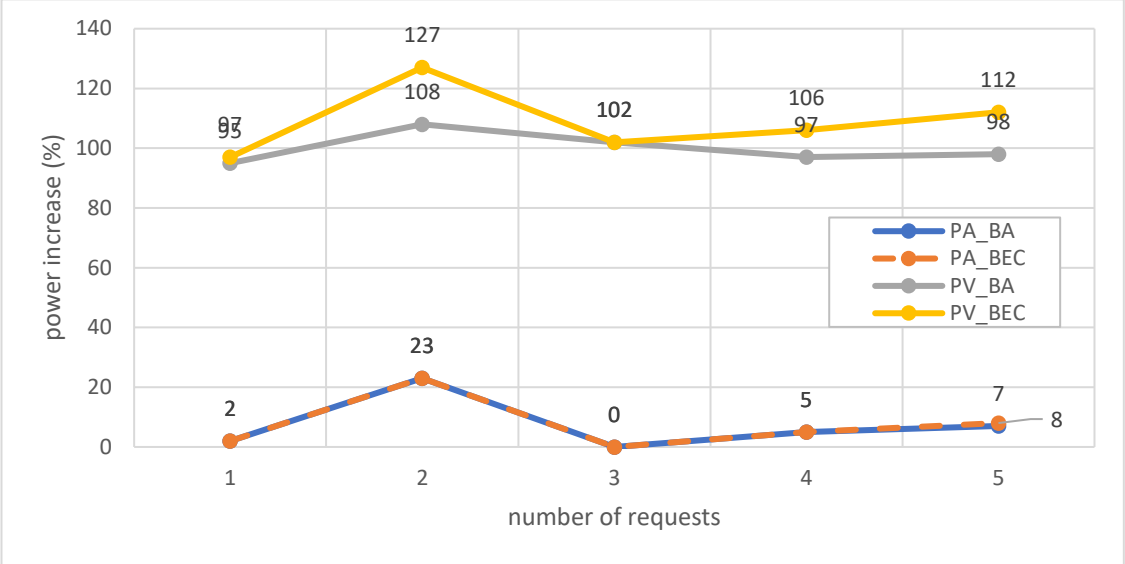

Figure 6.23: Increase in power consumption of APR scenarios considering multiple demands of medium requirements

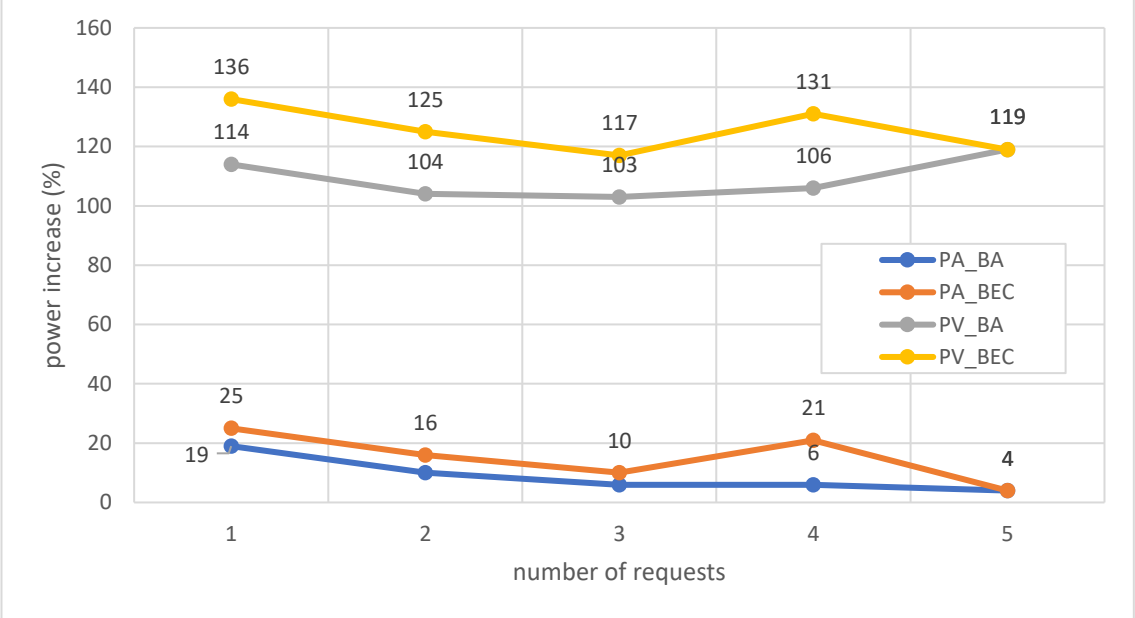

Figure 6.24: Increase in Processing power consumption of APR scenarios considering multiple demands of high requirements

6.4.2.2 Idle Processing Resilience (IPR)

Figure 6.25 - Figure 6.27 show no change in the behaviour of the model when allocating processing demands under the IPR scheme. However, significant reduction in the processing power consumption can be detected as observed in Figure 6.28 - Figure 6.30.

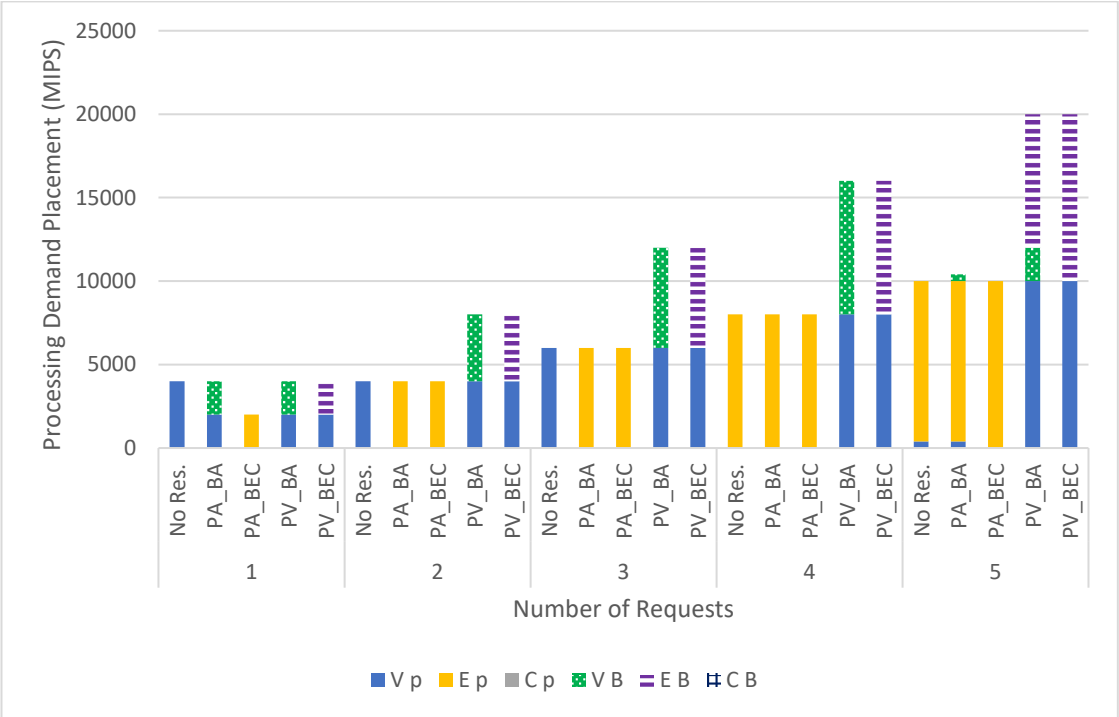

Figure 6.25: Processing demand placement of IPR scenarios considering multiple demands of low requirements

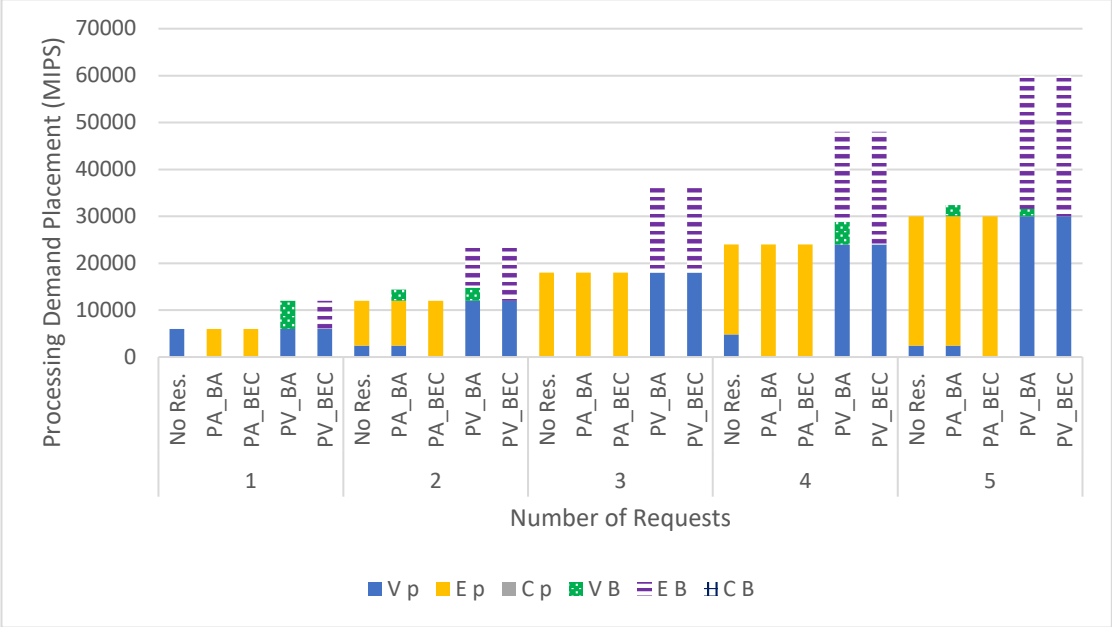

Figure 6.26: Processing demand placement of IPR scenarios considering multiple demands of medium requirements

