Optimised Green IoT Network Architectures

Zaineb Talib Saeed Al-Azez

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

The University of Leeds
School of Electronic and Electrical Engineering

July 2018

The candidate confirms that the work submitted is 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.

Chapter 3 is partially based on the work from:

Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Virtualization framework for energy efficient IoT networks," *2015 IEEE 4th International Conference on Cloud Networking (CloudNet), Niagara Falls*, ON, 2015, pp. 74-77.

Prof. Elmirghani, the supervisor, suggested the study of cloud computing platform for IoT networks. He also checked the MILP model and revised the paper. Dr. Lawey, worked with the student on the MILP model development and revised the paper. Dr. Taisir, helped develop the first virtualization MILP model. The PhD student wrote the literature review, developed the Mixed Integer Linear Programming model.

Chapter 4 is based on the work from:

Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Energy efficient IoT virtualization framework with passive optical access networks," *2016 18th International Conference on Transparent Optical Networks (ICTON)*, Trento, 2016, pp. 1-4.

Prof. Elmirghani, suggested to expand our original IoT network by adding PON as an access network. Dr. Lawey, worked with the student on the MILP model development and revised the paper. Dr. Taisir, helped develop the first virtualization MILP model. The PhD student wrote the literature review, developed the Mixed Integer Linear Programming model.

And:

Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "PON Supported IoT Virtualization framework".

Prof. Elmirghani, suggested to expand our IoT network by adding PON as an access network, He also checked and helped develop the MILP model, suggested the heuristic steps and revised the paper. Dr. Lawey, worked with the student on the MILP model development. Dr. Taisir, helped develop the first virtualization MILP model. The PhD student wrote the literature review, developed the Mixed Integer Linear Programming model and the heuristic.

Chapter 5 is based on work from:

Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Energy Efficient IoT Virtualization Framework with Peer to Peer Networks".

Prof. Elmirghani, the supervisor, suggested the study of P2P communication platform with IoT network. He also checked and helped develop the MILP model, suggested the heuristic steps and revised the paper. Dr. Lawey, suggested to study processing in IoT by using P2P platform, worked with the student on the MILP model development and revised the paper. Dr. Taisir, helped develop the first Peer to Peer MILP model. The PhD student wrote the literature review, developed the Mixed Integer Linear Programming model and the heuristic.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

© 2018 The University of Leeds and Zaineb Talib Saeed Al-Azez

Dedication

*This thesis is dedicated to my daughter **Retaj** and my husband **Ahmed**.*

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Jaafar for the unlimited support of my PhD study and related research, for his patience, motivation, kindness and immense knowledge.

I would like to acknowledge Dr. T. H. El-Gorashi, my co-supervisor for her support, kindness and useful discussions especially during thesis writing.

I would like to thank Dr. Ahmed Lawey for his guidance, great support and constructive suggestions.

I would acknowledge the Higher Committee for Education Development in Iraq (HCED) for funding my PhD.

I want to express my deep gratitude to my beloved and dearest friend Dr. Katherine Taylor, you will be always my hero who changed my life forever. Thank you very much for your unlimited support and encouragement. Your friendship was and always will be the best thing happened in my life.

I would like to express my appreciation to my beloved friends, my soul mate Rwayda, my kindest friend Dhamyaa and my sweet friend and little sister Marwa. Your friendship comforted me and helped me escape many hurdles during the years of my study.

I awe my deepest gratitude towards my beloved family back home, my father Talib, my mother Saleema and my kind elder brothers Abdul-Razaq and Mushtaq for their unconditional support, encouragement and love and for supporting me spiritually throughout writing this thesis and my life in general.

Finally, I am in debate to my young family. My daughter, the apple of my eye and my angel Retaj. I thank her for making my life so fulfilled and rewarding and for her support and patience during my thesis writing. I dedicate my final acknowledgements to the love of my life, my dear husband Ahmed for his continued support and understanding during my pursuit of PhD degree that made the completion of thesis possible. I greatly value his contribution and deeply appreciate his belief in me.

Abstract

The work in this thesis proposes a number of energy efficient architectures of IoT networks. These proposed architectures are edge computing, Passive Optical Network (PON) and Peer to Peer (P2P) based architectures.

A framework was introduced for virtualising edge computing assisted IoT. Two mixed integer linear programming (MILP) models and heuristics were developed to minimise the power consumption and to maximise the number of served IoT processing tasks. Further consideration was also given to the limited IoT processing capabilities and hence the potential of processing task blockage. Two placement scenarios were studied revealing that the optimal distribution of cloudlets achieved 38% power saving compared to placing the cloudlet in the gateway while gateway placement can save up to 47% of the power compared to the optimal placement but blocked 50% of the total IoT object requests.

The thesis also investigated the impact of PON deployment on the energy efficiency of IoT networks. A MILP model and a heuristic were developed to optimally minimise the power consumption of the proposed network. The results of this investigation showed that packing most of the VMs in OLT at a low traffic reduction percentage and placing them in relays at high traffic reduction rate saved power Also, the results revealed that utilising energy efficient PONs and serving heterogeneous VMs can save up to 19% of the total power.

Finally, the thesis investigated a peer-to-peer (P2P) based architecture for IoT networks with fairness and incentives. It considered three VM placement scenarios and developed MILP models and heuristics to maximise the number of processing tasks served by VMs and to minimise the total power consumption of the proposed

network. The results showed that the highest service rate was achieved by the hybrid scenario which consumes the highest amount of power compared to other scenarios.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ABI	Allied Business Intelligence
AE	Active Ethernet Network
AON	Active Optical network
ATM PON	Asynchronous Transfer Mode Passive optical Network
BPON	Broadband Passive optical Network
co	central office
DHTs	Distributed Hash Tables
DSL	Digital Subscriber Line
EEPIV	Energy Efficient PON supported IoT Virtualisation
EEVIN	Energy Efficient Virtualisation for IoT Networks
EEVIPN	Energy Efficient Virtualised IoT P2P Networks
EPON	Ethernet Passive optical Network
FSAN	Full-Service Access Network
FTTX	fiber-to-the-x
GEM	Gigabit Passive optical Network Encapsulation Method
GPON	Gigabit Passive optical Network
GPS	Gateway Placement Scenario
IaaS	Infrastructure as a Service

ICT	information and communication technology
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union-Telecommunications
M2M	Machine to Machine
MILP	mixed integer linear programming
NG-PON	Next Generation-PON
ODN	Optical Distribution Network
OFDM-PON	Orthogonal Frequency Division Multiplexing Passive optical Network
OLT	Optical Line Terminal
ONU	Optical Network Unit
OPS	Optimal Placement Scenario
P2MP	point to multi-points
P2P	Peer to Peer
PaaS	Platform as a Service
PON	Passive Optical Network
POTS	Plain Old Telephone Service
SaaS	Software as a Service

TDMA	Time Division Multiple Access
TDM-PON	Time Division Multiplexing PON
VM	Virtual Machine
WDM-PON	Wave Division Multiplexing PON
WLAN	Wireless Local Area Network

Chapter 1: Introduction

As a result of the exponential growth of the Internet, the CO_2 emissions and energy consumption of information and communication technology (ICT) networks are undergoing a dramatic increase. This increase is one of the significant challenges that may hinder the expanding scale of the internet. According to [1], the electricity consumption of servers costed 7.2 billion dollars globally in 2005 and the information technology (IT) infrastructure in the US consumed an estimated 61 billion kWh of energy in 2006 as explained in [2]. Moreover, ICT generates an estimated 2% of the global CO_2 emissions [2]. The European Union reported that in order to keep the rise in the global temperature under the 2°C, the volume of emissions should be decreased by 15%-30% before the year 2020 [3]. Consequently, more attention must be given to improving energy efficiency and sustainability within the Internet and the ICT industries.

IoT is the concept of making objects smarter through connectivity. Its ultimate purpose is to connect all physical objects to the Internet in order to exchange information. As a consequence, communications across the Internet would then take on three forms: human-human, human-things and things-things.

IoT represents a major evolution in legacy data communication. It is predicted that there will be 75.44 billion IoT interconnected devices in 2025 [4]. This growing level of connected devices has paved the way for futuristic smart applications in healthcare, agriculture, transportation, manufacturing, smart homes and machine-to-machine (M2M) communications [5], [6]. There are, however, many key challenges such as reliability, security, interoperability and scalability [7]. In addition, one of the main challenges that must be confronted by IoT architects is the energy efficiency and greening the networks [8], which is currently garnering attention in both academic and industrial arenas. IoT is also expected to benefit from the wide spectrum of proposed energy efficient network solutions. Cloud computing was investigated as one of the solutions for energy efficiency in networks and data centres [9]-[13]. However, with the exceptional amount of data generated by IoT objects (expected to generate 2.3 trillion gigabytes of data every day by 2020) [5], emerging cloud computing with IoT gives rise to new challenges. Among these challenges is the demand for high communication bandwidth, security, and latency requirements [14].

A number of other solutions were suggested, one of which was processing the IoT data by the IoT object itself or by devices in the nearest layer to the objects. According to the estimations of the Allied Business Intelligence (ABI) research, processing and storing 90% of the data created by the endpoints will be done locally rather than within conventional clouds [14]. Since complicated data processing tasks cannot be completed by most IoT devices and sensors because of their limited capabilities, edge computing was proposed to provide more efficient resources.

Energy constraint is a dominant trait of most IoT end nodes, adding to that, the wireless modules are well known for their hunger for energy. Therefore, processing and computation offloading to the edge of the network is a key method to save energy [5, 15]. One of the recent paradigms for edge computing is the cloudlet [16]. Cloudlets are used to extend the computing capabilities of conventional cloud computing in data centres to the edge of networks. This can assist in tackling the resource poverty of IoT object computation [17].

Based on the essential concept of edge computing, this thesis investigates the optimisation of IoT data processing to minimise the power usage and improve the energy efficiency by presenting several architectures for IoT networks.

Emerging edge computing in the proposed IoT networks presented in this work is imperative to support the confined IoT objects' resources by providing them with high processing capabilities through exploiting VMs based mini-clouds (cloudlets). As a result, edge computing frees the upper layers of the network from the burden of data processing by reducing the traffic flow in the network and reducing its power consumption. Optimising the number of VMs, their location, and the distribution of the cloudlets at the edge of the network is the main goal of this thesis to save the total power consumption of IoT network. The cooperative nature of IoT can extend P2P communication systems [18] by providing P2P processing which is introduced in this work. P2P processing in IoT networks can add another degree of energy saving as investigated in this thesis. The investigation and the study in this thesis are carried out by developing MILP models for the proposed architectures, where the results are evaluated and verified using real time heuristics models.

1.1 Research Objectives

The presumption in this thesis is that virtualisation can improve energy efficiency in IoT networks. To investigate this presumption, the main objectives are to:

1. Design an energy efficient virtualisation IoT platform considering the optimum placement of VMs in mini-clouds (cloudlets) to process the IoT objects queries as close as possible to the objects. In addition, explore the effects of centralised vs. decentralised edge computing on the proposed design in term of power consumption reduction.

2. Evaluate the potential power savings when IoT networks are supported by PON as it is one of most favourable access networks in term of high bandwidth, long access distance and power consumption.
3. Investigate the use of P2P distributed systems in distributing the computational burden of the required tasks in IoT networks to peer nodes (IoT objects and VMs). Moreover, study the impact of this type of distribution on IoT network performance in terms of power consumption and system capability when serving the required tasks.

1.2 Original Contributions

The following are the main contributions of this thesis:

1. A MILP model was developed to design an energy efficient edge computing platform for IoT networks. The number and location of mini clouds, as well as the placement of VMs were optimised to reduce the total power consumption induced by traffic aggregation and VMs processing. The optimal distribution of mini clouds in the IoT network yielded a total power savings of up to 38% when compared to processing IoT data in a single mini cloud located at the gateway layer.

2. A heuristic (Energy Efficient Virtualisation for IoT Networks), (EEVIN) was developed for real-time implementation of the energy efficient edge computing platform for IoT networks. The power savings and performance achieved by the heuristic approach were compared with those achieved by the MILP model.

3. The dependency of VMs' CPU utilisation on the number of served IoT objects and the power consumption and blocking percentage of IoT object requests was investigated by developing a MILP. A heuristic was developed to validate the MILP investigation. The results showed that locating the mini cloud at the gateway layer can yield a total power saving up to 47% compared to the optimal distribution of mini clouds in the IoT network with a 50% blocking rate of the IoT objects' requests as the CPU utilisation of the gateway was not enough to satisfy all the IoT objects.
4. A framework for an energy efficient edge computing platform for IoT accompanied by a PON was developed. This framework was evaluated using a MILP model. Energy efficiency was achieved here by optimising the placement and number of the mini clouds and VMs as well as by utilising energy efficient routes. The results indicated that up to 19% of the total power can be saved by using energy efficient PONs and serving heterogeneous VMs compared to using energy inefficient OLTs and serving highly homogenous VMs. A heuristic was developed with comparable power saving to that of the MILP model.
5. An energy efficient P2P platform for an IoT network was proposed. The peers in this work were objects and VMs. According to which peers can process the task query (only VMs; or only objects; or by both), three scenarios were considered to investigate the power consumption savings and the system capability of task processing. A MILP model was developed to investigate these scenarios. The results showed that the highest performance in terms of executing tasks was achieved by the hybrid scenario, where both objects and VMs served the required tasks. This scenario can serve up to 76% of the

processed task requests, but with higher energy consumption as compared to the other scenarios. For real time implementation, a heuristic was developed to mimic the MILP model behaviour.

1.3 Related Publications

The following conference papers have been published:

1. Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Virtualization framework for energy efficient IoT networks," *2015 IEEE 4th International Conference on Cloud Networking (CloudNet), Niagara Falls*, ON, 2015, pp. 74-77.
2. Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Energy efficient IoT virtualization framework with passive optical access networks," *2016 18th International Conference on Transparent Optical Networks (ICTON)*, Trento, 2016, pp. 1-4.

Other publications prepared in this work

- Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "PON Supported IoT Virtualization framework".
- Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Energy Efficient IoT Virtualization Framework with Peer to Peer Networks".

1.4 Thesis Outline

Following this chapter, the thesis is organised as follows:

Chapter 2 provides an overview of the background needed for the main topics addressed in this thesis, including a review of the architectures of IoT systems, IoT applications, the challenges experienced in developing IoT systems and the research efforts to improve the energy efficiency of IoT systems. It also reviews cloud computing, access networks, PON, peer-to-peer communication systems as well as MILP modelling and optimisation.

Chapter 3 introduces an energy efficient edge computing MILP model and its results considering VMs CPU utilisation independently of the number of served IoT objects. It proposes a heuristic for energy efficient for IoT network virtualisation (EEVIN1) and compares its performance to the MILP model. Following this, it introduces a new MILP model that investigates the possibility of blocking some service requests in case the VM's CPU utilisation increases with increase in the number of served IoT objects. It studies the impact of this MILP model on the network performance and energy efficiency and compares its results with the first MILP model. It optimises the number and location of the distributed cloudlets and the placement of the VMs by introducing two scenarios of optimally distributing cloudlets in the network elements and centrally locating a single cloudlet in the gateway. Finally, it proposes a second heuristic (EEVIN2) to mimic second MILP model behaviour and compares its results to the MILP model and to the first heuristic (EEVIN1).

Chapter 4 provides an extension to the energy efficient edge computing framework presented in Chapter 3 by adding a PON. It introduces a MILP developed to optimise the placement of mini clouds and VMs as well as utilisating energy efficient routes. It proposes an Energy Efficient PON supported IoT Virtualisation (EEPIV) and compares its results to the introduced MILP model.

Chapter 5 presents an energy efficient P2P platform for IoT networks and an associated MILP model. An investigation of power savings and the system capability in terms task processing is undertaken through three scenarios: The first scenario is the VMs only scenario, where the task requests are processed using hosted VMs in relays only. The second scenario is the objects only scenario, where the task requests are processed using the IoT objects only. The last scenario is the hybrid scenario, where the task requests are processed using both IoT objects and VMs. This chapter proposes a real time heuristic (Energy Efficient Virtualised IoT P2P Networks EEVIPN) and it compares its results to the MILP model results.

Chapter 6 summarises the main contributions of the thesis and provides recommendations and possible directions of future work.

Chapter 2 : Overview of IoT Networks

2.1 Introduction

IoT can represent a revolutionary future of data communication. The term IoT was used for the first time by Kevin Ashton in 1999 [19]. IoT is concerned with the interconnection of all physical things/objects through the Internet. These objects are wired or are wirelessly connected with unique identifiers. According to this principle, things/objects have the ability to communicate and cooperate with each other in order to achieve their common purposes. According to International Telecommunication Union (ITU) vision of the IoT [20], the main purpose of IoT is the connectivity of any one at any time in any place which is done by using any path, network and any service as illustrated in Figure 2.1 [21]. Consequently, IoT intends to make things more intelligent.

Things/objects should have the ability of interaction with the environment through sensing, actuating or simple interaction with people. Also, they should be able to compute and process a variety of services with different degrees of complexity. In addition, global communication through the Internet is required to guarantee interoperability. Finally, these things/objects should be supplied with different energy resources and this is done in order to achieve the aforementioned capabilities [22]. In fact, these capabilities are enabled by technologies such as RFID, M2M, WSN, mobile Internet and others. These technologies generate real world information and use such information in different categories of IoT applications [20].

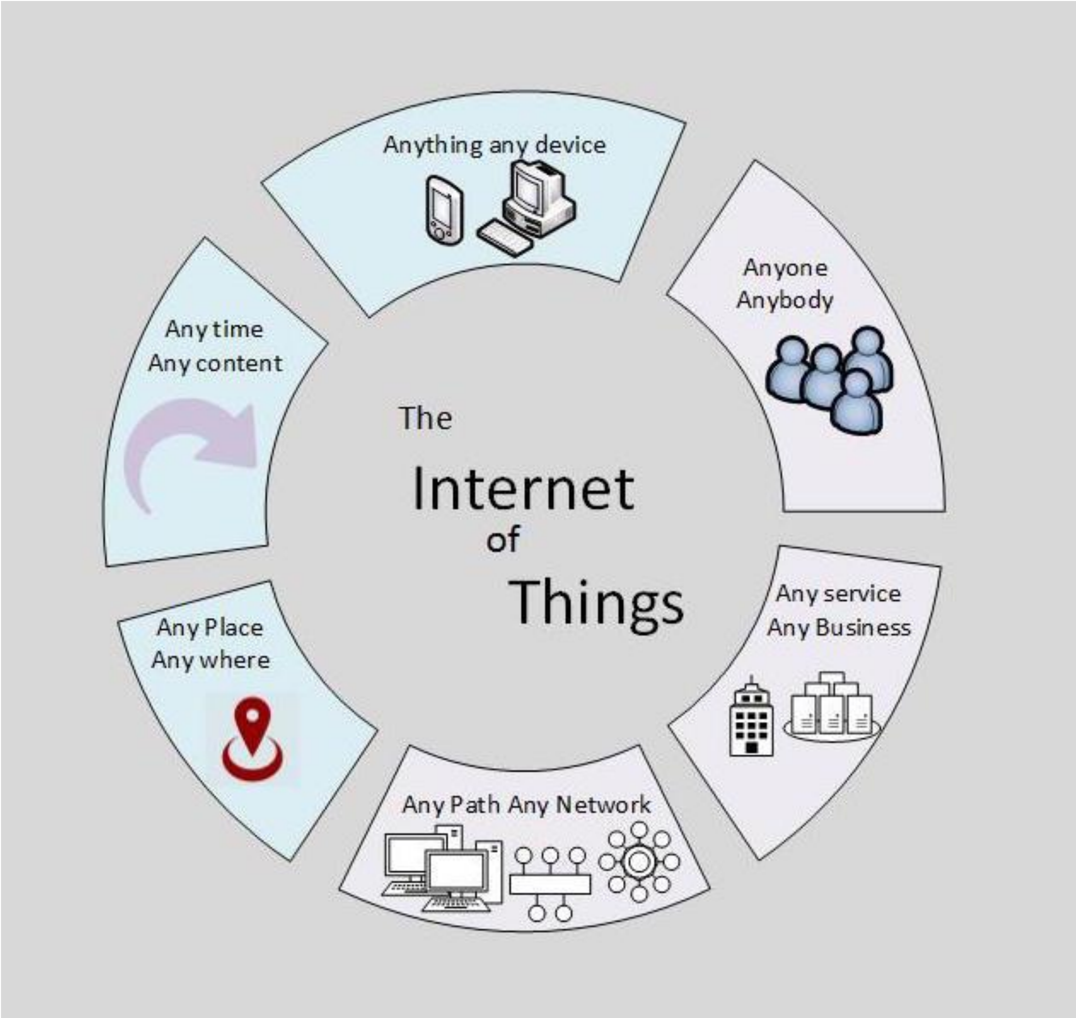

Figure 2.1 IoT concept [21]

2.2 IoT Architecture

IoT architecture can be described as a layered architecture as shown in Figure 2.2. This layered architecture has emanated from the mismatch between the original architecture of the Internet and the current and future demands. These demands result from connecting billions of objects through IoT.

The IoT architecture consists of 5 layers as illustrated bellow:

- **Perception Layer:** It can be called also the device layer [23] as it consists of physical objects and sensors. The sensors' information are gathered and

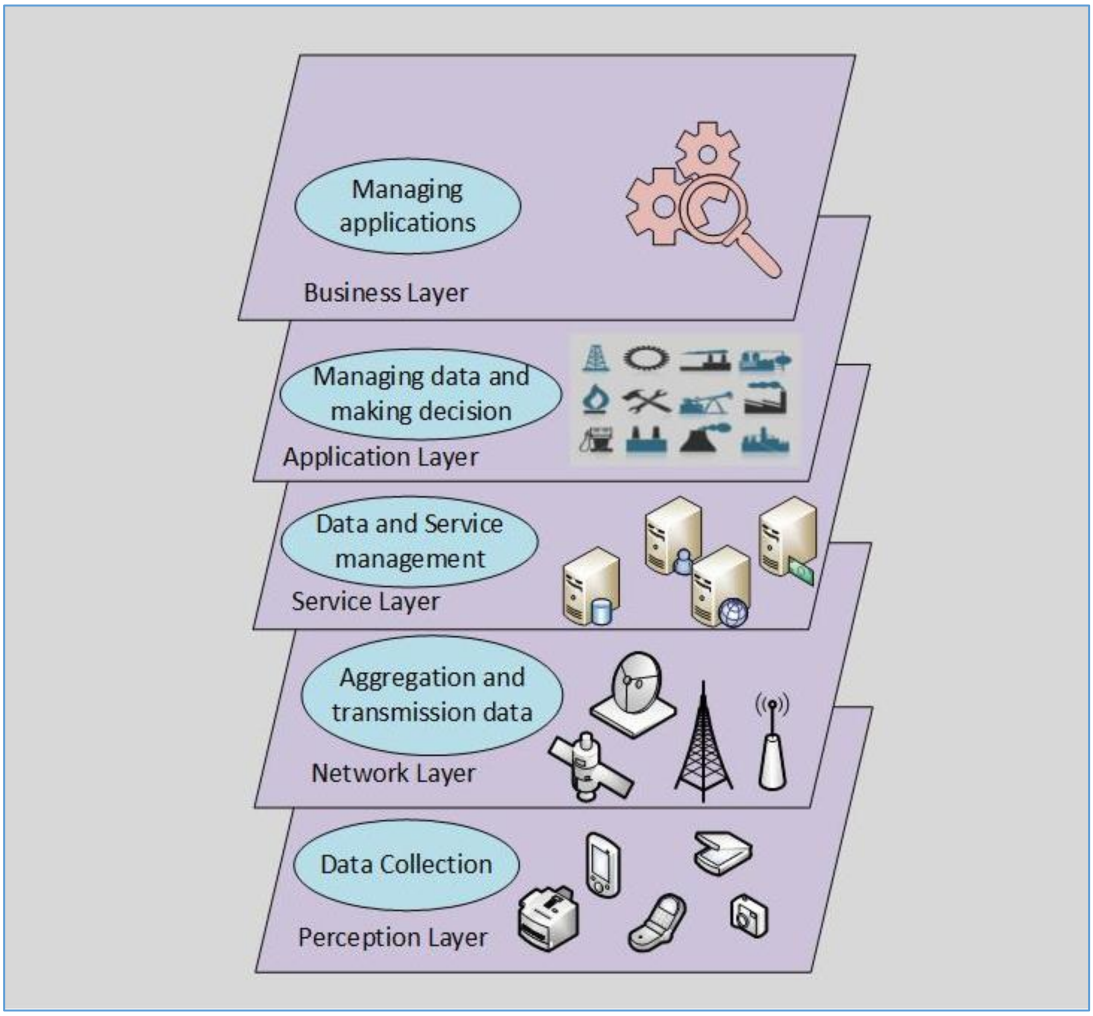

Figure 2.2 IoT layered architecture

exchanged in this layer by using different short distance transmission network technologies [19].

- **Network Layer:** This layer can be described as the transmission layer [23] because its function where it provides aggregation and transmission of the information received from the lower layer (perception layer). The network layer can be constructed from different communication network transmission mediums like wireless or wired communication networks etc [19].

  Using such technologies allows for aggregation and sharing of information between heterogeneous IoT networks. Some technologies can be used in this layer in order to process the aggregated information [19] such as using cloud platform for computing.

Actually, processing the information is one of the responsibilities of some of the upper layers. But sometimes the network layer and its upper layers can be used in accomplishing this task. In fact, this is depends on the technology used in the network.

- **Service Layer:** It is also called the middleware layer. Its basic function is controlling and managing different types of services that are implemented by the IoT devices. It also processes the information and makes decisions according to the results of this processing. It keeps the data in databases for storage purposes.
- **Application Layer:** It is responsible for the applications in IoT and for managing all issues that are related to these applications like performing information storage or making decisions. These tasks are based on the processed data that comes from the lower layer (middleware layer). Similar to the middleware layer, the intelligent processing of the information in the application layer can be done by the aid of cloud computing. The applications in IoT can play an important role in our life such as medical applications, agriculture applications, environmental applications etc [19].
- **Business Layer:** The need for applications management comes from the diversity of applications in IoT. This diversity results from using different networks architectures in IoT systems. Management is done by the business layer. Actually, this layer can be considered as the IoT manager. The most basic task that is performed by this layer is building a business model. This model is based on the data that comes from the processing in the lower layer.

In fact, this data also helps in designing future business strategies. In addition to the previous tasks, providing privacy for users is one of the tasks of the business layer [24].

2.3 IoT Applications

According to [25], the IoT applications categories are specified into four domains (personal and home, enterprise, utilities and, finally, mobile). This classification is based on many attributes such as the availability of the network, its coverage area, its scale and the impact of network users. The first domain, personal and home, is directly controlled and used by the network owners. For example, the control of home equipment applications. Similarly, the enterprise domain is also controlled by the owners but in a work environment such as managing light in a building. On the other hand, the information of the utilities domain do not focus on the user consumption. They are usually used for optimisation of the introduced service. For example, the continuous monitoring of electricity consumption will help modify the consumption behaviour of users in order to achieve an efficient energy consumption system. Finally, the mobile domain works with the applications that are related to smart transportation and traffic issues or monitoring transported items etc.

The work in [19] classified the applications according to their use into three basic modes. The first mode is object smart tag mode. It relates to object identification. RFID is one example of applications used in this mode. The next mode is the environmental monitoring and object tracking mode. Using sensor networks is essential in this mode in order to collect information about the objects status and act according to the information. For example, developing solutions for some environmental issues through the use of the collected information from the distributed environment sensors. The third mode is the intelligent control of object mode which is related to smart networks and cloud computing. This mode is similar to the previous mode in performance. It takes decisions in controlling objects according to the collected sensors information.

On the other hand, [26] identified three broad areas for application domains. The first considered domain is the society. Healthcare and smart cities are examples of this domain. The other significant broad area is the environment. It is related to applications like disaster alerting and smart environment etc. The last significant domain is industry, for example industrial control and retail applications.

Among all these application categories, there is a common trait which is intelligence. For example, in automation, data is collected and used to take decisions according to current status. Also, mobility can be used as part of the intelligence to ensure the movement of applications through different domains. In addition, IoT applications should have plug and play features to support the scalability of IoT systems [27].

2.4 IoT Challenges

The exponential growth of IoT deployment through the dramatic increase in connected devices, opens a lot of opportunities in different fields. At the same time, this significant growth is faced by a number of key challenges. The key challenges are described below:

- Standardisation and Interoperability: The huge deployment of any technology or service in IoT requires interoperability with other technologies and services. Interoperability in any IoT system is achieved by standardisation.

Making open standards undertakes seamless exchange of information among various objects and applications that are made by different manufactures.

- Privacy and Security: IoT systems require more security and privacy than the that required for traditional networks due to the heterogeneous nature of IoT systems. The variation of connected things/objects makes IoT more vulnerable to be attacked [26]. So, issues like data encryption, things safety, information confidentiality become major concerns when designing a new IoT system.
- Scalability Management: The continuous increase of interconnected devices in IoT systems results in increasing the importance of managing these devices. Actually, managing the data traffic among these devices represents a real key challenge in designing the IoT systems [26].
- Green IoT [23]: The wide deployment and complexity of IoT systems results in more power consumption. Consequently, this consumption results in increase in cost increasing and environmental impacts.

Collectively, all these factors make the green IoT as a crucial issue that should be addressed.

2.5 Energy Efficiency in IoT

Green IoT network design is one of the major challenges experienced in the IoT networks where its importance is driven by the dramatic growth of the IoT deployment. This growth results in more connected devices that consume energy. More energy consumption leads to an increase in the energy costs and causes more CO_2 emissions. All these aforementioned factors urge researchers in both academia and industry to investigate a number of technologies to cope with the surge in the energy consumption.

There is a recent trend in research towards proposing IoT platforms based on local computing close to the objects such as fog and edge computing. The authors of [28] studied and compared the energy consumption of distributed nano data centres (nDCs) with the energy consumed by centralised data centres for content and applications distribution. They investigated the factors that result in saving energy by using the nDCs such as the attached access network and the type of installed applications. In [29], a combination of fog computing and microgrid is proposed in order to reduce the energy consumed by IoT applications. A set of measurements and experiments were implemented considering different processing and traffic requirements. In [29], it was shown that dynamic decisions can be made by the proposed IoT gateway to minimise the consumed energy by choosing the most efficient location for processing a task in the fog or in the cloud. This decision is affected by the type of deployed IoT application, weather forecasting and the availability of the renewable sources.

The wireless sensor networks (WSN) represent one of the most basic building technologies of IoT. But, the integration between WSN and IoT is considered as a challenge that results from the capability of the low-power devices to connect to the Internet. One of the recent solutions to this problem is through the use of 6loWPAN technology. This technology is used in [30] with the goal of minimising the power consumption. Hierarchical directional routing was proposed in order to solve the sleeping mode problem. The efficiency of this routing method and its accuracy were analysed with examples and reduction in power consumption was achieved. This was done by optimising the number of hops between any two communicating nodes in the network.

Achieving a green deployment of large scale IoT is proposed by [31] as a solution for the problem of direct integration between WSN and IoT. This deployment was based on a general hierarchical framework. The energy consumption was minimised by using optimisation methods in this hierarchical system. The optimisation was not concerned with the energy consumption only but was based on load balancing and cost.

2.6 Cloud Computing

The large-scale proliferation of storage and processing services required by users urges the researchers to propose a number of substantial computing models. The most significant and successful paradigm is cloud computing. It is an adaptable environment [32] as it provides services to end users as required (on-demand).

Cloud computing is a term that describes the provision of computing resources such as storage and processing services to user through the Internet. Rapid and efficient services are provided through the world by establishing very large virtualised data-centres. In 2014, the International Data Corporation declared that cloud services and their enabling technologies will be increased by 25% [33].

2.6.1 Deployment Model of Cloud Computing Service:

The deployment models of cloud computing services are illustrated in Figure 2.3.

1. **Public Cloud:** the public clouds is used by the general users or large organisations. It allows the users to use the storage, applications and other services that are provided by the cloud providers. In fact, it is based on a pay-

per-use model [34] and that means the payment is charged according to the amount of the provided services and the time duration these services are used [34, 35]. Such types of cloud models can be considered a least cost option compared to other types. On the other hand, because of the generality of its users, public clouds are less secure and this is considered a significant issue that should be addressed when services based on public clouds are provided to the users.