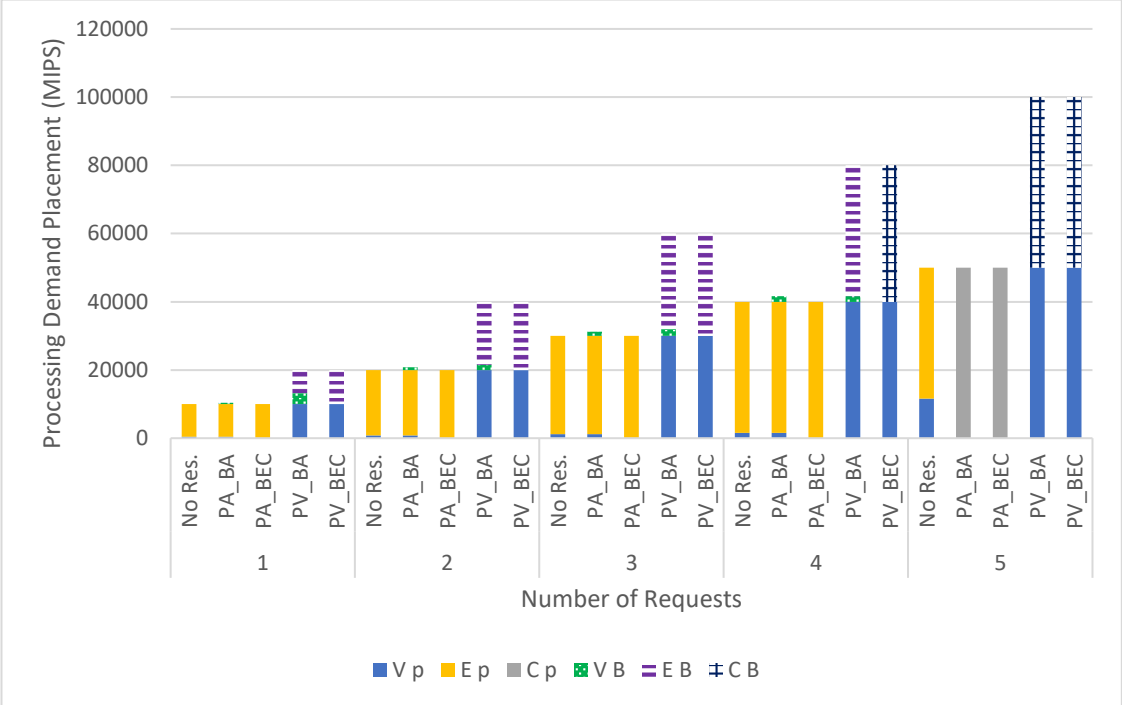

Figure 6.27: Processing demand placement of IPR scenarios considering multiple demands of high requirements

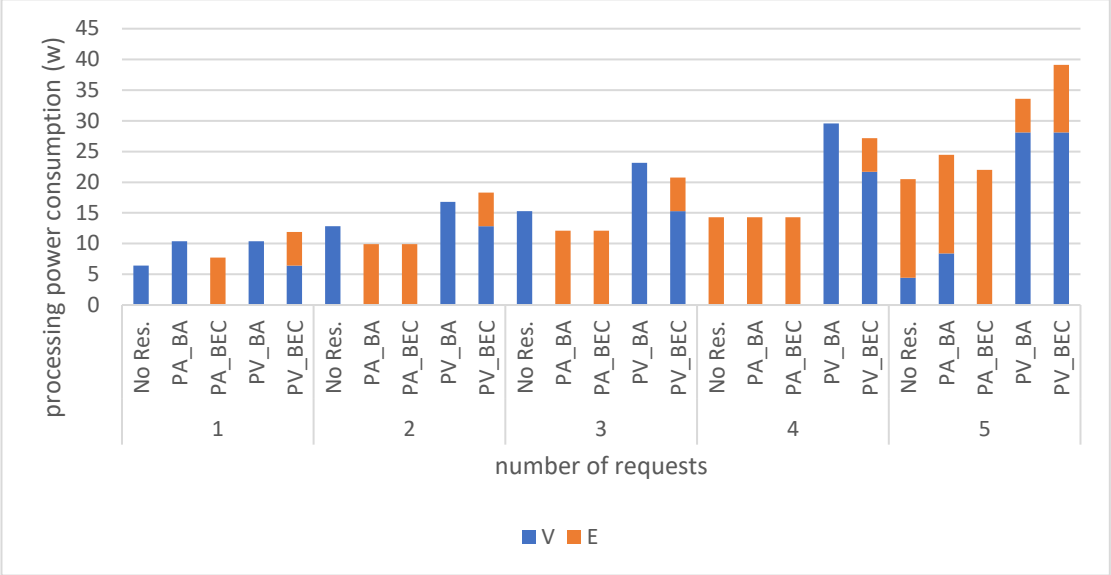

Figure 6.28: Processing power consumption of IPR scenarios considering multiple demands of low requirements

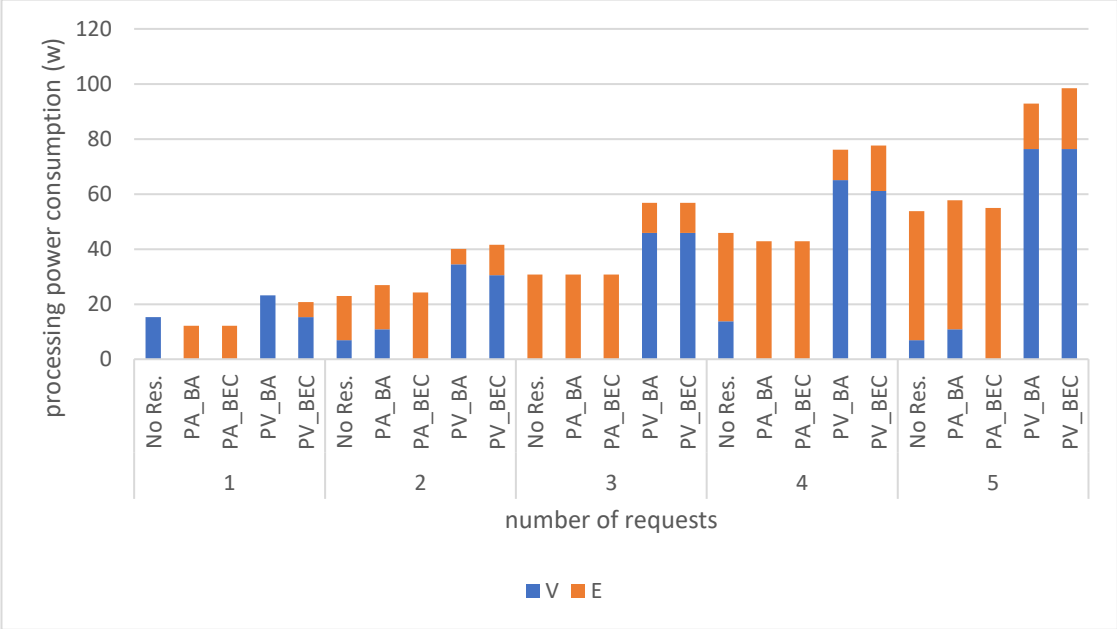

Figure 6.29: Processing power consumption of IPR scenarios considering multiple demands of medium requirements