2. **Private Cloud:** This type of models is used by one organisation only and the cloud is operated and managed by the organisation itself or a third party [32]. The security of the systems used is enhanced as a result of the management mechanism of the resources and applications of this model [34]. Also, another advantage that results from this dedicated network control, is the optimum customisation of the provided resources to meet the needs of the organisation customers[35].

   According to the type of cloud computing provider, the private cloud can be classified to two categorises:

   - **On- Premise Private Cloud:**

     The cloud provider in this format is the data centre of the organisation itself. In fact, it can provide a high level of security but on the other hand there is a limitation in scalability and size of the cloud resources [35].

   - **Externally-Hosted Private Cloud:**

     The cloud resources in this model are provided externally. The cloud here is an external environment but with high privacy level [35].

3. **Community Cloud:**

   This type of clouds is constructed by more than one organisation with the same interests. The polices and requirements of these organisations are shared between them. In addition, they are managed either by one of these organisations or by a third party [34].

4. **Hybrid Cloud:**

   It is a combination of two types of cloud models (the public cloud and private cloud model). The provided cloud resources depend on the size of the required demand. If the demand is computing intensive, then the resources are provided from the public cloud instead of the private cloud.

2.6.2 Characteristics of Cloud Computing:

According to the national institute of standards and technology (NIST), cloud computing has five significant characteristics [36]:

1. **On-Demand Self Service:**

   The cloud services provision the users with resources according to the users' needs. Actually, it is the same concept as utilities provisioning. For example, provisioning of water or electricity and the charges depend on the amount of usage resources [36].

2. **Broad Access:**

   Cloud consumers can access the cloud computing resources by using different platforms such as smartphones and laptops [36].

3. **Resource Pooling:**

Cloud computing enables the sharing of pooled storage and computing resources among cloud consumers. These pooled resources are served by the same hardware [36].

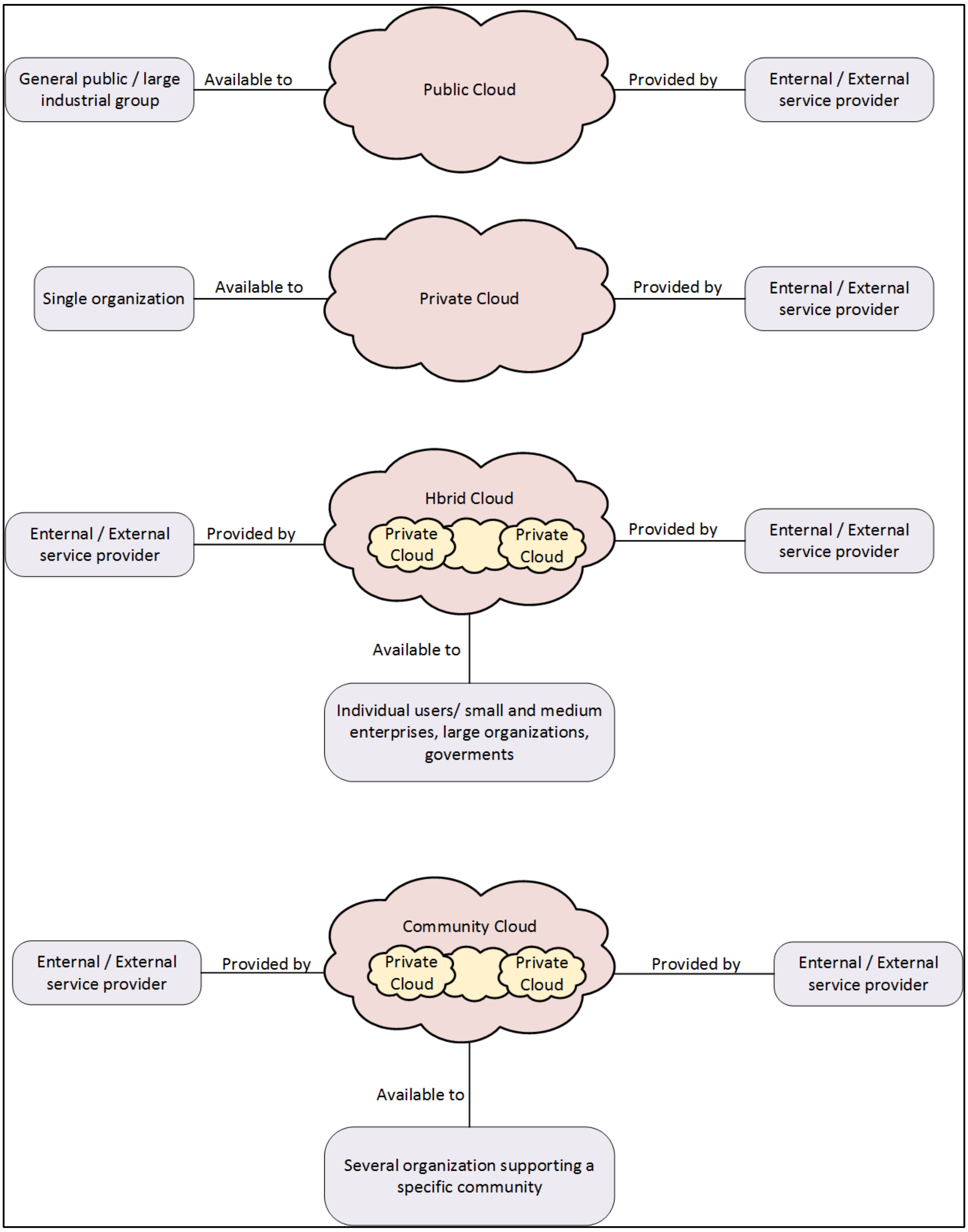

Figure 2.3 Cloud deployment models [36]

2.6.3 The Advantages of Cloud Computing:

1. Using applications in the cloud environment by the users requires only Internet connection and web browser. So, it is easier than using the same applications in local computers [34].
2. The rapid elasticity of the cloud computing does not just result in enhancing the performance of the deployed applications but also results in reducing the costs of deploying these applications.
3. Having offsite backup of the users' data or organisations data in cloud environment is very substantial and helpful for disaster recovery situations [34].
4. The cloud consumers are provisioned with cloud services that have a high level of reliability and availability because of the professional management by the cloud providers [37].
5. The environmental impacts and the increasing energy consumption are the main results of the used computing systems. Using the cloud computing environment results in reducing these harmful effects.
6. Using cloud computing results in reducing the costs of building new systems especially for the organisations. This cost reduction comes from avoiding the need for installing and managing underutilised hardware and software.

2.7 Cloud computing with IoT

The integration of a cloud computing platform with IoT becomes a very significant issue; because the rapid prevalence of IoT technologies and applications in addition to the vast current and expected amount of data that come from billions of connected objects through internet.

This resources provisioning is the key for efficient utilisation of the aggregated data from the IoT objects.

2.7.1 Cloud Computing Services Models

Cloud computing services are provisioned according to the system requirements. So, these services can be categorised into three types:

- Software as a Service (SaaS): can also be called application service provider [38]. In this service model, the applications software are provided by remote servers where the user can use these applications as services for example, email services [39]. In IoT, SaaS model is used for monitoring services application domain [27].
- Platform as a Service (PaaS): this service model provides a development environment for the user that can be accessed to develop, test and deploy the applications and cloud services [40]. One of the examples of this model is Google App Engine [40]. The ability to access the IoT data and control services is provided by this model [27].
- Infrastructure as a Service (IaaS): this model is usage-based payment computing infrastructure [38]. Commonly, it is provided for the users as computing, storage and virtualised servers like Amazon Elastic Compute Cloud services [39]. In IoT, IaaS is suitable for resources access models and for sensors and actuator business models [27].

2.7.2 Advantages of Cloud-IoT Integration

In fact, the integration of cloud computing with IoT systems brings significant advantages to IoT. First of all, the responsibility of the cloud for managing the required IoT resources improves the reliability of the IoT systems. Secondly, the heterogeneity of the connected objects/things in IoT systems can be supported by the cloud. In Addition, scalability is one of the most important features that the IoT systems should be characterised with reference to. Using the cloud supports this significant IoT characteristic and this comes from the cloud's ability to manage resource allocation dynamically. Finally, cloud computing is characterised as on-demand service provisioning [21] as stated previously in its definition. This essential attribute plays an important role in fulfilling the users' requirements and maximising the resources utilisation concurrently.

2.7.3 Challenges of Cloud-IoT Integration

The integration of IoT and cloud experiences some challenges that should be addressed. For instance, accomplishing a reliable communication among objects/things and applications. In additional, issues like the capability of sharing resources among different cloud systems and dealing with the security and privacy issues are also considered as significant challenges that are faced by the cloud-IoT integration.

2.8 Access Networks

The essential role played by the Internet in people everyday lives, for example in education, business, social life and entertainment leads to revolutionary growth in generated traffic [41]. The surge in traffic demand along with the continuous and dramatic increase in end users results in increasing the bandwidth requirements with broadband services [41, 42]. The access network is a section of the telecommunications network which connects the service provider to the end users [42]. Access networks fall into two basic categories, wireless and wired technologies.

2.8.1 Wireless access networks

Wireless technologies are widespread because of their low-cost deployment and their ability to tackle the mobility challenges in access networks [42, 43]. The most important standards of wireless access technologies are WiFi (802.11) and WiMax (802.16). The coverage area of WiFi comprises local areas such as houses or enterprises. WiFi has been used for Wireless Local Area Network (WLAN) for the last 15 years with limited range of 100-200 m [42, 44]. The maximum data rate that can be achieved currently in commercial WiFi systems is 600 Mb/s, however higher data rates are planned [42]. WiFi operates in the unlicensed spectrum at of 2.4 GHz [44] with 20 MHz or 40 MHz as channel bandwidth (but also operates in the 5 GHz window) [42]. On other hand, WiMax covers wider areas and for that it is used in Wireless Metropolitan Network (WMAN). WiMax covers (5km-15km) for mobile wireless stations and up to 50 km for fixed wireless stations with 70 Mbps as a maximum data rate [45]. The range of frequency bands for fixed standard is 2 to 11 GHz while, it is below 6 GHz for the mobile standard [44]. The bandwidth of channels in WiMax is 1.25 – 20 MHz [42].

2.8.2 Wired access networks

Wired access networks fall into two basic categories, copper and fiber. Copper is the earliest access network technology particularly Digital Subscriber Line (DSL) which started from deploying Plain Old Telephone Service (POTS) and ended with deploying broadband access to end users [46]. The performance of all types of DSL is quietly affected by the noise level which depends on the length of the copper loop [44, 48]. The length of the copper loop varies according to the type of the used DSL technology. For example, ADSL can provide 24 Mb/s over 5 km and one of the latest DSL technologies (G.fast 106) can provide 1 Gb/s over 250 m [46].

The other wired access network technology is fiber. Non-conducting, light weight and small diameter of fiber cables lead to making the fiber access technology one of the most powerful rivals amongst all deployed access technologies at present time [48]. Fiber-to-the-x (FTTX) is a generic term used to describe fiber access technologies [49-52]. Depending on how close the deployed fiber is to the user, x in (FTTX) can represent the following [49-52]:

- Node in (FTTN) where the street cabinet is the final destination of the fiber cable which can be far from the end user, several kilometres.
- Cabinet in (FTTC) where it is similar to (FTTN) but with much closer distance to end user.
- Home in (FTTH) where the user's premises or home is the final destination of the fiber cable.
- Desk in (FTTD) where the user's desk is the final destination.

According to the architecture of the optical access network, there are three basic topologies as shown in Figure 2.4: point to point fiber, Active Ethernet Network (AE) and Passive Optical Network (PON) [49, 53].

a. Point-to-point network: is an access network where a separate fiber is deployed between the central office (co) and the subscriber as shown in Figure 2.4 (a). It is a straightforward network architecture but it adds some costs compared to other topologies. The additional costs come from requiring a high number of transceivers equal to two times the total number of subscribers as each dedicated fiber link requires 2 transceivers. In addition, the total required fiber length equals the average length of each fiber link times the number of subscribers.

b. Active Ethernet Network (AE): is also called Active Optical network (AON) [52]. It is similar to point-to-point fiber as it provides each subscriber with even dedicated bandwidth, but in the topology, the signal is distributed through electronic equipment such as routers or switches as shown in Figure 2.4 (b). Using such devices reduces the length of the fiber used as there is only one link from central office to the used switch. On other hand this approach increases the number of transceivers required for the added link and also requires electric power.

c. Passive Optical Network (PON): is called passive as it uses passive components such as splitters instead of using switches or any electric equipment as shown in Figure 2.4 (c). It can be considered as a point-to-multipoint optical network. Using passive devices only leads to reducing the number of transceivers used in previous topologies to be equal to the number of subscribers, adding to the one used in the central office. these devices do not require electric power to run as they are passive.

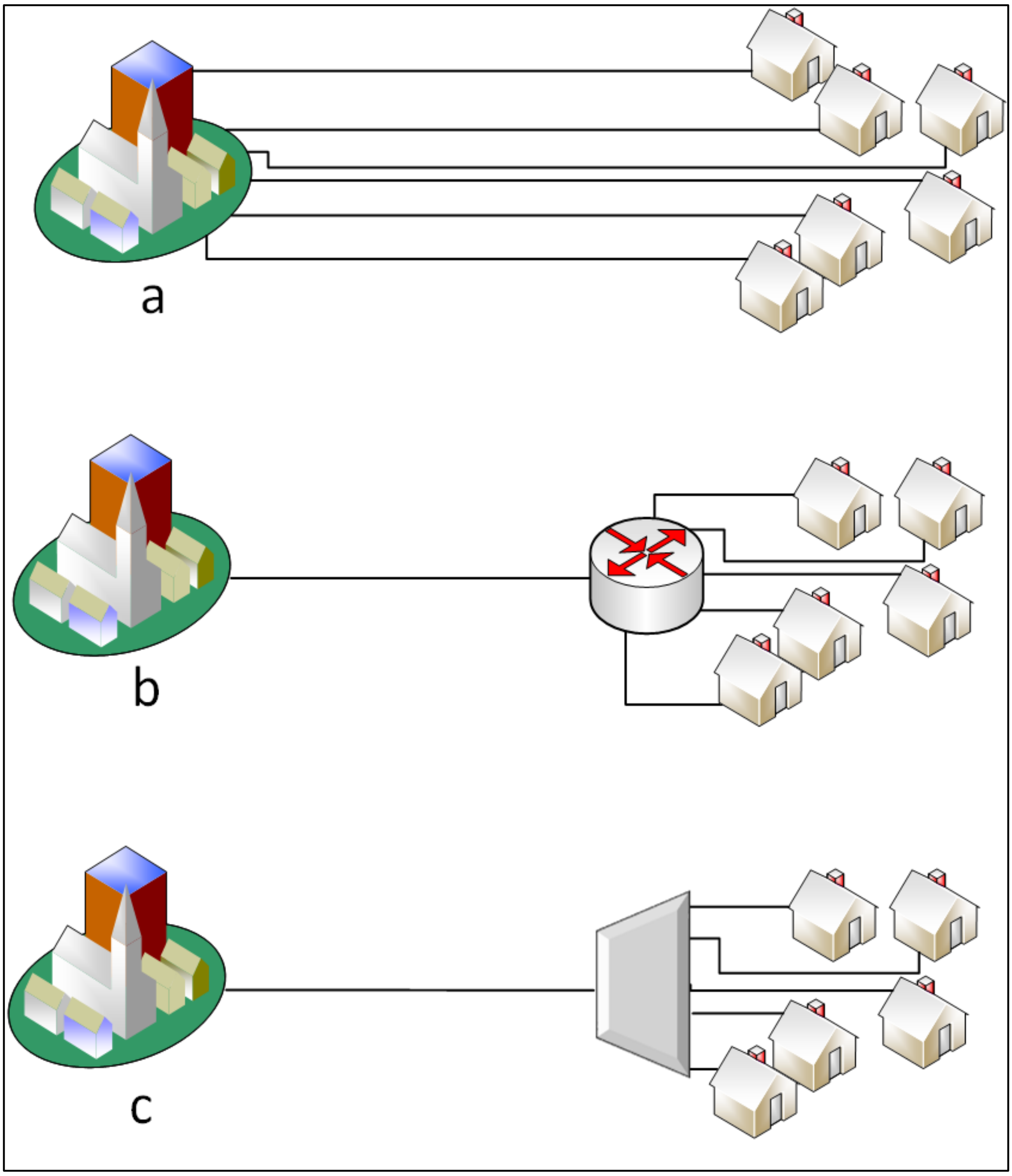

Figure 2.4 Optical access networks (a) point to point fibre (b) Active Ethernet Network (c) Passive Optical Network [49]

2.9 Passive Optical Networks (PON)

The preference of using PON as the fiber access network comes from many PON advantages such as cost and energy efficiency, long distance reach, minimising fiber deployment, allowing downstream video broadcasting as it considers point-to-multipoint architecture and it is an easily expandable architecture [42, 50, and 54]. The general architecture of PON (shown in Figure 2.5) encompasses three essential parts. The first represents the CO side, it contains the Optical Line Terminal (OLT). The other part is closer to the end users, it contains the Optical Network Unit (ONU). The third part is the middle part which connects the previous two parts, it is called the Optical Distribution Network (ODN) and it contains the optical fibers and splitters [42, 49 and 50].

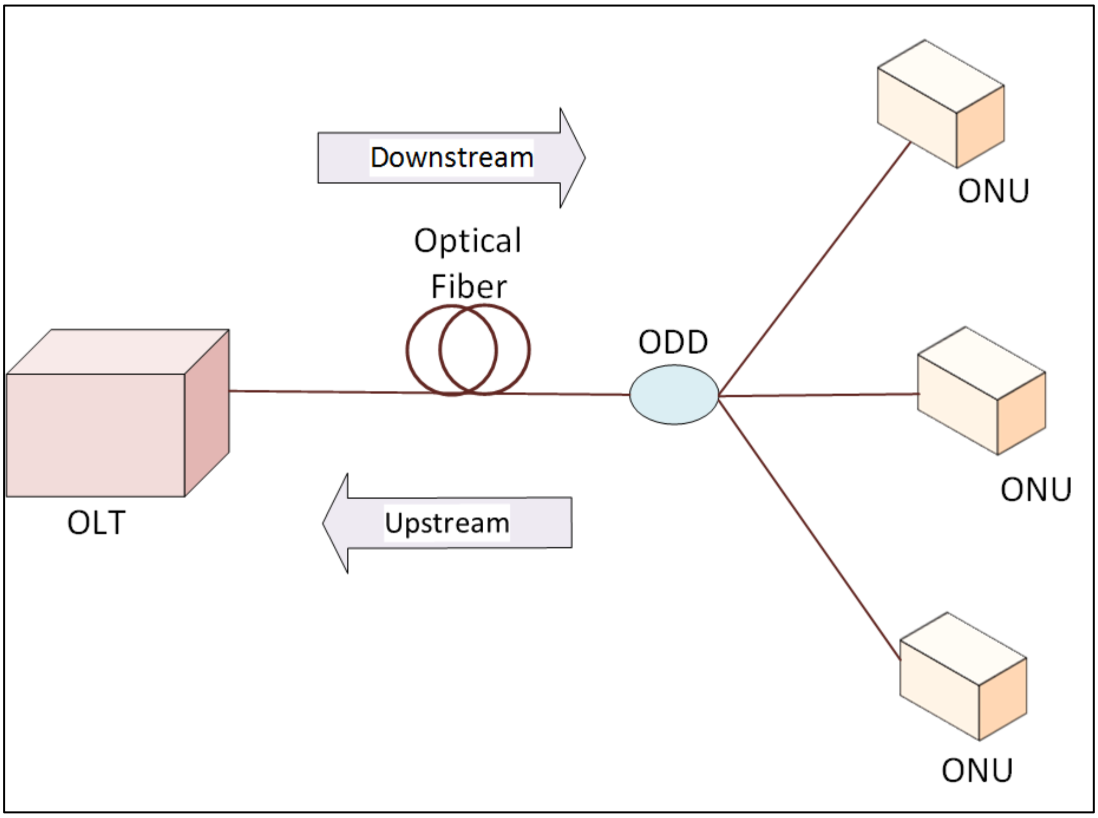

Figure 2.5 The general architecture of PON network [42]

2.9.1 Multiplexing techniques in PON

According to the multiplexing techniques used in PON, there are three basic categories: Time Division Multiplexing PON (TDM-PON), Wavelength Division Multiplexing PON (WDM-PON) and Orthogonal Frequency Division Multiplexing PON (OFDM-PON).

2.9.1.1 TDM-PON

Is the most popular PON access network technology because of its cost efficiency and feasibility [55, 56]. In TDM-PON, the available bandwidth is shared by all ONUs. There two wavelengths used, one represents the downlink wavelength from OLT to ONUs while the other one represents the uplink wavelength from ONUs to OLT [42]. In TDM-PON, time slots are assigned to ONUs for receiving and transmitting their data [55] as illustrated by Figure 2.6.

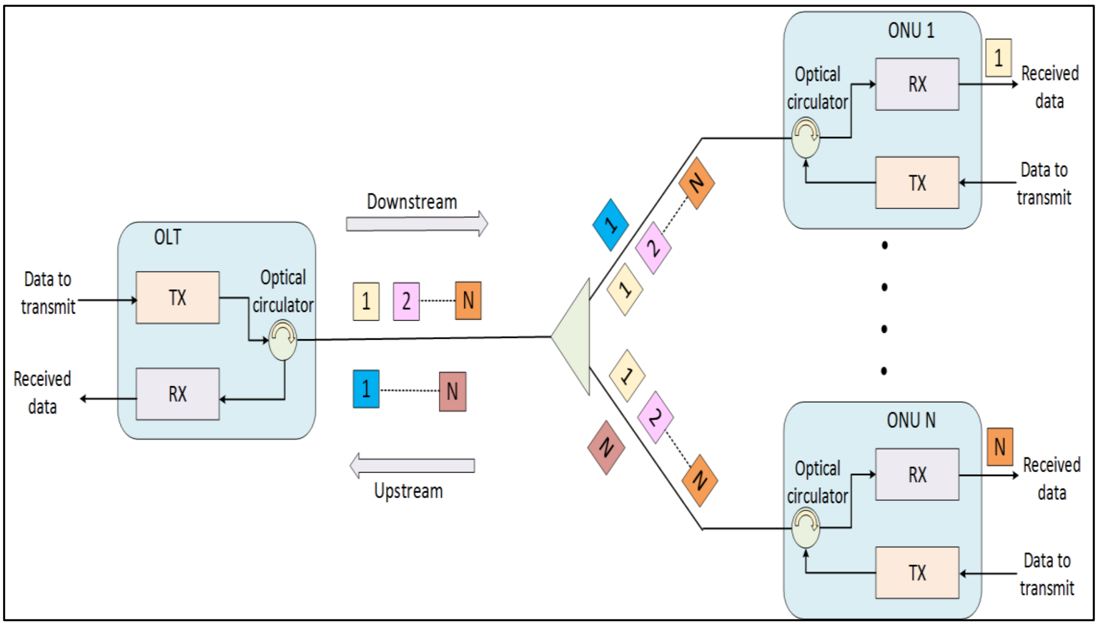

Figure 2.6 TDM-PON architecture [42]

2.9.1.2 WDM-PON

In WDM-PON, there is dedicated bandwidth for each ONU as multiple wavelengths can be supported by one optical fiber. Therefore, each user can take full advantage of the dedicated bandwidth [42, 54] as illustrated by Figure 2.7. This means a virtual point to point connection is set up between the OLT and each ONU. This point-to-point connection provided by WDM-PON leads to many advantages such as the simplicity of implementing the MAC layer control as it does not need to use media access controllers that are required by point-to-multi-point (P2MP) PON networks. In addition, high security can be guaranteed in such communication technique [49]. However, WDM-PON is considered a costly system because the assigned bandwidth for each user requires dedicated components [42].

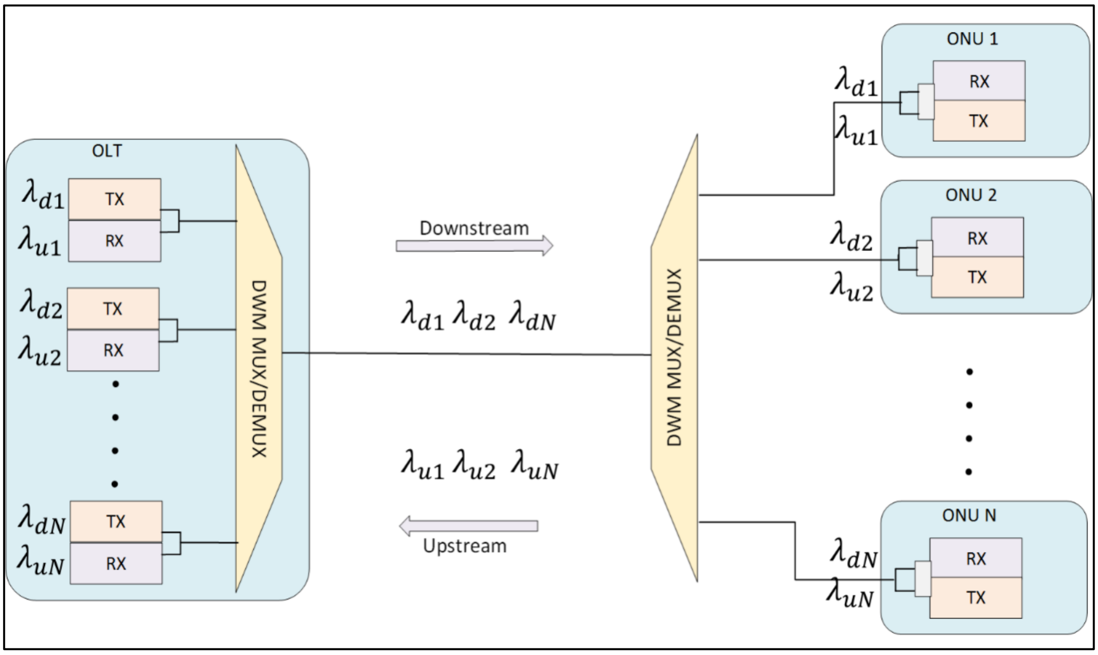

Figure 2.7 WDM-PON architecture [42]

2.9.1.3 OFDM-PON

OFDM-PON are considered as the most interesting systems by researchers due to their features that meet future requirements of the next generation PON [57, 58]. Like the conventional PON, the architecture of OFDM-PON has two wavelengths, one for uplink and the other for the downlink where the available bandwidth is shared by the ONUs at the end user [54] as shown in Figure 2.8. So, OFDM-PON can be considered as (P2MP) system [42]. Using OFDM modulation provides many benefits to the system such as cost reduction, high spectral resources exploitation and dynamic bandwidth allocation [54, 57-58]. However, the main disadvantages that can be caused by using OFDM are requirements of complex receivers and the impact of noise and frequency offset as well as the possible high peak to average ratio of OFDM signals [42, 54].

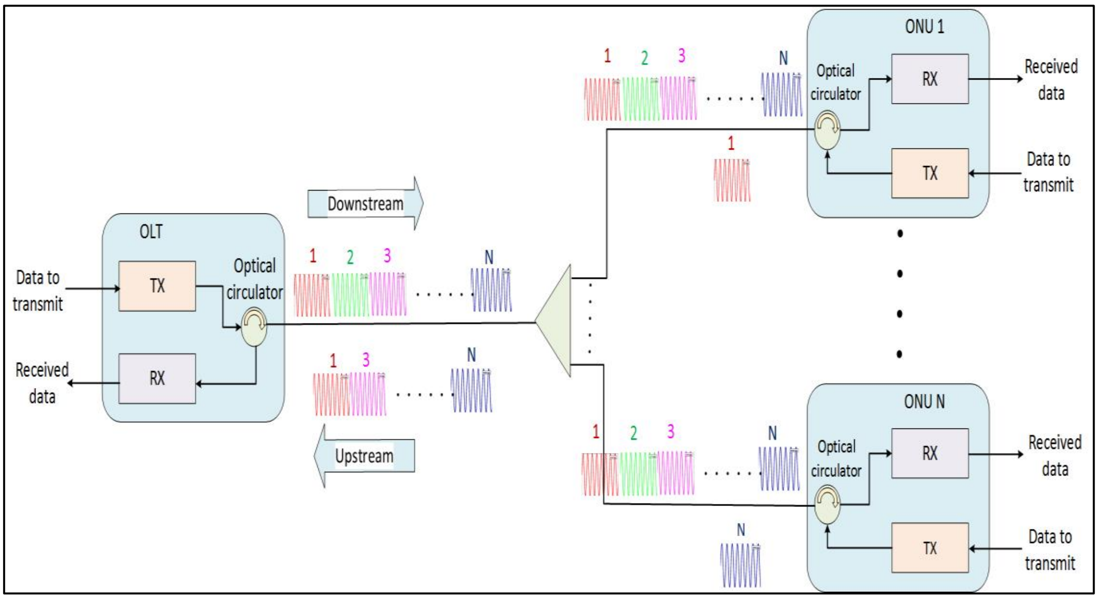

Figure 2.8 OFDM-PON architecture [42]

2.9.2 Standards

Access networks might not succeed or evolve without standards. PON standards are formulated by two standardisation groups. The first group is represented by the collaboration between the Full-Service Access Network (FSAN) group and the International Telecommunication Union-Telecommunications (ITU-T) and the second group is represented by the Institute of Electrical and Electronics Engineers (IEEE) [59]. The main PON standards formed by FSAN and IEEE are as follows:

2.9.2.1 Broadband PON (BPON)

Broadband Passive Optical Network (BPON is an extended version of the first standard Asynchronous Transfer Mode Passive Optical Network (ATM PON) (APON) that was specified by FSAN, resulting in publishing the ITU-T G.983 series recommendations [60]. The provided downstream traffic in BPON is limited to 622 Mbps and it is achieved via a TDM multiplexing system while the upstream traffic is limited to 155 Mbps and it is achieved via Time Division Multiple Access (TDMA) [61, 62].

2.9.2.2 Gigabit PON (GPON)

Gigabit Passive Optical Network (GPON) is the enhanced version of BPON introduced to handle the bandwidth and protocol limitations. Its requirements are defined by FSAN which are specified through ITU-T G.984 series recommendations [63]. GPON provides 2.5 Gbps for downstream operation and provides 1.25 Gbps for upstream operation [64]. The data transport in GPON is supported by Ethernet, TDM and ATM by using Gigabit Passive optical Network Encapsulation Method (GEM) [63].

2.9.2.3 Ethernet Passive Optical Network (EPON)

Ethernet PON (EPON) is formulated by IEEE 803.2ah standard [65]. The deployment of EPON is based on Ethernet and IP technologies [66]. EPON provides 1.25 Gbps for both downstream and upstream traffic [67]. The multiplexing schemes used in both uplink and downlink are TDM and TDMA [49, 68].

2.9.2.4 Next Generation-PON (NG-PON)

Both of IEEE and FSAN_ITU working groups work on standardisation of NG-PON [69]. The IEEE NG-PON is specified within 802.3av for achieving 10G-EPON. The target of 10G-EPON is to support two upstream data rates of 1 Gbps and 10 Gbps and downstream data rate of 10 Gbps [70]. On the other side, FSAN and ITU-T investigate the evolution of GPON by considering two phases: one for mid-term which is represented by NG-PON1 and the other for the long-term which is represented by NG-PON2 [71]. In turn, NG-PON1 has two architectures: the first architecture is XG-PON1 (X means here ten in Latin). Its aim is to have 10 Gbps as downstream data rate and 2.5 Gbps as upstream data rate while the second architecture is called XG-PON2. The target of this architecture is to have 10 Gbps for both downstream and upstream architectures [71]. The main feature of these two architectures is the compatibility with the GPON standards and the consideration of the same ODN [71]. In contrast, NG-PON2 may not be compatible with GPON standards [72]. However, the target of NG-PON2 is to achieve higher data rates up to 40 Gbps [71].