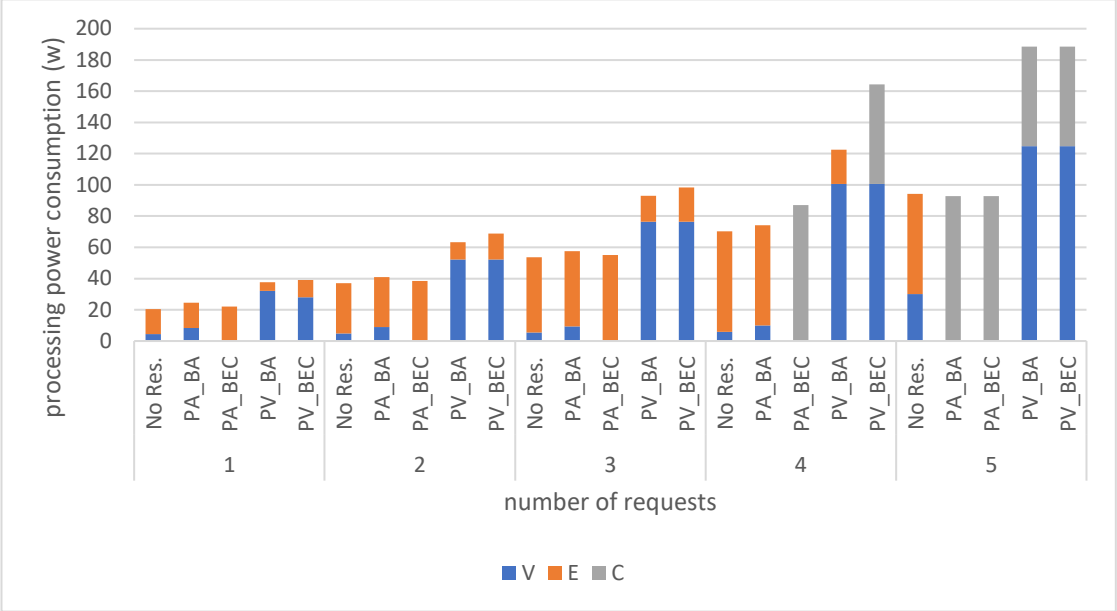

Figure 6.30: Processing power consumption of IPR scenarios considering multiple demands of high requirements

Figure 6.31 - Figure 6.33 show the increase in power consumption of the idle resilience scheme scenarios in comparison with the No-Resilience scenario for the low, medium, and high demands. For the PV-BEC scenario the increase in power consumption is reduced by an average of 32%, 40% and 34% for low, medium and high demands, respectively compared to the power consumption increase resulting from the active resilience.

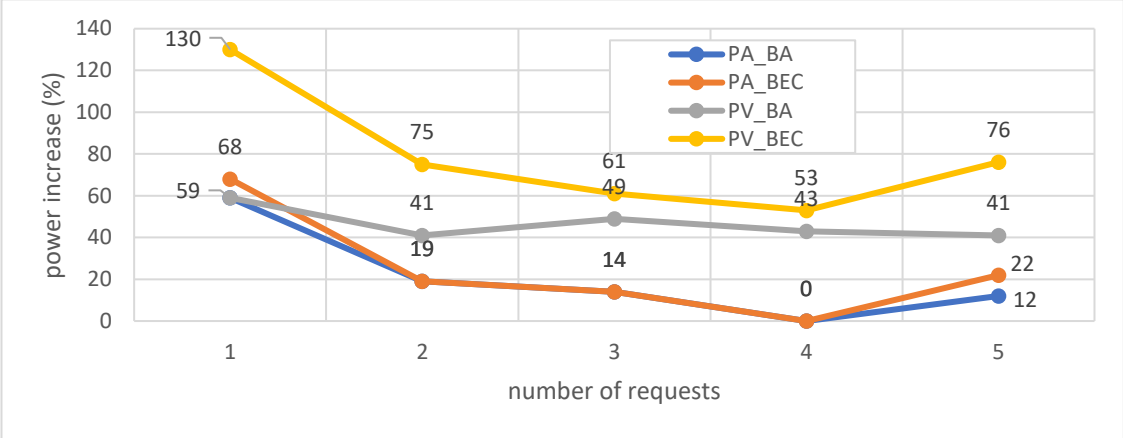

Figure 6.31: Increase in power consumption of IPR scenarios considering multiple demands of low requirements

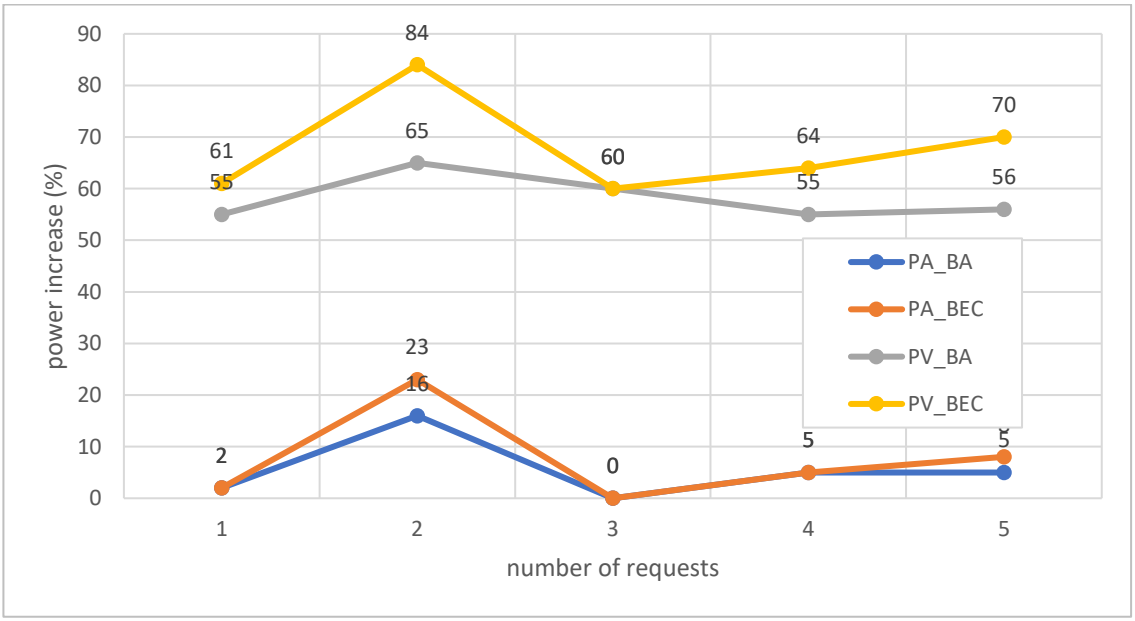

Figure 6.32: Increase in power consumption of IPR scenarios considering multiple demands of medium requirements

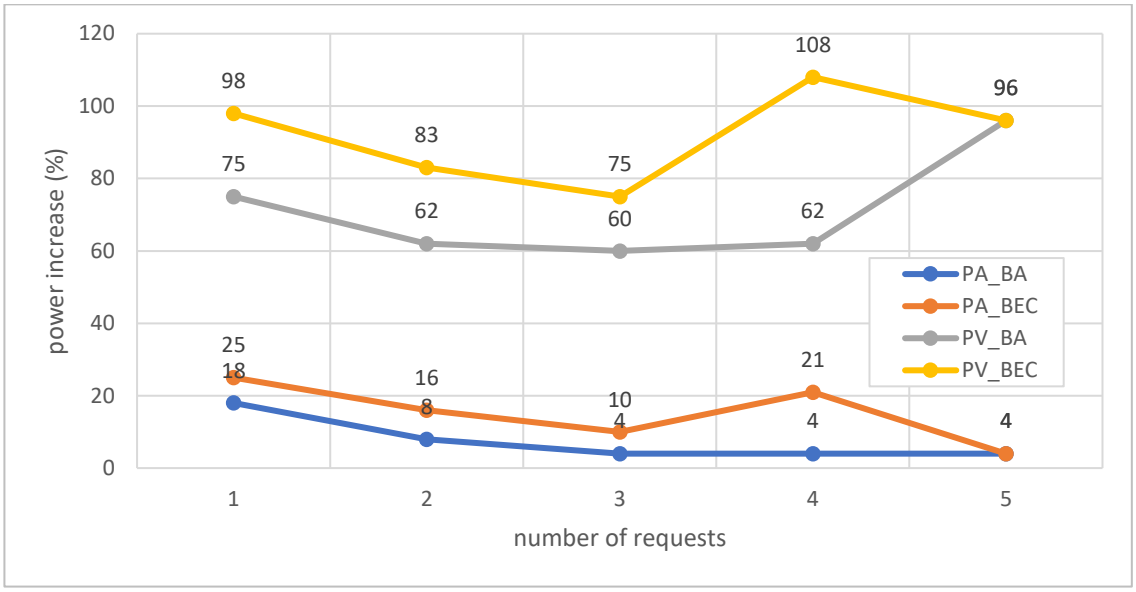

Figure 6.33: Increase in power consumption of IPR scenarios considering multiple demands of high requirements

6.4.3 MILP Results Verifications

To verify the results of the resilient model, we conduct a validation using three checkpoints of known optimal solutions. As seen in Table 6-3 - Table 6-5. We produce a check point for a PA_BA case with active resilience, PV_PA with active resilience, and PV_PA with idle resilience

Table 6-3: Analytic verification checkpoint # 1: Power minimisation using PA_BA scenario with APR