2.10 Peer- to -Peer Networks

In computer networks, sharing the resources could be into two main categories: centralised and distributed computations [73]. P2P networks are based on distributed computation in which resources are shared by a number of autonomous, heterogeneous, and distributed peers where the participants share part of their resources with other peers [74]. In such a distributed paradigm, the downloading peer instantaneously shares its downloaded files with other peers in the network [75]. Underneath the P2P umbrella, a number of applications can be highlighted such as file sharing, distributed computation, and collaboration [73]. One of the main challenges in P2P is security; where the distributed resources sharing without central control results in a more vulnerable architecture compared to centralised resources sharing [74]. Another important challenge is reliability of the P2P systems, how such systems tolerate faults, and the resilience degree. In addition, flexibility of the P2P systems is also one of the challenges based on how smoothly the peers join and leave the network [74]. Accordingly, many architectures were implemented in the past few years to tackle the challenges in P2P networks which could be classified according to two main concepts: the degree of centralisation and the network structure [73].

2.10.1 Degree of Centralisation

P2P architectures can be categorised according to how many servers are used in the network to facilitate the peers' interaction [76].

2.10.1.1 Purely Decentralised Architectures

In purely decentralised architectures, all the peers can act as a server and client at the same time without any central management which results from the equivalent capabilities of all peers in the network. The term (servents) is used to refer to the nodes in such networks as a shortcut for servers and clients words. Such architectures are not vulnerable to a single point of failure. However, the limited knowledge of each peer results in making the search mechanism for the requested resources more complex which in sequence impacts the network scalability [77, 78].

2.10.1.2 Partially Centralised Architectures

These architectures have similar basis as purely decentralised architectures with the distinction of considering supernodes in the network. Supernodes represent some of the network nodes with sufficient resources to play a more important role in the network than others. As the supernodes are dynamically selected in the network, they are immune to the single point failure. Moreover, if one of the supernodes fails for any reason, they will be replaced by another candidate node [74].

2.10.1.3 Hybrid Decentralised architectures

In hybrid decentralised architectures, copies of metadata about all the peers in the network are stored by a central server. The role of the central server in this type of architectures is to help along the (end-to-end) communication and data exchange between two peers clients by providing the location of shared files by the active peers according to the queries received from other peers to the central server. Obviously, using the central server results in reduction both of searching time for the data and the traffic between the network nodes. However, such systems are exposed to the single point of failure [74, 79].

2.10.2 Network Structure

The structure of the P2P networks can be classified according to the logical topology of these networks and how the required data can be located in the network. P2P networks can be categorised into structured and unstructured networks [78].

2.10.2.1 Unstructured Networks

Searching for a file in this category of networks is based on a random mechanism since there is no mapping between the location of the required file and its identifier. The unlinked relation between the location of the file and its identifier results in a lack of the information related to which nodes have the desired files in the network. The simple implementation is the basic benefit that can be gained by using the unstructured networks. In addition, such networks are unscalable as the resources-searching queries grow dramatically with the increase in the number of peers in the network. In addition, there is a high probability of long response time because of searching the entire network [80, 81].

2.10.2.2 Structured Networks

In contrast to unstructured networks, the structured networks have a mapping between the desired data to provide distributed indexing in the form of what is called Distributed Hash Tables (DHTs). Providing such distributed routing tables by the structured network tackles the challenges that are faced by the unstructured networks such as reducing the time required for finding the requested data by routing the queries through the network to the right destination. They also address the scalability issue [80, 81].

2.11 Linear Programming

The first step towards linear programming was taken by J.B.J. Fourier in 1827. He made this step by publishing a method that solves linear inequalities systems [82]. 1940 witnessed the start of using the word “Programming”. It is used to indicate the scheduling of activities in large organisations. The level of every activity was represented as a variable whose value should be computed and specified. These variables are involved in a set of mathematical equations or inequalities. These equations represented the restrictions in the aforementioned scheduling process and are called the constraints. The presence of these constraints was not enough to find the optimal solutions to determine the variables’ values. So, in order to find the optimal values, the designers started using an objective function in addition to using the constraints. The objective function can be defined as a function of variables with a specific purpose such as minimisation of the costs or maximisation of the profits [83].

Actually, the linear programming term comes from the linearity of the functions that represent the objectives and the constraints in any such programme [83]. The first effective algorithm for solving linear programmes was invented by George Dantzig in 1947. This algorithm was called the simplex algorithm [84]. This invention coincided with appearance of the computer. This concurrency resulted in the possibility of computerising the linear problems and this was the real reason behind the rapid development of mathematical optimisation methods and their significant use in our life [82, 83].

2.11.1 General Format of Linear program

Any linear program is a combination of basic parts. These parts represent the algebraic model of this program. All models start with a declaration or definition statements part. This declaration part defines sets, parameters and variables that are used in the mathematical equations and inequalities of the model.

Sets can be defined as arbitrary groups of objects that are used by the model. In addition, parameters can be described as the constants in the mathematical equations and inequalities. Parameters can be specified as binary value, integer value or symbolic and even their value can be constrained for example:

Param d > 0 integer;

which means d is a positive integer. Finally, variables are similar to parameters but their values are not fixed. The values of variables should be specified through optimisation [85].

The second part of any linear program is the linear function of the aforementioned parameters and variables that should be optimised [86]. This linear function is called the objective function. Regarding the optimisation purpose, the objective function usually starts with **minimise** or **maximise** expressions. The optimisation of the objective function is restricted to other linear equalities or inequalities functions that consist of the sets, parameters and variables of the program. These restriction functions are called constraints. These constraints usually start with (subject to) expression. Satisfying all these constraints by a point (set of variables) makes this point a feasible point. The whole set of these points is called the feasible region [83, 85].

Here is an example of the general form of a linear programme using AMPL[Ref]:

Set I;

Set J;

Param c{i in I}; # param means it is a parameter;

Param a{i in I, j in J};

Param b{j in J};

Var x{i in I} > 0; # var means it is a variable;

Maximise obj : sum {i in I} c[i] * x[i]; # objective function with name of obj;

Subject to cons1{j in J}: sum { i in I} a[i,j] * x[i] <= b[j]; # constraint with name of cons1;

2.11.2 Benefits of Modelling and MILP:

Modelling with MILP has many special benefits [84]:

- Modelling with MILP calculates the possibilities easier than simulation and building systems.

- Specifying design problems can be captured by MILP in a very concise way compared with verbal descriptions. MILP can further be used to solve these problems.

Modelling a design problem with MILP gives a deep perception of the problem itself with a possibility of showing some pitfalls in the design. In fact, the optimal solution of the problem may show an unaccepted result. This result can urge the modeller to make a significant structural change to some parts of the model itself.

2.11.3 Network Modelling and Optimisation

There are many ways to formulate network optimisation problem in communication systems. In this thesis, node-link formulation is used to formulate the network optimisation problems as both the demands and the links in this work models are directed. For any demand, the nodes between two end nodes (source and destination) are considered as intermediate nodes. The traffic flow in the network is subjected to flow conservation. According to the flow conservation, if the total traffic flows at the incoming link for a node is equal to the total traffic flows at the outgoing link; then the node is considered as an intermediate node. If the outgoing traffic of a node is equal to the demand then the node is considered as a source node and it is considered as a destination node when the incoming traffic is equal to the demand [85].

To illustrate network modelling and optimisation using node-link formulation, consider the three nodes network problem in Figure 2.9. The network demand $\hat{h}_{12}$ is assumed to be sourced by node 1 and terminated at node 2. The network demand will be split at node 1 into two non-negative variables: $\tilde{x}_{13,12}$ and $\tilde{x}_{12,12}$. The part before the comma in the variable subscript represents the link between two nodes while the second part after the comma represents the demand. For instance, “13” in the variable $\tilde{x}_{13,12}$ subscript represents the link between nodes 1 and 3, while “12” represent the demand from node 1 to 2. If the flow conservation is applied at each node, the following equation can be written for the demand from node 1 to 2 [85]:

$$
\begin{array}{cccccc}
\tilde{x}_{12,12} & +\tilde{x}_{13,12} & & = & \hat{h}_{12} \\
 & -\tilde{x}_{13,12} & +\tilde{x}_{32,12} & = & 0 \\
-\tilde{x}_{12,12} & & -\tilde{x}_{32,12} & = & -\hat{h}_{12}
\end{array}
$$

If two other demands are considered, the first one from node 1 to 3 and the second from node 2 to 3 as in figures 2.10, 2.11 respectively, the following two sets of equations can be obtained

$$
\begin{array}{cccccc}
\tilde{x}_{12,13} & +\tilde{x}_{13,13} & & = & \hat{h}_{13} \\
-\tilde{x}_{12,13} & & +\tilde{x}_{23,13} & = & 0 \\
 & -\tilde{x}_{13,13} & -\tilde{x}_{23,13} & = & -\hat{h}_{13}
\end{array}
$$

$$
\begin{array}{ccccc}
\tilde{x}_{21,23} & +\tilde{x}_{23,23} & = & \hat{h}_{23} \\
-\tilde{x}_{21,23} & +\tilde{x}_{13,23} & = & 0 \\
-\tilde{x}_{13,23} & -\tilde{x}_{23,23} & = & -\hat{h}_{23}
\end{array}
$$

Considering the links between the network nodes; the sum of all traffic flows in any link between two nodes should not exceed its capacity. For instance, the link capacity for the traffic flows from nodes 1 to 2 is expressed by the following inequality:

$$\tilde{x}_{12,12} + \tilde{x}_{12,13} \leq \tilde{c}_{12}$$

where $\tilde{c}_{12}$ is the link capacity for traffic flows from node 1 to node 2.

And the link capacity constraint from node 2 to node 3 is expressed as following:

$$\tilde{x}_{13,12} + \tilde{x}_{13,13} + \tilde{x}_{13,23} \leq \tilde{c}_{32}$$

Now by putting all these elements together, the network modelling problem can be written as:

The objective function is to minimise:

$$F = \tilde{x}_{12,12} + \tilde{x}_{13,12} + \tilde{x}_{32,12} + \tilde{x}_{12,13} + \tilde{x}_{13,13} + \tilde{x}_{23,13} + \tilde{x}_{21,23} + \tilde{x}_{13,23} + \tilde{x}_{23,23}$$

and it is subject to

$$\begin{array}{ccccccccccc}
\tilde{x}_{12,12} & +\tilde{x}_{13,12} & & & & & & & & = & \hat{h}_{12} \\
 & -\tilde{x}_{13,12} & +\tilde{x}_{32,12} & & & & & & & = & 0 \\
-\tilde{x}_{12,12} & & -\tilde{x}_{32,12} & & & & & & & = & -\hat{h}_{12} \\
 & & & \tilde{x}_{12,13} & +\tilde{x}_{13,13} & & & & & = & \hat{h}_{13} \\
 & & & -\tilde{x}_{12,13} & & +\tilde{x}_{23,13} & & & & = & 0 \\
 & & & & -\tilde{x}_{13,13} & -\tilde{x}_{23,13} & & & & = & -\hat{h}_{13} \\
 & & & & & & \tilde{x}_{21,23} & & +\tilde{x}_{23,23} & = & \hat{h}_{23} \\
 & & & & & & -\tilde{x}_{21,23} & +\tilde{x}_{13,23} & & = & 0 \\
 & & & & & & & -\tilde{x}_{13,23} & -\tilde{x}_{23,23} & = & -\hat{h}_{23} \\
\tilde{x}_{12,12} & +\tilde{x}_{12,13} & & & & & & & & \leq & \tilde{c}_{12} \\
 & & \tilde{x}_{21,23} & & & & & & & \leq & \tilde{c}_{21} \\
 & & & \tilde{x}_{13,12} & +\tilde{x}_{13,13} & +\tilde{x}_{13,23} & & & & \leq & \tilde{c}_{13} \\
 & & & & & & \tilde{x}_{23,13} & +\tilde{x}_{23,23} & & \leq & \tilde{c}_{23} \\
 & & & & & & & & \tilde{x}_{32,12} & \leq & \tilde{c}_{32}
\end{array}$$

recall that all $\tilde{x} \geq 0$

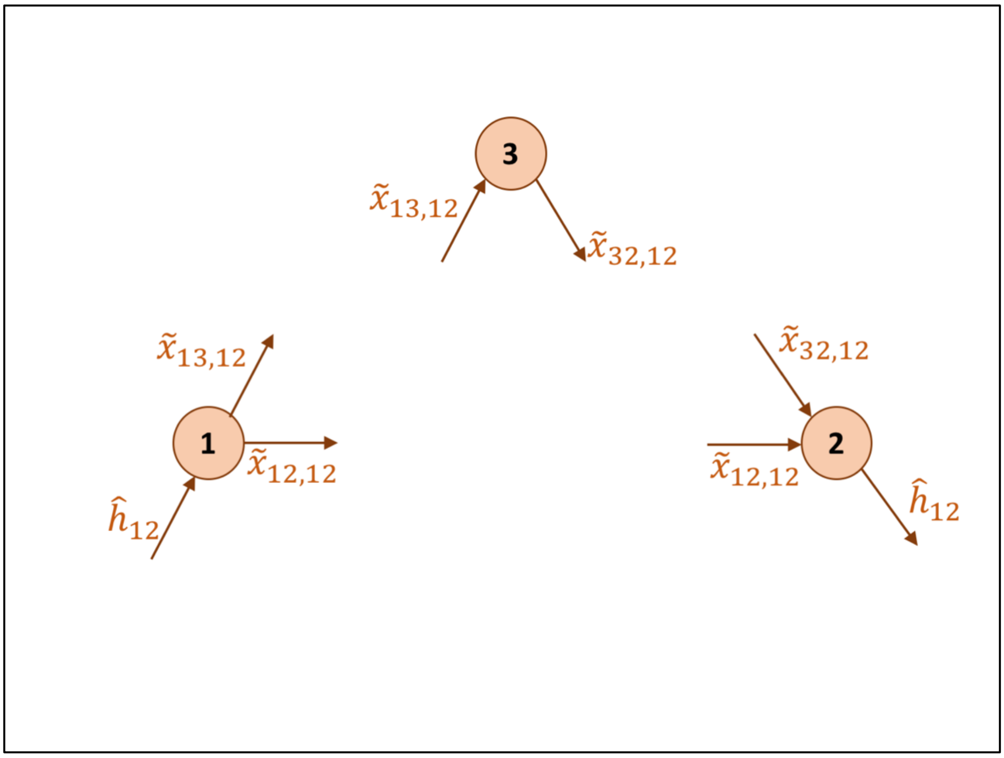

Figure 2.9 three nodes network modelling problem with traffic demand from node 1 to 2

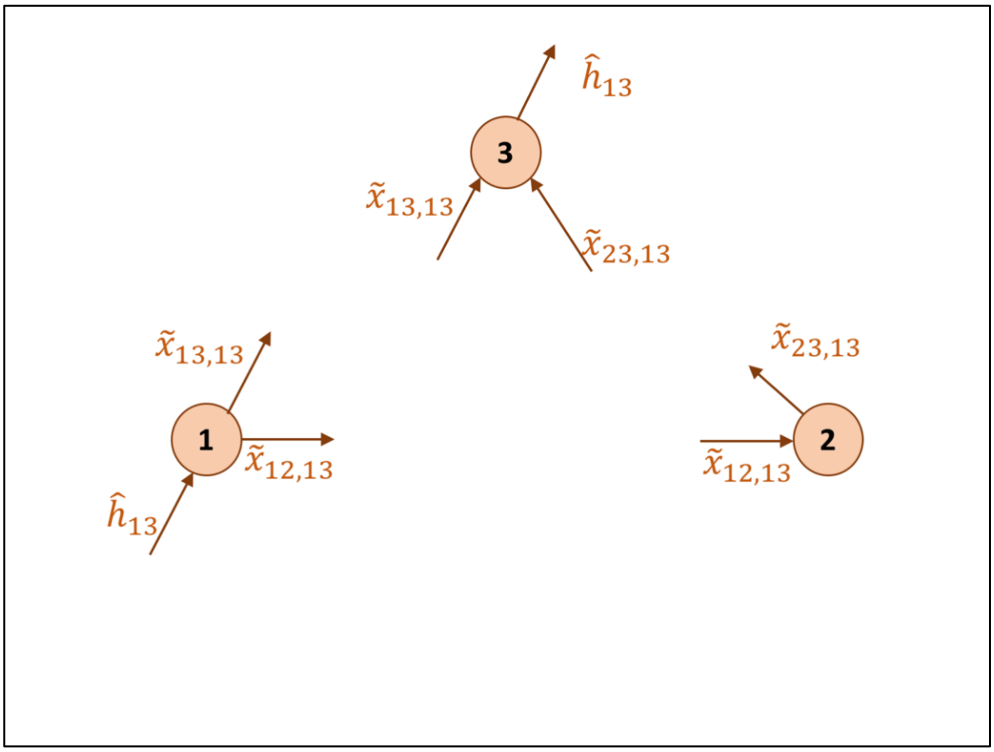

Figure 2.10 three nodes network modelling problem with traffic demand from node 1 to 3

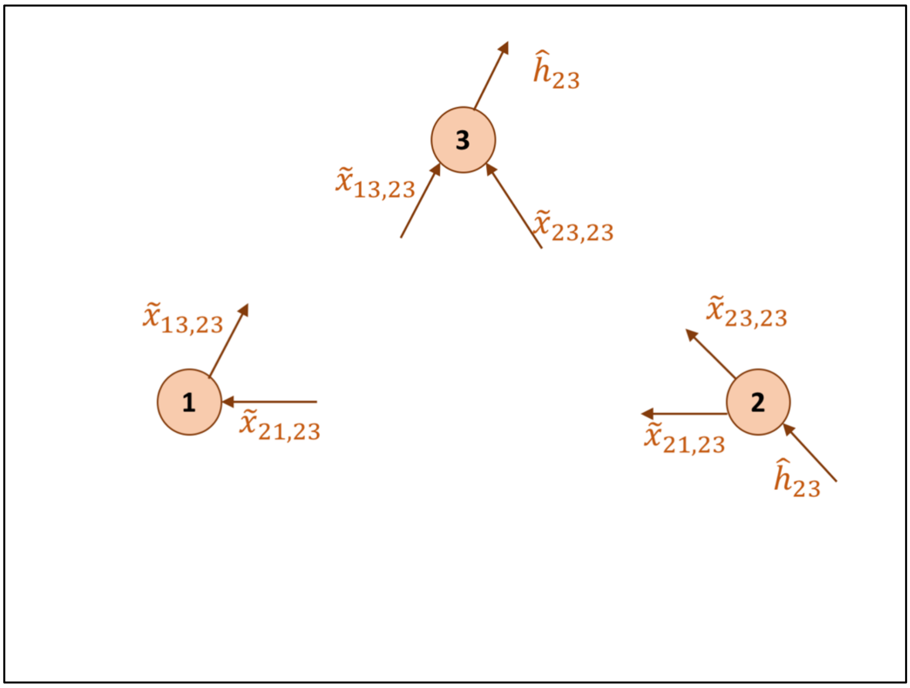

Figure 2.11 three nodes network modelling problem with traffic demand from node 2 to 3

2.11.4 Network Flows Examples

2.11.4.1 Shortest Path Model

One of the type of network models is the shortest path problem. Commonly, there is a network with two substantial nodes in addition to other nodes. The source node and destination node. In fact the goal of the optimisation in such problems is to find a route between those two nodes with minimum weight. This weight can be represented by costs, delays etc. For instance, handling the delay problem in a phone network through optimising the call routing to find the least delay path between two nodes [86].

2.11.4.2 Maximum Flow Model

This type of network models is concerned with the capacity of the links in the network but is not concerned with the cost issue in the problem calculations. In such problems, there is a limitation on the flow of each link that is represented by the maximum flow value. In fact, the major issue that is considered by this type of models is to find the bottleneck in the network in order to handle it. Furthermore, the applications that use the maximum flow models have the goal of handling a maximum number of items. For example, distributing materials through a distribution system by using distributing channels with limited capacity, for example from LA to Boston as shown in Figure 2.12 [86].

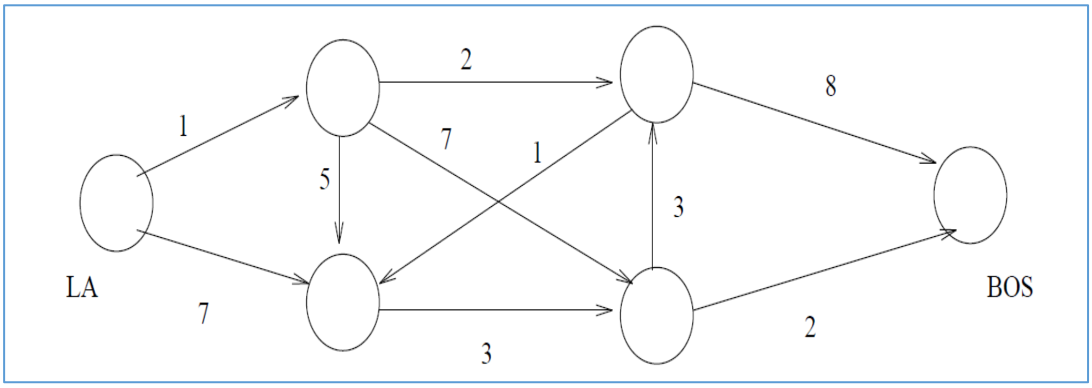

Figure 2.12 Example of distribution system of materials between LA and Boston

2.11.4.3 Transportation Model

The network model in such problems is constructed from **s** source nodes with $\mathbf{i_s}$ supplied units and **d** destination nodes with j_d demanded units. The minimisation of cost of transporting these units from the source to destination nodes is the goal of implementing this type of network models. The common assumption in this model is the supply units equal to the demand units. According to this assumption, there is always a feasible point solution for such problems [86].

2.12 Summary

This chapter has given a detailed review of IoT and the main topics that the following chapters in this thesis are based on. A description of IoT architectures has been provided with a detailed overview of IoT applications and the key challenges experienced by its deployment. A general review of cloud computing and access networks (especially PON and P2P networks) has also been presented to provide a clear understanding of the network architectures that will be presented in the coming chapters. A review of the current research efforts undertaken to enhance energy efficiency in IoT has been carried out. Finally, an overview of linear programming method was also presented.

Chapter 3 : Virtualisation Framework for Energy Efficient IoT Networks

3.1 Introduction

Emerging edge computing in IoT is essential to support the limited IoT objects' resources by providing them with high resources such as high processing and storage capabilities. The paradigm of emerging edge computing in IoT (cloudlet) opens the door to a large number of future application scenarios. However, it also faces some challenges such as security and energy efficiency. In this chapter, a design of an energy efficient edge computing platform for IoT networks is introduced and is optimised by developing a MILP model. In the developed model, the IoT network consists of four layers. The first (lowest) layer is represented by IoT objects. The networking elements (relays, coordinator and gateway) are located within the upper three layers, respectively. These networking elements aggregate and process the traffic produced by IoT devices. The processing of IoT traffic is achieved by Virtual Machines (VMs) hosted by cloudlet distributed over all the IoT network. We have optimised the total number of cloudlets, their location and the placement of VMs to reduce the total power consumption induced by traffic flow and processing. Two cases have been studied, in the first case we have considered VMs' CPU utilisation independently of the number of served IoT objects and in this case all the IoT objects can be served and satisfied. While in second case, the VMs' CPU utilisation increases with increasing the number of served IoT objects. We also consider the possibility of blocking some of the service requests in case there are not enough processing resources to handle all these requests. The power consumption has been minimised in the above cases by developing two MILP models. In both cases, we introduced two scenarios related to VMs placement. The first scenario, referred to as Gateway Placement Scenario (GPS), restricts VM hosting at the gateway element only, so that IoT data are aggregated and processed by one cloudlet at the gateway. The second scenario, referred to as Optimal Placement Scenario (OPS), allows full flexible VM placement at the relays, the coordinator and the gateway elements. Based on the MILP models principles, we have developed two heuristics (Energy Efficient Virtualisation for IoT Networks), (EEVIN1) and (EEVIN2) to mimic, in real time, the behaviour of the MILP models.

3.2 MILP Model

The MILP model considers the architecture shown in Figure 3.1. This architecture consists of four typical layers. The first / lowest layer is constructed from IoT objects. The second layer hosts the relay elements that aggregate traffic from IoT objects. The third layer hosts one coordinator element that aggregates the relay traffic. Finally, the fourth layer hosts one gateway element that aggregates the coordinator traffic. In the proposed framework, each element in the three upper layers is capable of hosting VMs that can process the traffic aggregated at that element. VMs process IoT data to extract a particular form of useful knowledge depending on the VM type, e.g. temperature gradient trends. The extracted knowledge traffic has a lower data rate compared to the original un-processed traffic. This reduced-traffic conveying knowledge is sent to the gateway at the fourth layer. The gateway provides means to connect the IoT network to the Internet. Each IoT object specialises in performing a single task only; therefore, it is assigned to a single corresponding VM type.

The MILP objective in case 1 is the minimisation of the total power consumption. In case 2 the MILP model has two objectives. In addition to the main objective of minimising the total power consumption, the model aims to maximise the number of served objects where it considers blocking in this case. The total power consumption is composed of the traffic-induced power consumption in the four layers plus the processing-induced power consumption of the VMs hosted by the cloudlets located in the networking elements at the upper three layers.

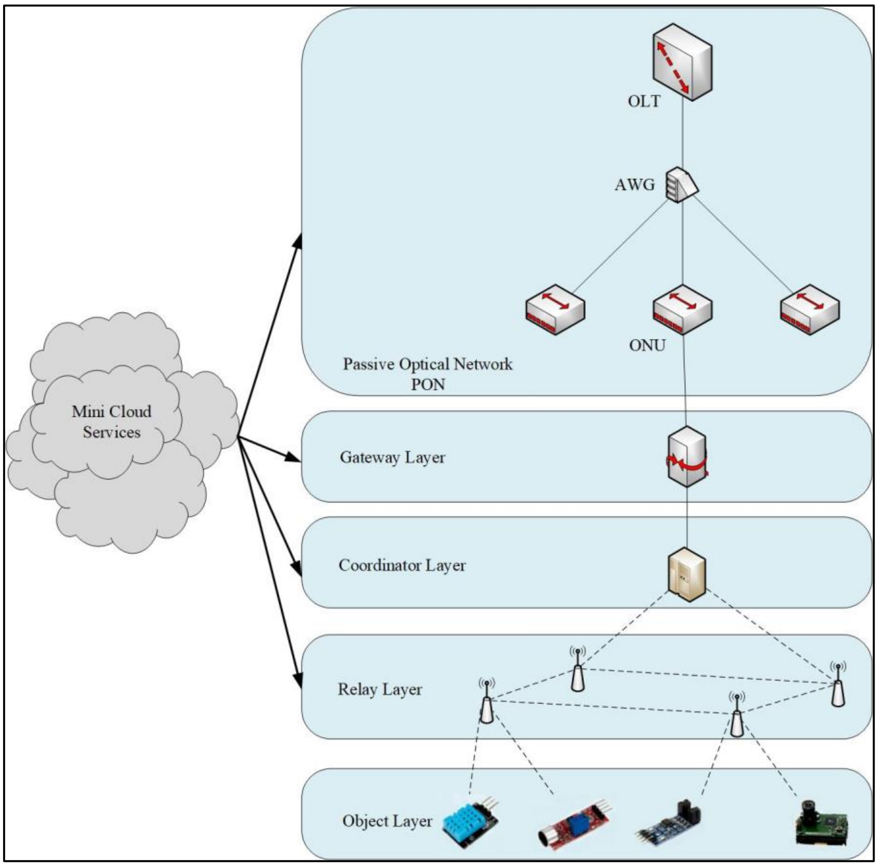

Figure 3.1 The architecture considered

The MILP is subject to certain constraints that control the placement of the VMs, their capacity in terms of number of served IoT objects, locations of cloudlets and flow conservation for the IoT original and reduced traffic. In the case of capacitated VMs, the model optimises the number of replicas of each VM. Capacitated VMs can serve a limited number of IoT objects.

To provide more clarity in the MILP expressions and notations, we have used superscripts to index the type of variables and parameters while we have used the subscripts as indices of these variables and parameters. Table 3-1 defines the parameters used in the MILP model:

Table 3-1 List of parameters and their definitions.

Notation	Description
O	Set of IoT objects
R	Set of relays
C	Set of coordinators
G	Set of gateways
TN	Set of all IoT network nodes ($TN = O \cup R \cup C \cup G$)
N_x	Set of neighbours of node x ($N_x, x \in TN$)
CN	Set of networking elements candidates for the cloudlets placement ($CN = R \cup C \cup G$)
VM	Set of virtual machines types
λ_{ov}^{upt}	Un-processed traffic from the IoT object o to the virtual machine v, in kbps
d_{xy}	Distance between the node pair (x,y) in the IoT network, in meters
ϵ	Transmission amplifier power coefficient, in joule/(bit.m^2)
E^{ot}	IoT object energy per bit for transmission, in joule/bit
E^{rt}	Relay energy per bit for transmission, in joule/bit
E^{rr}	Relay energy per bit for receiving, in joule/bit
E^{ct}	Coordinator energy per bit for transmission, in joule/bit
E^{cr}	Coordinator energy per bit for receiving, in joule/bit
E^{gr}	Gateway energy per bit for receiving, in joule/bit

W_{vc}	Normalised workload of the virtual machine v in cloudlet c
RMP	Maximum processing power consumption of relay elements (Watt)
CMP	Maximum processing power consumption of coordinator elements (Watt)
GMP	Maximum processing power consumption of gateway elements (Watt)
LM	Maximum number of IoT objects served by virtual machine v
γ	Large number
β	Large number, in bps
F	Traffic reduction factor
A	Traffic induced power consumption scaling factor

Table 3-2 defines the variables used in the MILP model:

Table 3-2 List of variables and their definitions

Notation	**Description**
λ_{ovc}^{upt}	Un-processed traffic from the IoT object o to the virtual machine v located in the cloudlet c
λ_{oc}^{upt}	Un-processed traffic from IoT object o to the cloudlet c placed in a networking element
$\lambda_{\mathrm{ocxy}}^{\mathrm{upt}}$	Un-processed traffic from the IoT object o to the cloudlet c placed in the candidate networking element passing through the link between the nodes pair (x,y)
λ_{xy}^{upt}	Un-processed traffic between the nodes pair (x,y)
λ_{xy}^{pt}	Processed traffic between the nodes pair (x,y)
λ_{cg}^{pt}	Processed traffic from the cloudlet c placed in the candidate networking element to the gateway g

λ_{cgxy}^{pt}	Processed traffic from cloudlet c placed in the candidate networking element to the gateway g passing through the link between the nodes pair (x,y)
B_{ovc}	$B_{ovc} = 1$ if the IoT object o is served by the virtual machine v which is hosted by the cloudlet c, otherwise $B_{ovc} = 0$
I_{vc}	$I_{vc} = 1$ if the virtual machine v is placed in the cloudlet c, otherwise $I_{vc} = 0$
H_c	$H_c = 1$ if a cloudlet c is built at the candidate networking element, otherwise $H_c = 0$
TW_c	Total normalised workload of the cloudlet c built at candidate networking element
PC^{rp}	Total processing induced power consumption of the relays
PC^{cp}	Total processing induced power consumption of the coordinator
PC^{gp}	Total processing induced power consumption of the gateway
PC^{otr}	Total traffic induced power consumption of the IoT objects
PC^{rtr}	Total traffic induced power consumption of the relays
PC^{ctr}	Total traffic induced power consumption of the coordinator
PC^{gtr}	Total traffic induced power consumption of the gateway

The total IoT processing induced power consumption is composed of:

1) The processing induced power consumption of each relay:

$$PC^{rp} = TW_c \cdot RMP \qquad (3\text{-}1)$$

$$\forall c \in R$$

2) The processing induced power consumption of each coordinator:

$$PC^{cp} = TW_c \cdot CMP \qquad (3\text{-}2)$$

$$\forall c \in C$$

3) The processing induced power consumption of each gateway:

$$PC^{gp} = TW_c \cdot GMP \tag{3-3}$$

$$\forall c \in G$$

Equations (3-1), (3-2) and (3-3) evaluate the processing induced power consumption of relay, coordinator and gateway respectively by considering the maximum power of the CPU used in them and the total normalised workload utilisation of the placed cloudlet.