Case: Allocating a demand of 2 Mb and 4000 MI **Optimal Solution: Allocate the demand to Edge Node as primary node (to avoid backup)**				
Power Calculation				
Node	Vehicle	Edge	OLT + Metro + Core	Cloud Server
Processing Power (W)		(4000 * 0.0011) + 2 = 6.5		
Networking Power (W)	(2 * 0.0901667) + 1.05 = 1.41	5.5		
Total Power (W)	13.4			

MILP Power (W)	16

Table 6-4: Analytic verification checkpoint # 2: Power minimisation using PV_BA scenario with APR

Case: Allocating a demand of 2 Mb and 4000 MI **Optimal Solution: Allocate to 4 Vehicles for primary and backup**				
Power Calculation				
Node	Vehicle	Edge	OLT + Metro + Core	Cloud Server
Processing Power (W)	(8000 * 0.00123) + (4 * 3.95) = 25.64			
Networking Power (W)	(4 * 2 * 0.00625) + (5 * 1.05) = 5.3			
Total Power (W)	**30.94**			
MILP Power (W)	**30.94**			

Table 6-5: Analytic verification checkpoint # 3: Power minimisation using PV_BA scenario with IPR

Case: Allocating a demand of 2 Mb and 4000 MI **Optimal Solution: Allocate to 4 Vehicles for primary and backup**				
Power Calculation				
Node	Vehicle	Edge	OLT + Metro + Core	Cloud Server
Processing Power (W)	(4000 * 0.00123) + (4 * 3.95) = 20.72			
Networking Power (W)	(4 * 2 * 0.00625) + (5 * 1.05) = 5.3			
Total Power (W)	26.02			
MILP Power (W)	26.02			

6.5 Summary

In this chapter, we extend the model of Chapter 4 to include resilience measures and studied the impact of these measures on the power consumption. The resilience measures are needed when the processing demand is allocated to vehicular nodes. We investigated two resilience schemes, APR where both primary and backup nodes consume processing power, and IPR where only idle processing power is consumed by the backup nodes. In each scheme, four scenarios are implemented based on the allowed allocation layer of primary and backup processing. The results showed that whenever possible, the model tried to avoid using the vehicular nodes as primary processors to avoid the need for back up. Also, a combination of vehicular nodes and edge nodes is the optimal choice of location for primary and back up processing, except for very high demand sizes. In this case the model choses to perform primary processing in the cloud. For the APR, introducing resilience resulted in significant increase in the power consumption compared to the result of Chapter 4 where no resilience measures are introduced. The increase can be as high as 149% when only vehicular nodes are allowed as primary nodes. Having more flexibility in the allocation of primary demand in the three layers causes the increase in the power consumption to be reduced to a more tolerable value of 23%. Using the IPR showed an average reduction of 34% in the power consumption increase compared to the APR.

Chapter 7
Conclusion and Future Research Directions

7.1 Summary of Current Work

This thesis addressed the problem of distributed processing allocation in the context of a vehicular environment. An end-to-end vehicular cloud architecture was developed, in which vehicles are clustered in vehicular clouds, willing to sharing their computational resources. The vehicular clouds layer is also supported with edge nodes layers and the central cloud layer, to provide extra choices for process allocation. The architecture was evaluated by developing a MILP model to optimally allocate user demands over the three processing layers. The three layers VC architecture has become well-known in the literature. Much research emphasised on the energy efficiency or reduced latency as a result of the use of VC in comparison to cloud. However, to the best of our knowledge, no one had quantitatively evaluated those benefits, and most efforts were dedicated to improvements in the routing algorithms and resource allocation strategies at the VC (and edge) levels only. To that end, our work provided numerical values for the energy and delay when using the distributed VC and edge nodes against the use of the conventional cloud as baseline.

The metrics for evaluation focused on the minimisation of energy and delay. The contributions of the work can be summarised as follows:

In Chapter 3, a vehicular cloud architecture was proposed, introducing the processing layers, communication interfaces, and control and coordination mode. The Mixed integer linear programming has been introduced as the major evaluation tool, and the main input parameters and impacting factors were explained.

In Chapter 4, a MILP model was developed to optimally allocate processing demands received from end users, with an objective to minimise the total power consumption. a range of test cases were considered to test the performance of the model, such as varying the demand size, limiting the processing demand splitting, splitting the traffic demand in accordance with processing, and serving multiple demands. The results show that the vehicles and edge layer can achieve power savings as high as 84% for lower demands volumes, the processing efficiency gradually decreases as the demand size increases. Furthermore, splitting a processing demand improves the energy efficiency of processing in the vehicles and edge nodes by 71%. The results show that applications which require proportional traffic splitting, among the processing destinations serving the demand, can be more efficiently processed by vehicles and edge nodes. This increases the average power savings to 3%-16% compared to cloud processing for higher demand volumes. The results showed that the allocation of processing demand was affect by limitations of the vehicular communications data rate due to the need to replicate traffic demand to each of the processing destinations. In the chapter, a heuristic for real-time allocation of demands has been developed and produced comparable results to the MILP model.

In Chapter 5, the MILP model was modified to optimally allocate processing resources to on-demand smart city applications while minimising the energy consumption and end-to-end-delay. When minimising energy consumption, the cloud was optimally selected, with 7% increase in end-to-end delay compared to the delay minimisation scenario. For delay minimisation, processing is optimally distributed over the three processing levels which increased the processing energy consumption. Joint optimisation of energy consumption and delay maintained the energy efficiency of the energy minimisation scenario while limiting the increase in end-to-end delay to 7% compared to the delay minimisation scenario.

In Chapter 6, the MILP model was modified to introduce resilience measures into the architecture. Whenever a vehicle is chosen to process a demand, a backup node received the same demand for safekeeping for later, when the primary vehicle is no longer available. The model had two resilience mechanisms, active processing resilience and idle processing resilience. Both mechanisms were tested with different scenarios to optimally choose the primary and backup nodes with minimised power consumption objective. The results showed that whenever possible, the model tried to avoid using the vehicular nodes as primary processors to avoid the need for back up. Compared to the non-resilient model, the increase can be as high as 149% when only vehicular nodes are allowed as primary nodes. The increase can be at only 23% when flexibility in the allocation of primary demand in the three layers is granted.

7.2 Future Directions

The work presented in the thesis can be extended in different directions. Following are some of the potential future research that aim at having a comprehensive and complete study of the end-to-end vehicular network architecture.

7.2.1 Control and Coordination

In the description of the architecture, we stated that a vehicular cloud falls under the control of an edge node, which has the knowledge needed in terms of the available resources to make demand allocation decisions. The time and power consumed in collecting this information, as well as, in exchanging any messages between a demand source and the corresponding edge node or any destination assigned to serve it are ignored in the current work. However, in a distributed and dynamic environment such as vehicular networks, these control and coordination messages contribute to service delay and power consumption and can be substantial. To complete the evaluation of the architecture performance, researching this topic is important in the future.

7.2.2 Dynamicity and mobility

All the test cases in the thesis were carried out in a parking lot, deploying a static vehicular cloud. However, one of the characterising challenges of vehicular networks is the mobility of vehicles and the dynamic ad hoc topology. As we already implement a distance-dependent networking power consumption, we plan on taking it further and have distance change over time. We would inspect how this will impact the power, delay, and the level of vehicular resources

utilisation, especially considering the high velocity expected in vehicles. There are also different types of transportation network layout that should be considered in inner roads or highways, urban or rural.

7.2.3 Security and privacy

One of the main concerns of users of remote resources is the assurance of their security and privacy. In vehicular clouds, this issue is double faced. First, a user would need to guarantee that security measures are available in distributed resources such as vehicles and edge nodes, and that these measures are as efficient as the ones found in conventional clouds operated by well-known technology companies. Also, they would need to ensure that the privacy of their personal data is not jeopardised when sent to vehicles owned by other people. On the other hand, the owners of the vehicles clustering in the vehicular clouds would have similar concerns. They need to be confident that their own data that might be running or stored in their vehicles will not be exposed. Also, they need to have protection against hackers and cyber-attacks, and to know that the request for services is a legitimate one. Research in this topic is still very lacking in the context of vehicular clouds, and it is hoped to make significant contribution in this area in our future work.

7.2.4 Virtual machine migration

One of the important issues to be addressed in the future is the vehicular virtual machine migration. The topic of virtual machine migration has been thoroughly investigated for central cloud. However, the same cannot be said for vehicular clouds. Also, the algorithms and techniques developed for central clouds cannot

readily be applied to vehicular clouds as the paradigm and the reasons for migration are very different. While load balancing and improved utilisation are the main triggers for migration in conventional networks, mobility and short availability of resources are the main reasons in the vehicular network. It is one of the challenging issues that we aim to address in the future to provide migration algorithms and different evaluation metrics that can be applicable to our architecture and other vehicular networks architecture.

7.2.5 Introduction of SDN

When the architecture was proposed in Chapter 4, it was mentioned that vehicular clouds are under the control of edge nodes and that edge nodes would have knowledge of the surrounding environment and the changes in the availability of resources and other updates on the topology. The explanation implicitly implied the concept of software-defined networking, which is popular for handling ad hoc and dynamic architecture. A lot of work is already taking place to leverage SDN in vehicular networking. In the future, we would want to explicitly introduce software-defined networking into the architecture. We would try to view the architecture in the SDN light, as control plane and data plane, and see the changes to be made, the techniques available to SDN and applicable to VN, and the issues that are solvable.

7.2.6 Incentives

As vehicles are the main building blocks of vehicular clouds, measures should be taken to encourage the owners of the vehicles to share their resources. Aside from ensuring security and privacy as mentioned above, the vehicle owners

would expect to gain from allowing their devices to participate in the architecture. The incentives can take many forms. The most obvious one might be monetary gain, provided by service providers who take over the control of the vehicles during the time they are used. Another way is to have some extra benefits, e.g. extended online service subscription, granted to the vehicle owner in exchange for use of resources. The study of the topic is crucial to judge the feasibility of the architecture.

7.2.7 Power Sources

Modern vehicles' OBU and the many sensors are mainly powered by the car battery. Even though the lifetime of battery is usually long, adding the extra work of sharing computational resources and servicing different types of applications would surely affect this time. It is an important issue, which is mostly neglected up to this point, to investigate how vehicular clouds can drain the vehicle power supplies. The issue has many scenarios to consider, such as the installation of separate battery, adding limitations on the allowed use of power supplies, and the use of other types of modern vehicles like electric and hybrid ones. The work can also be extended to include another metric to measure the effect on the environment and the emission of CO2 to the air and how to green vehicular clouds.

7.2.8 Propagation Delay Domination

Proximity to end users is one of the main attractive features of VC. One of the main issues with the conventional cloud is the long distance, which in turn increases propagation delay. However, with present technologies, other metrics

such as processing efficiency and data rates influence the choice of processing allocation and they can dominate over the propagation delay induced latency. If the processing capacity available at a node is low, then the processing latency can dominate. Similarly, if the capacity of routers and switches is comparable to the input traffic volume, then queueing latency can dominate. It is our ambition to expand our work to find what future processing and communication capabilities can be bestowed on vehicles so that propagation delay is the dominating factor in the decision of processing destination. In other words, when allocating a resource, the nearest processing resource would be chosen as the processing and bandwidth capacities are comparable in all processing layers.

References

[1] S. Kamboj and N. S. Ghumman, "A survey on cloud computing and its types," pp. 2971-2974: Bharati Vidyapeeth, New Delhi as the Organizer of INDIACom - 2016.

[2] J. Elmirghani *et al.*, "GreenTouch GreenMeter core network energy-efficiency improvement measures and optimization," vol. 10, no. 2, pp. A250-A269, 2018.

[3] A. M. Al-Salim, A. Q. Lawey, T. E. El-Gorashi, J. M. J. I. T. o. N. Elmirghani, and S. Management, "Energy efficient big data networks: impact of volume and variety," vol. 15, no. 1, pp. 458-474, 2018.

[4] S. Abdelhamid, H. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," *IEEE Network,* vol. 29, no. 1, pp. 12-17, 2015.

[5] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, "Rethinking Vehicular Communications: Merging VANET with cloud computing," pp. 606-609: IEEE.

[6] E. Lee, E. K. Lee, M. Gerla, and S. Y. Oh, "Vehicular cloud networking: architecture and design principles," *IEEE Communications Magazine,* vol. 52, no. 2, pp. 148-155, 2014.

[7] L. Gu, D. Zeng, and S. Guo, "Vehicular cloud computing: A survey," pp. 403-407: IEEE.

[8] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 6, pp. 3860-3873, 2016.

[9] C. Liang and F. R. Yu, "Wireless Network Virtualization: A Survey, Some Research Issues and Challenges," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 1, pp. 358-380, 2015.

[10] F. Malandrino, C.-F. Chiasserini, and C. Casetti, "Virtualization-based evaluation of backhaul performance in vehicular applications," *Computer Networks,* vol. 134, pp. 93-104, 2018.

[11] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE,* vol. 99, no. 7, pp. 1162-1182, 2011.

[12] E.-K. Lee, M. Gerla, G. Pau, U. Lee, and J.-H. Lim, "Internet of Vehicles: From intelligent grid to autonomous cars and vehicular fogs," *International Journal of Distributed Sensor Networks,* vol. 12, no. 9, p. 1550147716665500, 2016.

[13] R. Hussain, Z. Rezaeifar, and H. Oh, "A Paradigm Shift from Vehicular Ad Hoc Networks to VANET-Based Clouds," *Wireless Personal Communications,* vol. 83, no. 2, pp. 1131-1158, 2015.

[14] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications,* vol. 37, pp. 380-392, 2014/01/01/ 2014.

[15] S. Grafling, P. Mahonen, and J. Riihijarvi, "Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications," pp. 344-348.
[16] A. Fitah, A. Badri, M. Moughit, and A. Sahel, "Performance of DSRC and WIFI for Intelligent Transport Systems in VANET," *Procedia Computer Science,* vol. 127, pp. 360-368, 2018/01/01/ 2018.
[17] I. Yaqoob, I. Ahmad, E. Ahmed, A. Gani, M. Imran, and N. Guizani, "Overcoming the Key Challenges to Establishing Vehicular Communication: Is SDN the Answer?," *IEEE Communications Magazine,* vol. 55, no. 7, pp. 128-134, 2017.
[18] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 12, pp. 9457-9470, 2016.
[19] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5G: The next generation of mobile communication," *Physical Communication,* vol. 18, pp. 64-84, 2016.
[20] C.-F. Lai, Y.-C. Chang, H.-C. Chao, M. S. Hossain, and A. Ghoneim, "A Buffer-Aware QoS Streaming Approach for SDN-Enabled 5G Vehicular Networks," *IEEE Communications Magazine,* vol. 55, no. 8, pp. 68-73, 2017.
[21] E. Larsson *et al.*, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine,* vol. 52, no. 2, pp. 186-195, 2014.
[22] X. Ge, J. Ye, Y. Yang, and Q. Li, "User Mobility Evaluation for 5G Small Cell Networks Based on Individual Mobility Model," *IEEE Journal on Selected Areas in Communications,* vol. 34, no. 3, pp. 528-541, 2016.
[23] L. Wang, T. Han, Q. Li, J. Yan, X. Liu, and D. Deng, "Cell-Less Communications in 5G Vehicular Networks Based on Vehicle-Installed Access Points," *IEEE Wireless Communications,* vol. 24, no. 6, pp. 64-71, 2017.
[24] B. M. Masini, A. Bazzi, and E. Natalizio, "Radio Access for Future 5G Vehicular Networks," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1-7.
[25] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. P. C. Rodrigues, and A. Vinel, "Data Offloading in 5G-Enabled Software-Defined Vehicular Networks: A Stackelberg-Game-Based Approach," *IEEE Communications Magazine,* vol. 55, no. 8, pp. 100-108, 2017.
[26] C. Lai, H. Zhou, N. Cheng, and X. S. Shen, "Secure Group Communications in Vehicular Networks: A Software-Defined Network-Enabled Architecture and Solution," *IEEE Vehicular Technology Magazine,* vol. 12, no. 4, pp. 40-49, 2017.
[27] M. H. Eiza, Q. Ni, and Q. Shi, "Secure and Privacy-Aware Cloud-Assisted Video Reporting Service in 5G-Enabled Vehicular Networks," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 10, pp. 7868-7881, 2016.