The total IoT traffic induced power consumption is composed of:

1) The traffic induced power consumption of each IoT object :

$$PC^{otr} = \sum_{y \in R} \lambda_{xy}^{upt} \cdot \left(E^{ot} + \epsilon \cdot d_{xy}^{2}\right) \tag{3-4}$$

$$\forall x \in O$$

2) The traffic induced power consumption of each relay:

$$PC^{rtr} = \sum_{y \in R \cup C : y \neq x} \left(\lambda_{xy}^{upt} + \lambda_{xy}^{pt}\right) \cdot \left(E^{rt} + \epsilon \cdot d_{xy}^{2}\right) + \sum_{y \in O \cup R : y \neq x} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{rr} \tag{3-5}$$

$$\forall x \in R$$

3) The traffic induced power consumption of the coordinator:

$$PC^{ctr} = \sum_{y \in G} \left(\lambda_{xy}^{upt} + \lambda_{xy}^{pt}\right) \cdot \left(E^{ct} + \epsilon \cdot d_{xy}^{2}\right) + \sum_{y \in R} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{cr} \quad \forall x \in C \tag{3-6}$$

4) The traffic induced power consumption of the gateway:

$$PC^{gtr} = \sum_{y \in C} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{gr} \quad \forall x \in G \tag{3-7}$$

The traffic induced power consumption equations consist of two basic parts, the sending part and the receiving part. Both parts are based on the radio energy dissipation equation (Friis free-space equation) used in [87].The power consumption equals bit rate times the propagation energy per bit as shown in equations (3-4)-(3-7). ϵ in these equations is a parameter that relates the transmitter amplifier energy usage (joules per bit) to the distance the signal has to travel. Therefore, ϵ has units of joule/(bit*m^2) where the power is assumed to decay in proportion to the square of the distance . Equation (3-4) represents the traffic induced power consumption of the IoT objects. This equation considers the sending traffic only because the traffic received by the IoT objects is considered in this model as signalling messages with tiny data size that can be ignored. On the other hand, equation (3-7) considers only the receiving traffic induced power consumption of the gateway as the gateway layer is the highest layer in the proposed model.

In case 1, we investigated minimising the power consumption of the proposed model assuming that all IoT objects in the proposed network are served and satisfied as the

VMs' CPU utilisation does not depend on the number of served IoT objects. The following is the main objective of case 1 and the constraints it is subjected to:

Objective: Minimise

$$\sum_{c \in R} PC^{rp} + \sum_{c \in C} PC^{cp} + \sum_{c \in G} PC^{gp} + A.\left(\sum_{x \in O} PC^{otr} + \sum_{x \in R} PC^{rtr} + \sum_{x \in C} PC^{ctr}\right) + \sum_{x \in G} PC^{gtr} \quad (3\text{-}8)$$

Equation (3-8) gives the model objective which is to minimise the total power consumption of the IoT network due to traffic aggregation and processing. The scaling factor A is introduced to ensure that the traffic induced power consumption in the networking elements is comparable to their processing induced power consumption.

Subject to:

1) IoT network un-processed traffic constraints

$$\sum_{\forall c \in CN} \lambda_{ovc}^{upt} = \lambda_{ov}^{upt} \quad (3\text{-}9)$$

$$\forall o \in O, \forall v \in VM$$

$$\lambda_{oc}^{upt} = \sum_{\forall v \in VM} \lambda_{ovc}^{upt} \quad (3\text{-}10)$$

$$\forall o \in O, \forall c \in CN$$

$$\sum_{y \in N_x} \lambda_{ocxy}^{upt} - \sum_{y \in N_x} \lambda_{ocyx}^{upt} = \begin{cases} \lambda_{oc}^{upt} & if\ x = o \\ -\lambda_{oc}^{upt} & if\ x = c \\ 0 & otherwise \end{cases} \tag{3-11}$$

$$\forall\ o\ \in O, \forall c\ \in CN, \forall x\ \in TN$$

$$\lambda_{xy}^{upt} = \sum_{c\ \in\ CN} \sum_{o\ \in\ O} \lambda_{ocxy}^{upt} \tag{3-12}$$

$$\forall\ x\ \in TN, \forall y\ \in\ N_x$$

Constraint (3-9) ensures that the total un-processed traffic flowing from the IoT object o to all the cloudlets c hosting instances of the virtual machine v equal to the un-processed traffic from the object o to the virtual machine v. Constraint (3-10) calculates the traffic flowing from IoT objects to each cloudlet hosted by the candidate networking element. It ensures that the total un-processed traffic from the IoT object o to all the virtual machines v placed in cloudlet c is equal to the un-processed traffic from the object o to cloudlet c hosted by candidate networking element. Constraint (3-11) represents the flow conservation for the un-processed traffic from the IoT object o to the cloudlet c hosted by a networking element. It ensures that the total un-processed outgoing traffic is equal to the total un-processed incoming traffic for each IoT node except for the source and the destination. Constraint (3-11) represents the total unprocessed traffic between any IoT node pair (x, y).

2) IoT network processed traffic constraints

$$\sum_{\forall g \in G: c \notin G} \lambda_{cg}^{pt} = F \cdot \sum_{\forall o \in O} \lambda_{oc}^{upt} \tag{3-13}$$

$$\forall c \in CN$$

$$\sum_{y \in N_x: y \neq O} \lambda_{cgxy}^{pt} - \sum_{y \in N_x: y \neq O} \lambda_{cgyx}^{pt} = \begin{cases} \lambda_{cg}^{pt} & if\ x = c \\ -\lambda_{cg}^{pt} & if\ x = g \\ 0 & otherwise \end{cases} \tag{3-14}$$

$$\forall c \in CN, \forall g \in G, \forall x \in TN: x \neq O: c \neq g$$

$$\lambda_{xy}^{pt} = \sum_{c \in CN} \sum_{g \in G: c \neq g} \lambda_{\mathrm{cgxy}}^{pt} \tag{3-15}$$

$$\forall x \in CN, \forall y \in N_x \cap CN$$

Constraint (3-13) calculates the reduced traffic flowing from each cloudlet c hosted by candidate networking element to the gateway g. Constraint (3-14) represents the flow conservation for the processed traffic from each cloudlet c hosted by candidate networking element to the gateway g. It ensures that the total processed outgoing traffic is equal to the total processed incoming traffic for each IoT node except for the source and the destination. Constraint (3-15) represents the total processed traffic between any IoT node pair (x,y).

3) Virtual machine placement and workload constraints

$$\sum_{o \in O} \lambda_{ovc}^{upt} \geq I_{vc} \tag{3-16}$$

$$\forall v \in VM, \forall c \in CN$$

$$\sum_{o \in O} \lambda_{ovc}^{upt} \leq \beta \cdot I_{vc} \tag{3-17}$$

$$\forall v \in VM, \forall c \in CN$$

$$\sum_{v \in VM} I_{vc} \geq H_c \tag{3-18}$$

$$\forall c \in CN$$

$$\sum_{v \in VM} I_{vc} \leq \gamma . H_c \tag{3-19}$$

$$\forall c \in CN$$

$$TW_c = \sum_{v \in VM} W_{vc} \cdot I_{vc} \tag{3-20}$$

$$\forall c \in CN$$

$$TW_c \leq 1 \tag{3-21}$$

$$\forall c \in CN$$

Constraints (3-16) and (3-17) place the virtual machine v in cloudlet c if cloudlet c is serving some IoT objects requests for this virtual machine by setting the binary indicator $I_{vc} = 1$. β is a large enough number with units of bps to ensure that I_{vc}= 1 when $\sum_{o\,\in O} \lambda_{ovc}^{upt}$ is greater than zero, otherwise I_{vc}= 0. Constraints (3-18) and (3-19) build a cloudlet c in the candidate networking element if this networking element is chosen to host at least one virtual machine v, where γ is a large enough unitless number to ensure that H_c= 1 if $\sum_{v\,\in VM} I_{vc}$ is greater than zero, otherwise H_c= 0. Constraint (3-20) calculates the total normalised workload of each built cloudlet c. The total workload of each cloudlet in case 1 of this model depends on the number of VMs placed in this cloudlet but it does not depend on the number of served IoT objects. Constraint (3-21) ensures that the total normalised workload of each cloudlet c does not exceed its capacity limit.

4) GPS implementation constraint

$$\sum_{c1\,\in R\cup C} H_{c1} = 0 \tag{3-22}$$

Constraint (3-22) restricts placing the cloudlets only in the gateway by preventing placing any cloudlet in the relay and coordinator layers.

5) Virtual machines capacity constraints

$$\beta \cdot \lambda_{ovc}^{upt} \geq B_{ovc} \tag{3-23}$$

$$\forall\, o\, \in O, \forall v\, \in VM, \forall c\, \in CN$$

$$\lambda_{ovc}^{upt} \leq \beta \cdot B_{ovc} \tag{3-24}$$

$$\forall\, o\, \in O, \forall v\, \in VM, \forall c\, \in CN$$

$$\sum_{o \in O} B_{ovc} \leq LM \tag{3-25}$$

$$\forall v \in VM, \forall c \in CN$$

Constraints (3-23) and (3-24) ensure that the binary indicator B_{ovc} is set to 1 if the object o is served by the virtual machine v hosted by cloudlet c, otherwise $B_{ovc} = 0$. Constraint (3-25) ensures that the number of IoT objects served by each virtual machine v placed in cloudlet c does not exceed a certain limit (LM).

In case 2, we have considered VMs with CPU utilisation that increases according to the increase in the number of IoT objects served by these VMs. So, in this case, blocking the requests of IoT objects is taken into consideration in case the processing resources (workload capacity) of all cloudlets in the network are not enough for all required VMs. There are therefore two main objectives in this case, they are: maximising the serving rate of IoT objects and minimising the power consumption as illustrated by equation (3-26). The same power calculations we used in case 1 to calculate the processing induced power consumption and the traffic induced power consumption are used in case 2. As a result of the difference between the main objectives in both cases, there are some changes in the constraints each case is subjected to. The main objective and the changes in the constraints are discussed below.

Objective: Maximise

$$\sum_{o \in O} \sum_{v \in VM} \sum_{c \in CN} S.B_{ovc}$$

$$- \left(\sum_{\forall c \in R} PC^{rp} + \sum_{\forall c \in C} PC^{cp} + \sum_{\forall c \in G} PC^{gp} \right)$$

$$- \left(A. \left(\sum_{x \in O} PC^{otr} + \sum_{x \in R} PC^{rtr} + \sum_{x \in C} PC^{ctr} \right) \right.$$

$$\left. + \sum_{x \in G} PC^{gtr} \right) \quad (3\text{-}26)$$

Equation (3-26) gives the model objective after considering blocking where the number of satisfied objects served by hosted VMs in cloudlet c is maximised while the network and processing power consumptions are minimised. The parameter S is used to scale the number of served objects so that it becomes comparable to the consumed power.

Most of the constraints that case 1 is subjected to are used in case 2 except constraints (3-20), (3-23) and (3-24). These constraints are not used any more in case 2 and we replaced them by other constraints to accomplish the main objective of the proposed model in this case.

$$B_{ovc} \leq \lambda_{ovc}^{upt} \quad (3\text{-}27)$$

$$\forall o \in O, \forall v \in VM, \forall c \in CN$$

$$\sum_{c \in CN} B_{ovc} \leq 1 \quad (3\text{-}28)$$

$$\forall o \in O, \forall v \in VM$$

In case 2, we used constraints (3-27) and (3-28) to set the value of B_{ovc} so that it can be used in the maximisation objective as it is the indicator of the IoT objects' serving rate. Constraint (3-27) ensures that the binary indicator B_{ovc} is set to 1 if there is an uploaded traffic from object o to the virtual machine v hosted by cloudlet c, otherwise $B_{ovc} = 0$. Constraint (3-28) ensures that only one copy of the required VM hosted by one cloudlet serve each IoT object.

$$TW_c = \sum_{v \in VM} \sum_{o \in O} B_{ovc} . W_{vc} \qquad (3\text{-}29)$$

$$\forall\, c \in CN$$

Instead of using constraint (3-20) in calculating the workload of each cloudlet in case 1 we used constraint (3-29) in this case to evaluate the total workload of each cloudlet depending on the number of the served IoT objects by the required VMs placed in these cloudlets.

3.3 Results of the MILP model

Figure 3.2 shows the distribution of IoT objects, relays and coordinator elements within an area of 30m×30m. The gateway, not shown in Figure 3.2, is 100m away from the coordinator. We have considered 50 IoT objects, 25 relays, one coordinator and one gateway. The IoT objects are randomly and uniformly distributed and a relay element is placed every 6m [88]. Therefore, as an area of 30m×30m is considered, this area is covered using 25 relay elements in total as shown in Fig. 3.2. Moreover, If the relays are far spaced, then the IoT nodes will consume high power in reaching the other relays. On the other hand, if the relays are closely spaced then their number increases which increases the total power consumption and the cost. We have considered the receiving and transmitting power consumption (including propagation losses and the power amplifier power consumption) for IoT objects and the networking elements [89]. Devices in the four layers communicate using the ZigBee protocol. Every element in our network consists of two basic parts represented by communication and processing parts. The specifications of the communication part used in objects, relays and coordinator are based on [90] while we used Cisco 910 industrial router [91] for the communication part of the gateway. Also, we provided the relays, coordinator and gateway elements with an Intel Atom Z510 CPU [92] to be used as the processing element. Table (3-3) shows the input parameters used in the model. We have considered a range of traffic reduction percentages after processing to investigate the impacts of different processing applications.

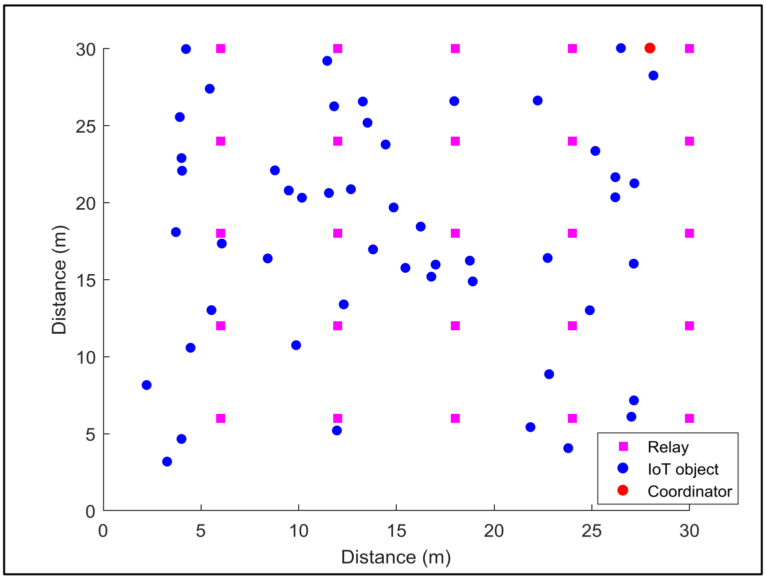

Figure 3.2 IoT deployment area

Table 3-3 List of input parameters

Parameter Name	Value
Traffic sent from IoT object to a VM type (λ_{ov}^{upt})	5kbps [93]
CPU maximum power consumption (RMP, CMP, GMP)	4.64 W[92]
Number of CPUs used in a relay, coordinator and gateway respectively	1, 2, 4
IoT object, relay and coordinator transmitting energy per bit (E^{ot}, E^{rt}, E^{ct})	50nJ/bit [90]
Relay and coordinator receiving energy per bit (E^{or}, E^{rr}, E^{cr})	50nJ/bit [90]
Gateway receiving energy per bit (E^{gr})	60µJ/bit [91]
Transmission amplifier power coefficient (ϵ)	255 pJ/(bit.m^2)[90]
VM type 1 normalised workload in relay, coordinator and gateway elements (W_{1c})	0.1, 0.05, 0.025 [94]
VM type 2 normalised workload in relay, coordinator and gateway elements (W_{2c})	0.2, 0.1, 0.05 [94]
VM type 3 normalised workload in relay, coordinator and gateway elements (W_{3c})	0.3, 0.15, 0.075 [94]
VM type 4 normalised workload in relay, coordinator and gateway elements (W_{4c})	0.4, 0.2, 0.1 [94]
Traffic reduction percentage (F)	{10, 30, 50, 70, 90}%
Distance between node pair (x, y) in the IoT network, in meters (d_{xy})	Within 30m × 30m [95]
γ, β, A, S	50, 10000000 bps, 5, 2

We have considered two scenarios. The first scenario, referred to as Gateway Placement Scenario (GPS), restricts VM hosting at the gateway element only, so that IoT data are aggregated and processed by one cloudlet at the gateway. The second scenario, referred to as Optimal Placement Scenario (OPS), allows full flexible VM placement at relays, the coordinator or the gateway elements. Both scenarios evaluate four different types of VMs. In case 1, the VMs' CPU utilisation depends on the VM type only and it is not a function of the number of served IoT objects, i.e. constant serving rate for example the application of video streaming where broadcasting the same information to multiple IoT objects does not increase the power consumption of the node in question. The VMs' CPU utilisation is also assumed to be independent of the traffic reduction percentage. This is because the same VM processing power can be assigned to tasks that can produce high traffic reduction such as temperature differential, or to image compression tasks that might not achieve large data reduction.