[28] J. Qiao, Y. He, and X. S. Shen, "Improving Video Streaming Quality in 5G Enabled Vehicular Networks," *IEEE Wireless Communications,* vol. 25, no. 2, pp. 133-139, 2018.
[29] H. Li, M. Dong, and K. Ota, "Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 10, pp. 7895-7904, 2016.
[30] S. Olariu, M. Eltoweissy, and M. Younis, "Towards autonomous vehicular clouds," *ICST Transactions on Mobile Communications and Applications,* vol. 11, no. 7–9, pp. e2-11, 2011.
[31] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications,* vol. 40, pp. 325-344, 2014.
[32] M. Gerla, "Vehicular Cloud Computing," in *2012 The 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2012, pp. 152-155.
[33] T. Kim, H. Min, J. Park, J. Lee, and J. Jung, "Analysis on characteristics of vehicle and parking lot as a datacenter," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, 2017, pp. 1-4.
[34] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, and I. Khalil, "Datacenter at the Airport: Reasoning about Time-Dependent Parking Lot Occupancy," *IEEE Transactions on Parallel and Distributed Systems,* vol. 23, no. 11, pp. 2067-2080, 2012.
[35] Huawei, "5G Unlocks a World of Oppertunities: Top Ten 5G Use Cases, ONLINE : https://www-file.huawei.com/-/media/CORPORATE/PDF/mbb/5g-unlocks-a-world-of-opportunities-v5.pdf?la=en&source=corp_comm," White Paper 2017.
[36] Ofinno, "Traffic Control and Vehicle-to-Everything (V2X) Communications, ONLINE: https://ofinno.com/wp-content/uploads/2018/03/Ofinno_V2X_WP.New_.pdf," White Paper 2018.
[37] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, and S. Cherkaoui, "RSU cloud and its resource management in support of enhanced vehicular applications," in *2014 IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 127-132.
[38] S. Vodopivec, J. Bešter, and A. Kos, "A survey on clustering algorithms for vehicular ad-hoc networks," in *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*, 2012, pp. 52-56.
[39] M. Abuelela and S. Olariu, "Taking VANET to the clouds," pp. 6-13: ACM.
[40] F. Malandrino, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Content Download in Vehicular Networks in Presence of Noisy Mobility Prediction," *IEEE Transactions on Mobile Computing,* vol. 13, no. 5, pp. 1007-1021, 2014.
[41] S. Igder, S. Bhattacharya, B. R. Qazi, and J. M. H. Elmirghani, "Standalone Green Cache Points for Vehicular Content Distribution Networks," in *2016 10th International Conference on Next Generation Mobile Applications, Security and Technologies (NGMAST)*, 2016, pp. 59-64.

[42] S. Igder, H. Idjmayyel, B. R. Qazi, S. Bhattacharya, and J. M. H. Elmirghani, "Load Adaptive Caching Points for a Content Distribution Network," in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, 2015, pp. 150-155.
[43] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: Stopped cars are not silent," pp. 431-435.
[44] H. Idjmayyel, W. Kumar, B. R. Qazi, and J. M. H. Elmirghani, "Energy and QoS - A new perspective in a city vehicular communication network," pp. 327-331.
[45] H. Idjmayyel, B. R. Qazi, and J. M. H. Elmirghani, "Energy Efficient Double Cluster Head Routing Scheme in a City Vehicular Network," pp. 1594-1599: IEEE.
[46] Z. Zhou, P. Liu, Z. Chang, C. Xu, and Y. Zhang, "Energy-efficient workload offloading and power control in vehicular edge computing," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2018, pp. 191-196.
[47] L. Gu, D. Zeng, S. Guo, and B. Ye, "Leverage parking cars in a two-tier data center," pp. 4665-4670: IEEE.
[48] S. Midya, A. Roy, K. Majumder, S. J. J. o. N. Phadikar, and C. Applications, "Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach," vol. 103, pp. 58-84, 2018.
[49] Y. Zhu, X. Liu, M. Li, and Q. Zhang, "POVA: Traffic Light Sensing with Probe Vehicles," *IEEE Transactions on Parallel and Distributed Systems,* vol. 24, no. 7, pp. 1390-1400, 2013.
[50] C. E. Palazzi, F. Pezzoni, and P. M. Ruiz, "Delay-bounded data gathering in urban vehicular sensor networks," *Pervasive and Mobile Computing,* vol. 8, no. 2, pp. 180-193, 2012.
[51] W. Q. Wang, X. Zhang, J. Zhang, and H. B. Lim, "Smart traffic cloud: An infrastructure for traffic applications," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, 2012, pp. 822-827: IEEE.
[52] S. Gupte and M. Younis, "Vehicular networking for intelligent and autonomous traffic management," in *2012 IEEE international conference on communications (ICC)*, 2012, pp. 5306-5310: IEEE.
[53] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla, "On the effectiveness of an opportunistic traffic management system for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems,* vol. 12, no. 4, pp. 1537-1548, 2011.
[54] R. Copeland *et al.*, "Technology assessment for mission-critical services on automotive virtual edge communicator (AVEC)," in *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2018, pp. 1-8.
[55] Z. Xiao, P. Li, V. Havyarimana, G. M. Hassana, D. Wang, and K. Li, "GOI: A Novel Design for Vehicle Positioning and Trajectory Prediction Under Urban Environments," *IEEE Sensors Journal,* vol. 18, no. 13, pp. 5586-5594, 2018.

[56] P. Doshi, D. Kapur, R. Iyer, and A. Chatterjee, "Smart mobility: Algorithm for road and driver type determination," in *2017 IEEE Transportation Electrification Conference (ITEC-India)*, 2017, pp. 1-4.
[57] S. Kumar, S. Gollakota, and D. Katabi, "A cloud-assisted design for autonomous driving," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 41-46.
[58] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "Carspeak: a content-centric network for autonomous driving," *ACM SIGCOMM Computer Communication Review,* vol. 42, no. 4, pp. 259-270, 2012.
[59] C.-M. Huang, M.-S. Chiang, D.-T. Dao, H.-M. Pai, S. Xu, and H. Zhou, "Vehicle-to-Infrastructure (V2I) offloading from cellular network to 802.11p Wi-Fi network based on the Software-Defined Network (SDN) architecture," *Vehicular Communications,* vol. 9, pp. 288-300, 2017/07/01/ 2017.
[60] A. M. Mustafa, O. M. Abubakr, O. Ahmadien, A. Ahmedin, and B. Mokhtar, "Mobility prediction for efficient resources management in vehicular cloud computing," in *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, 2017, pp. 53-59: IEEE.
[61] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Communications Surveys & Tutorials,* vol. 11, no. 4, pp. 19-41, 2009.
[62] M. Kim, I. Jang, S. Choo, and S. Pack, "On security in Software-defined vehicular cloud," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, 2016, pp. 1259-1260.
[63] M. Kim, I. Jang, S. Choo, J. Koo, and S. Pack, "Collaborative security attack detection in software-defined vehicular networks," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2017, pp. 19-24.
[64] L. Gafencu and L. Scripcariu, "Vehicular cloud: Overview and security issues," in *2018 International Conference on Development and Application Systems (DAS)*, 2018, pp. 78-82.
[65] H. Goumidi, Z. Aliouat, and S. Harous, "Vehicular Cloud Computing Security: A Survey," *Arabian Journal for Science and Engineering,* vol. 45, no. 4, pp. 2473-2499, 2020/04/01 2020.
[66] B. Baron *et al.*, "Virtualizing vehicular node resources: Feasibility study of virtual machine migration," *Vehicular Communications,* vol. 4, pp. 39-46, 2016/04/01/ 2016.
[67] R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai, and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," *Journal of Network and Computer Applications,* vol. 52, pp. 11-25, 2015/06/01/ 2015.
[68] T. K. Refaat, B. Kantarci, and H. T. Mouftah, "Virtual machine migration and management for vehicular clouds," *Vehicular Communications,* vol. 4, pp. 47-56, 2016.

[69] H. Yao *et al.*, "Migrate or not? Exploring virtual machine migration in roadside cloudlet-based vehicular cloud," vol. 27, no. 18, pp. 5780-5792, 2015.
[70] R. Yu, Y. Zhang, H. Wu, P. Chatzimisios, and S. Xie, "Virtual machine live migration for pervasive services in cloud-assisted vehicular networks," in *2013 8th International Conference on Communications and Networking in China (CHINACOM)*, 2013, pp. 540-545.
[71] A. Wang, M. Iyer, R. Dutta, G. N. Rouskas, and I. Baldine, "Network Virtualization: Technologies, Perspectives, and Frontiers," *Journal of Lightwave Technology,* vol. 31, no. 4, pp. 523-537, 2013.
[72] B. Yi, X. Wang, K. Li, S. k. Das, and M. Huang, "A comprehensive survey of Network Function Virtualization," *Computer Networks,* vol. 133, pp. 212-262, 2018.
[73] ETSI:. Available: http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_NFV-EVE005v010101p.pdf
[74] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," *IEEE Communications Magazine,* vol. 55, no. 5, pp. 94-100, 2017.
[75] X. Li *et al.*, "Network Slicing for 5G: Challenges and Opportunities," *IEEE Internet Computing,* vol. 21, no. 5, pp. 20-27, 2017.
[76] A. Bhatia, K. Haribabu, K. Gupta, and A. Sahu, "Realization of flexible and scalable VANETs through SDN and virtualization," pp. 280-282: IEEE.
[77] H. Li, K. Ota, and M. Dong, "Network Virtualization Optimization in Software Defined Vehicular Ad-Hoc Networks," pp. 1-5: IEEE.
[78] (January 2020). *Open Network Foundation O.N.F*. Available: https://www.opennetworking.org/
[79] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 1, pp. 27-51, 2015.
[80] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE,* vol. 103, no. 1, pp. 14-76, 2015.
[81] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys & Tutorials,* vol. 16, no. 3, pp. 1617-1634, 2014.
[82] H. Farhady, H. Lee, and A. Nakao, "Software-Defined Networking: A survey," *COMPUTER NETWORKS,* vol. 81, pp. 79-95, 2015.
[83] A. Lara, A. Kolasani, and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *IEEE Communications Surveys & Tutorials,* vol. 16, no. 1, pp. 493-512, 2014.
[84] D. Das, J. Bapat, and D. Das, "A Dynamic QoS Negotiation Mechanism Between Wired and Wireless SDN Domains," *IEEE Transactions on Network and Service Management,* vol. 14, no. 4, pp. 1076-1085, 2017.

[85] A. K. Rangisetti and B. R. Tamma, "Software Defined Wireless Networks: A Survey of Issues and Solutions," *Wireless Personal Communications,* journal article vol. 97, no. 4, pp. 6019-6053, December 01 2017.
[86] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software Defined Optical Networks (SDONs): A Comprehensive Survey," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 4, pp. 2738-2786, 2016.
[87] A. Aguado *et al.*, "Secure NFV Orchestration Over an SDN-Controlled Optical Network With Time-Shared Quantum Key Distribution Resources," *Journal of Lightwave Technology,* vol. 35, no. 8, pp. 1357-1362, 2017.
[88] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility Support for Fog Computing: An SDN Approach," *IEEE Communications Magazine,* vol. 56, no. 5, pp. 53-59, 2018.
[89] M. Chahal, S. Harit, K. K. Mishra, A. K. Sangaiah, and Z. G. Zheng, "A Survey on software-defined networking in vehicular ad hoc networks: Challenges, applications and use cases," *SUSTAINABLE CITIES AND SOCIETY,* vol. 35, pp. 830-840, 2017.
[90] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, "Towards software-defined VANET: Architecture and services," in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2014, pp. 103-110.
[91] Y. Ni, J. He, and L. Cai, "Data Dissemination in Software-Defined Vehicular Networks," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1-5.
[92] D.-J. Deng, S.-Y. Lien, C.-C. Lin, S.-C. Hung, and W.-B. Chen, "Latency Control in Software-Defined Mobile-Edge Vehicular Networking," *IEEE Communications Magazine,* vol. 55, no. 8, pp. 87-93, 2017.
[93] J. Rufino, M. Alam, J. Almeida, and J. Ferreira, "Software defined P2P architecture for reliable vehicular communications," *Pervasive and Mobile Computing,* vol. 42, pp. 411-425, 2017/12/01/ 2017.
[94] S. Correia, A. Boukerche, and R. I. Meneguette, "An Architecture for Hierarchical Software-Defined Vehicular Networks," *IEEE Communications Magazine,* vol. 55, no. 7, pp. 80-86, 2017.
[95] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular Adhoc Network with Fog Computing," pp. 1202-1207: IEEE.
[96] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing," *IEEE Communications Magazine,* vol. 55, no. 7, pp. 94-100, 2017.
[97] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for Vehicular Communications," *IEEE Communications Magazine,* vol. 56, no. 1, pp. 111-117, 2018.
[98] X. Ge, Z. Li, and S. Li, "5G Software Defined Vehicular Networks," *IEEE Communications Magazine,* vol. 55, no. 7, pp. 87-93, 2017.

[99] Z. He, D. Zhang, and J. Liang, "Cost-Efficient Sensory Data Transmission in Heterogeneous Software-Defined Vehicular Networks," *IEEE Sensors Journal,* vol. 16, no. 20, pp. 7342-7354, 2016.
[100] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. M. Leung, "SOVCAN: Safety-Oriented Vehicular Controller Area Network," *IEEE Communications Magazine,* vol. 55, no. 8, pp. 94-99, 2017.
[101] Z. Jiao, H. Ding, M. Dang, R. Tian, and B. Zhang, "Predictive Big Data Collection in Vehicular Networks: A Software Defined Networking Based Approach," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1-6.
[102] A. Hammadi and L. Mhamdi, "A survey on architectures and energy efficiency in Data Center Networks," *Computer Communications,* vol. 40, no. 0, pp. 1 - 21, 2014.
[103] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A Survey on Data Center Networking (DCN): Infrastructure and Operations," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 1, pp. 640-656, 2017.
[104] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," presented at the Proceedings of the ACM SIGCOMM 2008 conference on Data communication, Seattle, WA, USA, 2008.
[105] A. Greenberg *et al.*, "VL2: a scalable and flexible data center network," *SIGCOMM Comput. Commun. Rev.,* vol. 39, no. 4, pp. 51-62, 2009.
[106] "QFabric Architecture: Implementing a Flat Data Center Network," Juniper Networks.
[107] C. Guo *et al.*, "BCube: a high performance, server-centric network architecture for modular data centers," *SIGCOMM Comput. Commun. Rev.,* vol. 39, no. 4, pp. 63-74, 2009.
[108] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu, "FiConn: Using Backup Port for Server Interconnection in Data Centers," pp. 2276-2285.
[109] N. Farrington *et al.*, "Helios: a hybrid electrical/optical switch architecture for modular data centers," *SIGCOMM Comput. Commun. Rev.,* vol. 40, no. 4, pp. 339-350, 2010.
[110] Y.-H. K. Kang Xia, Ming Yangb and H. Jonathan Chaoa, "Petabit Optical Switch for Data Center Networks," Polytechnic Institute of New York University, Brooklyn, New York2010, Available: http://eeweb.poly.edu/chao/publications/petasw.pdf.
[111] Cisco. (December 2019). *Cisco Visual Networking Index Forecast 2017-2022*. Available: https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/
[112] C. Prakash and S. Dasgupta, "Cloud computing security analysis: Challenges and possible solutions," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 54-57.
[113] Z. Tari, X. Yi, U. S. Premarathne, P. Bertok, and I. Khalil, "Security and Privacy in Cloud Computing: Vision, Trends, and Challenges," *IEEE Cloud Computing,* vol. 2, no. 2, pp. 30-38, 2015.

[114] Z. Xiao and Y. Xiao, "Security and Privacy in Cloud Computing," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 2, pp. 843-859, 2013.
[115] B.R. (July 2020). *Report to Congress on Server and Data Center Energy Efficiency: Public Law 109-431. 2007.* Available: http://www.datacenterdynamics.com/epa-report-to-congress-on-server-and-data-center-energy-efficiency/30051.fullarticle
[116] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Distributed energy efficient clouds over core networks," 2014.
[117] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Green IP over WDM networks with data centers," *Journal of Lightwave Technology,* vol. 29, no. 12, pp. 1861-1880, 2011.
[118] A. M. Al-Salim, T. E. El-Gorashi, A. Q. Lawey, and J. M. J. I. O. Elmirghani, "Greening big data networks: Velocity impact," vol. 12, no. 3, pp. 126-135, 2017.
[119] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew, "Greening geographical load balancing," *IEEE/ACM Transactions on Networking (TON),* vol. 23, no. 2, pp. 657-671, 2015.
[120] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy Efficient Virtual Network Embedding for Cloud Networks," 2014.
[121] A. Beloglazov and R. Buyya, "Energy Efficient Resource Management in Virtualized Cloud Data Centers," pp. 826-831.
[122] E. Gurrola, S. Melendez, M. P. McGarry, P. J. Teller, D. Doria, and D. Bruno, "The effect of network delay estimation error on computation offloading decisions," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, 2015, pp. 325-327.
[123] H. Shoja, H. Nahid, and R. Azizi, "A comparative survey on load balancing algorithms in cloud computing," in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2014, pp. 1-5.
[124] M. F. Bari *et al.*, "Data Center Network Virtualization: A Survey," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 2, pp. 909-928, 2013.
[125] G. Ghoda, N. Meruliya, D. H. Parekh, T. Gajjar, D. Dave, and R. Sridaran, "A survey on data center network virtualization," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 3464-3470.
[126] T. Welsh and E. Benkhelifa, "On Resilience in Cloud Computing: A Survey of Techniques across the Cloud Domain," *ACM Comput. Surv.,* vol. 53, no. 3, p. Article 59, 2020.
[127] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 3, pp. 2244-2281, 2016.
[128] V. Prokhorenko and M. A. Babar, "Architectural Resilience in Cloud, Fog and Edge Systems: A Survey," *IEEE Access,* vol. 8, pp. 28078-28095, 2020.
[129] D. Sun, G. Chang, C. Miao, and X. Wang, "Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing

environments," *The Journal of Supercomputing,* vol. 66, no. 1, pp. 193-228, 2013/10/01 2013.
[130] A. Yousefpour *et al.*, "All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey," 2018.
[131] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal,* vol. 5, no. 1, pp. 450-465, 2018.
[132] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog Computing May Help to Save Energy in Cloud Computing," *IEEE Journal on Selected Areas in Communications,* vol. 34, no. 5, pp. 1728-1739, 2016.
[133] B. Yosuf, M. Musa, T. Elgorashi, A. Q. Lawey, and J. M. H. Elmirghani, "Energy Efficient Service Distribution in Internet of Things," in *2018 20th International Conference on Transparent Optical Networks (ICTON)*, 2018, pp. 1-4.
[134] H. Q. Al-Shammari, A. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Service embedding in IoT networks," *IEEE Access,* 2019.
[135] S. Igder, S. Bhattacharya, and J. M. Elmirghani, "Energy efficient fog servers for Internet of Things information piece delivery (IoTIPD) in a smart city vehicular environment," in *2016 10th International Conference on Next Generation Mobile Applications, Security and Technologies (NGMAST)*, 2016, pp. 99-104: IEEE.
[136] L. Zhao, J. Wang, J. Liu, and N. Kato, "Optimal edge resource allocation in IoT-based smart cities," *IEEE Network,* vol. 33, no. 2, pp. 30-35, 2019.
[137] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in *2017 IEEE international conference on edge computing (EDGE)*, 2017, pp. 17-24: IEEE.
[138] C. Zhu, G. Pastor, Y. Xiao, Y. Li, and A. Ylae-Jaeaeski, "Fog following me: Latency and quality balanced task allocation in vehicular fog computing," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018, pp. 1-9: IEEE.
[139] Y. Chen, E. Sun, and Y. Zhang, "Joint optimization of transmission and processing delay in fog computing access networks," in *2017 9th International Conference on Advanced Infocomm Technology (ICAIT)*, 2017, pp. 155-158.
[140] D. Xu *et al.*, "A survey of opportunistic offloading," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 2198-2236, 2018.
[141] C. S. M. Babou *et al.*, "Hierarchical Load Balancing and Clustering Technique for Home Edge Computing," *IEEE Access,* vol. 8, pp. 127593-127607, 2020.
[142] R. Beraldi, A. Mtibaa, and H. Alnuweiri, "Cooperative load balancing scheme for edge computing resources," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 94-100.
[143] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 2, pp. 2092-2104, 2020.

[144] Y. Kim, N. An, J. Park, and H. Lim, "Mobility Support for Vehicular Cloud Radio-Access-Networks with Edge Computing," in *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, 2018, pp. 1-4.
[145] M. Bousselham, N. Benamar, and A. Addaim, "A new Security Mechanism for Vehicular Cloud Computing Using Fog Computing System," in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1-4.
[146] I. Griva, S. Nash, and A. Sofer, *Linear and nonlinear optimization*, 2nd ed. (no. Book, Whole). Philadelphia: Society for Industrial and Applied Mathematics, 2009.
[147] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: a modeling language for mathematical programming*, 2nd ed. (no. Book, Whole). Pacific Grove, CA: Thomson/Brooks/Cole, 2003.
[148] E. Oki, *Linear programming and algorithms for communication networks: a practical guide to network design, control, and management* (no. Book, Whole). Boca Raton: CRC Press, 2013.
[149] Z. C. T. J. Cole Smith, "A Tutorial Guide to Mixed-Integer Linear Programming Models and Solution Techniques ", ed, 2007.
[150] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 2012.
[151] S. Olariu, "A Survey of Vehicular Cloud Research: Trends, Applications and Challenges," *IEEE Transactions on Intelligent Transportation Systems,* vol. 21, no. 6, pp. 2648-2663, 2020.
[152] A. Boukerche and E. Robson, "Vehicular cloud computing: Architectures, applications, and mobility," *Computer networks,* vol. 135, pp. 171-189, 2018.
[153] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, "IoT-Cloud Service Optimization in Next Generation Smart Environments," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS,* vol. 34, no. 12, pp. 4077-4090, 2016.
[154] C. Delgado, J. R. Gállego, M. Canales, J. Ortín, S. Bousnina, and M. Cesana, "On optimal resource allocation in virtual sensor networks," *Ad Hoc Networks,* vol. 50, pp. 23-40, 2016/11/01/ 2016.
[155] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: eliminating server idle power," *SIGARCH Comput. Archit. News,* vol. 37, no. 1, pp. 205–216, 2009.
[156] A. B. Reis and S. Sargento, "Statistics of parked cars for urban vehicular networks," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016, pp. 1-6.
[157] D. o. t. Environment:. *UK parking Standards* Available: https://www.infrastructure-ni.gov.uk/publications/parking-standards
[158] S. Gräfling, P. Mähönen, and J. Riihijärvi, "Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications," in *2010 Second International Conference on Ubiquitous and Future Networks (ICUFN)*, 2010, pp. 344-348.

[159] Cisco. (2019, January 2019). *Cisco Industrial Benchmark*. Available: https://www.cisco.com/c/dam/global/da_dk/assets/docs/presentations/vBootcamp_Performance_Benchmark.pdf
[160] SAVARI. (2018, August 2018). *MobiWAVE On Board Unit*. Available: http://savari.net/wp-content/uploads/2017/05/MW-1000_April2017.pdf
[161] AradaSystems:. (August 2018). *LocoMate mini 2 On Board Unit.* . Available: http://www.aradasystems.com/wp-content/uploads/2014/12/LocoMate_mini2_Datasheet_v2.01.pdf
[162] Espressif. (December 2019). *ESP8266EX Datasheet*. Available: https://www.espressif.com/en/support/download/documents
[163] RuckusWorks. (2017, October 2017). *T710 Series Access Points*. Available: http://www.ruckusworks.co.uk/datasheets/ds-zoneflex-t710-series.pdf
[164] J. Huang, Y. Meng, X. Gong, Y. Liu, and Q. Duan, "A Novel Deployment Scheme for Green Internet of Things," *IEEE Internet of Things Journal,* vol. 1, no. 2, pp. 196-205, 2014.
[165] G. Bekaroo and A. Santokhee, "Power consumption of the Raspberry Pi: A comparative analysis," in *Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech), IEEE International Conference on*, 2016, pp. 361-366: IEEE.
[166] R. Pi. (Auguest 2018). *Raspberry Pi 3 Model B*. Available: https://www.raspberrypi.org/documentation/faqs/
[167] F. G. ONU. (March 2020). *FTE7502 10G ONU*. Available: https://www.sumitomoelectric.com/wp-content/uploads/2016/01/OAN-7502-12212017-1.pdf
[168] Intel. (March 2020). *Intel® Xeon® Processor E5-2680 v2*. Available: https://ark.intel.com/content/www/us/en/ark/products/75277/intel-xeon-processor-e5-2680-v2-25m-cache-2-80-ghz.html
[169] Cisco. (December 2019). *Cisco Nexus 9300-EX Series Switches Data Sheet*. Available: https://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/datasheet-c78-742283.html
[170] Cisco. (December 2019). *Cisco Network Convergence System 5500 Series: Fixed Chassis Data Sheet*. Available: https://www.cisco.com/c/en/us/products/collateral/routers/network-convergence-system-5500-series/datasheet-c78-737935.html
[171] Cisco. (October 2019). *Cisco ME 4600 Series Optical Line Terminal Data Sheet*. Available: https://www.cisco.com/c/en/us/products/collateral/switches/me-4600-series-multiservice-optical-access-platform/datasheet-c78-730445.html
[172] A. N. Al-Quzweeni, A. Q. Lawey, T. E. Elgorashi, and J. M. J. I. A. Elmirghani, "Optimized Energy Aware 5G Network Function Virtualization," vol. 7, pp. 44939-44958, 2019.
[173] A. Shehabi *et al.*, "United states data center energy usage report," Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States)2016, Available: https://www.osti.gov/servlets/purl/1372902/.

[174] K. L. K. Sudheera, M. Ma, G. G. M. N. Ali, and P. H. J. Chong, "Delay efficient software defined networking based architecture for vehicular networks," in *2016 IEEE International Conference on Communication Systems (ICCS)*, 2016, pp. 1-6.
[175] A. Kawabata, B. C. Chatterjee, S. Ba, and E. Oki, "A Real-Time Delay-Sensitive Communication Approach Based on Distributed Processing," *IEEE Access,* vol. 5, pp. 20235-20248, 2017.
[176] C. Zhu *et al.*, "Folo: Latency and quality optimized task allocation in vehicular fog computing," 2018.
[177] Z. Ahmed, S. Naz, and J. Ahmed, "Minimizing transmission delays in vehicular ad hoc networks by optimized placement of road-side unit," *Wireless Networks,* vol. 26, no. 4, pp. 2905-2914, 2020.
[178] P. Ghazizadeh, S. Olariu, A. G. Zadeh, and S. El-Tawab, "Towards fault-tolerant job assignment in vehicular cloud," in *2015 IEEE International Conference on Services Computing*, 2015, pp. 17-24: IEEE.
[179] T. Adhikary *et al.*, "Quality of service aware reliable task scheduling in vehicular cloud computing," vol. 21, no. 3, pp. 482-493, 2016.
[180] C. Huang and K. Xu, "Reliable realtime streaming in vehicular cloud-fog computing networks," in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, 2016, pp. 1-6.
[181] H. H. R. Sherazi, R. Iqbal, and L. A. Grieco, "AREA: Adaptive Resilience Algorithm for clustering in Vehicular Ad-hoc Networks," in *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, 2019, pp. 1-3.