The total power consumption of the implemented scenarios is shown in Figure 3.3. The *x* axis represents the different traffic reduction percentages considered. Figure 3.3 divides the total power consumption into its two components: traffic-induced and processing-induced power consumption. The results show that GPS has a higher total power consumption compared to OPS. This is mainly due to the higher number of hops crossed by the IoT-to-VM traffic in the IoT network as all VMs are located in the upper fourth layer. In addition, GPS total power consumption is not affected by the different traffic reduction percentages considered in Figure 3.3. The reason is that the gateway used to host the VMs represents the last layer in traffic aggregation and processing.

Hence, the extracted knowledge is locally hosted by the gateway and not sent to the upper layers. Therefore, the traffic-induced power consumption for GPS comes only from the non-reduced traffic received from the lower layers, which is not affected by the different reduction percentages.

On the other hand, OPS manages to reduce the total power consumption compared to GPS. This is due to OPS' optimal VMs placement which reduces the number of hops between the VMs and IoT objects.

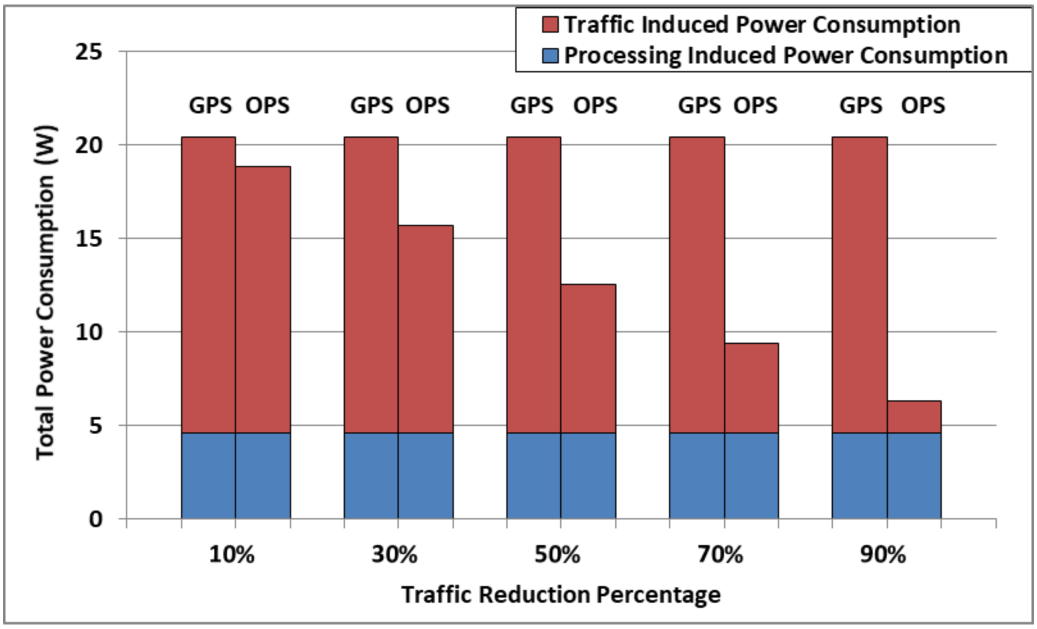

Figure 3.3 Total power consumption of GPS and OPS (case1).

The results also indicate that lower total power consumption is feasible with higher traffic reduction percentages for OPS. This is because the reduced "knowledge" traffic required a lower number of components in the IoT network elements, e.g. lower number of ports. Reducing the number of networking elements and/or their components allows them to be powered off, which achieves power efficiency. The total and network power savings for the OPS are 38% and 49%, respectively, compared to the GPS.

GPS and OPS consume the same processing-induced power, due to two reasons. First, VMs in both scenarios have similar total CPU utilisations as both scenarios serve similar input demands. Second, the VMs' power consumption is independent of the

VMs' placement as the IoT networking elements are assumed to be equipped with similar CPUs.

Both scenarios powered-on one VM copy for each VM type. This decision is influenced by the fact that the VMs considered are un-capacitated in terms of the maximum number of served IoT objects. Therefore, each VM type could serve all its objects using one VM copy only for both scenarios.

Figure 3.4 shows the total power consumption of the OPS considering capacitated VMs. We have specified the capacity of the VMs to serve 5, 10 or 15 objects.

Figure 3.4 shows that the power consumption increases with decrease in the number of objects per VM. The increase in power consumption is due to the need for more VM copies that have to be created for each VM type. This increases the CPU utilisation of the networking elements, and therefore, more power is consumed.

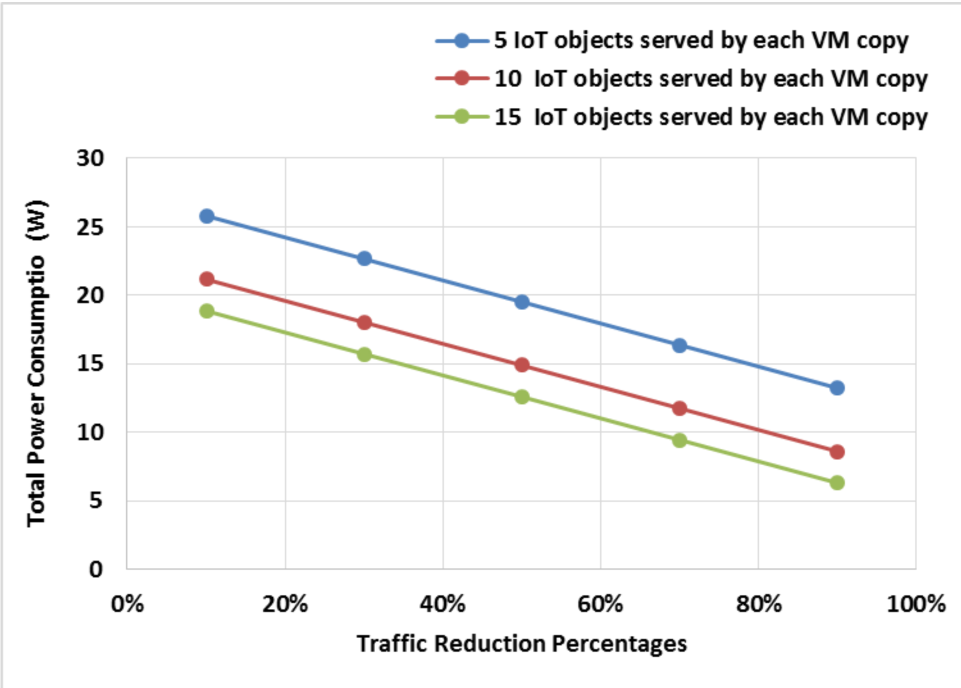

Figure 3.4 Power consumption considering capacitated VMs (case1)

In contrast with case1, case 2 MILP model shows that the total power consumption of OPS is much higher than that of GPS as illustrated by Figure 3.5. The low power consumption of GPS in this case results from considering the blocking of IoT objects' requests. Since the VMs' CPU utilisation increases with increase in the number of IoT objects, placing cloudlets in the gateway only would not be enough to handle all the IoT objects' requests of VMs. The gateway shows a blocking of 50% of the total IoT objects requests. The only VMs' types hosted by the cloudlet placed in the gateway are VM type 1 and type 2 with lower workload requirements compared with the other two types. On the other hand, all the IoT objects in OPS are served. Blocking 50% of the IoT objects' requests results in the low processing induced power consumption of the GPS compared to OPS shown in Figure 3.5.

By comparing the processing induced power consumption of OPS in both model cases (Figures (3.3) and (3.5)), it is obvious that OPS in case 2 consumes much higher power than in case 1. Since the processing induced power consumption is a function of the utilised workloads of the cloudlets as illustrated by equations (3-1) to (3-3) and since the workload of each cloudlet is the summation of utilised workloads of VMs as illustrated by equations (3-20) and (3-29) then increasing number of served IoT objects by VMs results in increase in their CPU utilisation in case 2. So, consequently this increase results in higher power consumption in terms of processing induced power.

According to the traffic induced power consumption in both cases, a different comparison between the two cases could be made. The traffic induced power consumption of GPS in both cases are the same; as the gateway layer is the highest layer and all the traffic are distained to the gateway to be processed even the blocked objects requests are sent to the gateway as it is the network gate to other networks in the internet. In addition, the traffic induced power consumption of OPS in both cases are very comparable but actually they are not equal. Most of the traffic induced power in OPS is sourced by the gateway due to its high communication power requirements. This makes the traffic induced power consumption of the other components barely visible compared with the gateway traffic induced power consumption. As it has been alluded previously, all traffic in the network is aggregated at the gateway and regardless to VMs and cloudlets distribution in the lower layers, the same amount of traffic is received by the gateway. Accordingly, the traffic induced power of the gateway is independent of the cloudlets and VMs locations; but it depends on the traffic reduction percentage.

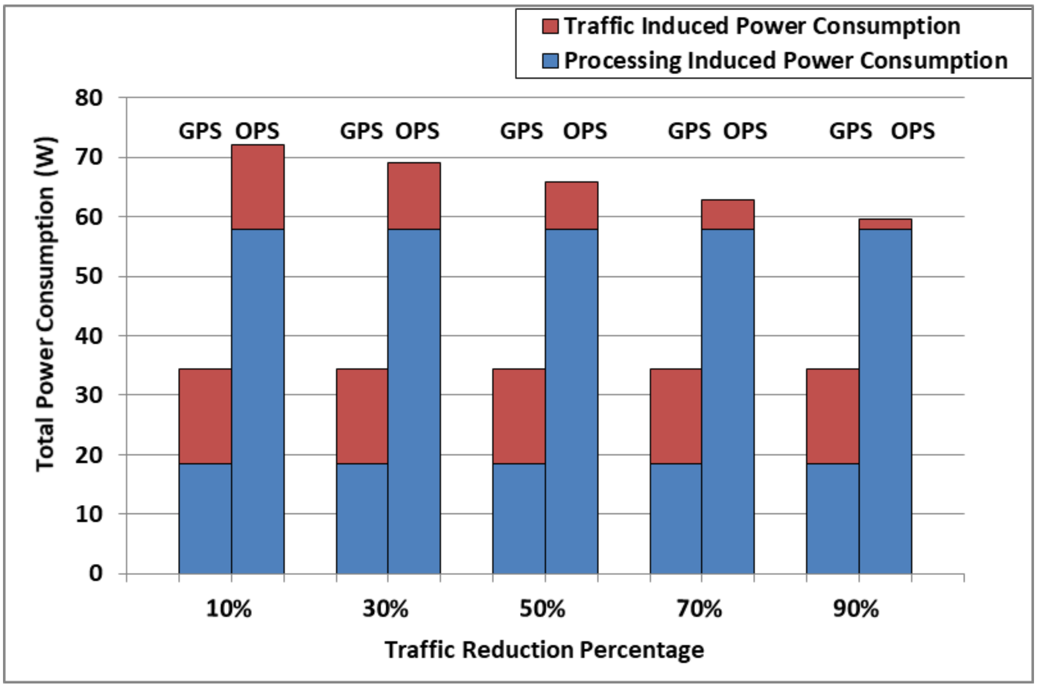

Figure 3.5 Total power consumption of GPS and OPS (case2)

To finalise the portrait of the proposed network picture, the power consumption of the other components in the lower layers is examined as follows:

First, according to the architecture of our proposed network, objects send their requests to the closest adjacent relays; as the relay layer is the aggregation layer of the object layer traffic. So, the objects produce the same amount of traffic induced power in both cases regardless of the cloudlets and VMs distribution in the network. Second, the situation of the coordinator is very similar to the gateway as it aggregates the traffic of all the network and sends it to the gateway. So the coordinator usually consumes the same amount of traffic induced power in both cases of the model as there is no impact of the lower layer on its traffic induced power consumption. However, there is a slight difference in the power consumed by the coordinator for both cases if a cloudlet is placed in the coordinator in one of the model cases. This difference results from the increase in received traffic by the coordinator as the IoT request should be served by the cloudlet placed in it. However, this difference is too small to be noticed and for that, it can be said that the coordinator consumes the same amount of traffic induced power in both cases.

The third and final traffic power comparison is related to the relay layer. Figure 3.6 shows the traffic induced power consumption of the relays in the two cases of the MILP model. The results show that the relays in case 1 consume more traffic induced power than in case 2. Since the utilisation of VMs' CPU in case 1 does not depend on the number of served IoT objects, there is no need for a large number of cloudlets to be placed in the network to handle these VMs as there is only one copy of each type of VMs needed. The cloudlets are placed in centralised locations and closer to the gateway. This results in increase in the number of hops that the unprocessed traffic passes through to reach the destined VM. On the other hand, this proximity to the gateway decreases the number of hops that the processed traffic passes through to be delivered to the gateway. While in case 2, many cloudlets are distributed through the network and are placed as close as possible to the IoT objects. This results in decreasing the number of hops that the traffic generated by IoT objects should pass through to reach the VM to be served. It however, at the same time, increases the number of hops that the processed traffic should pass through to reach the gateway. This trade-off between the number of hops that the unprocessed and processed traffic should pass through and the location and number of the cloudlets results in differences in the traffic induced power consumption of the relays in the two cases.

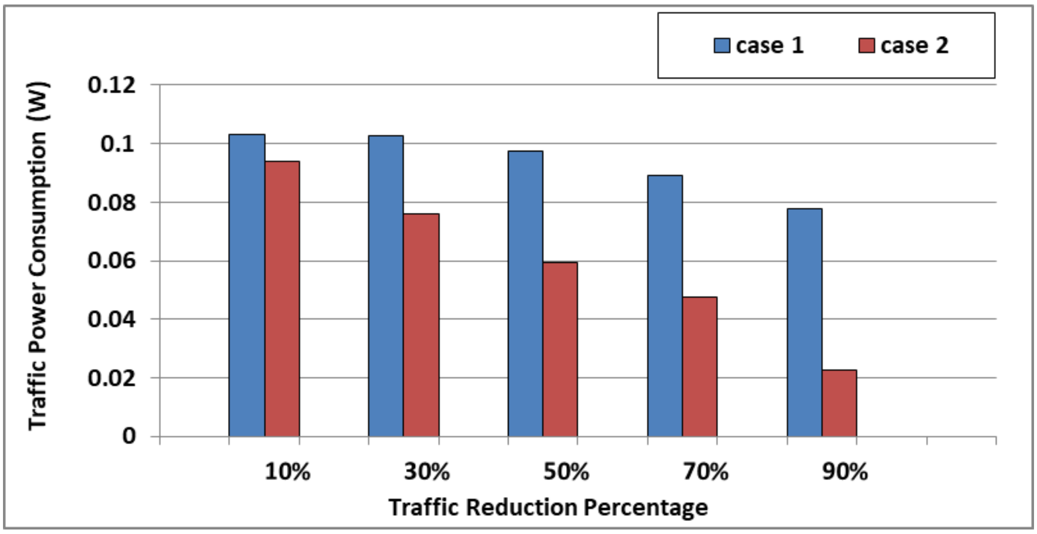

Figure 3.6 Traffic induced power consumption of the relays in cases 1 & 2

The capacitated VM in OPS-case2 with different capacities (in terms of number of served IoT objects) does not have any impact on the power consumption as shown in Figure 3.7. This results from hosting many VM copies by the distributed cloudlets in the network in case 2. The high number of VM copies results in allocating one VM copy for each IoT object. Figure 3.7 shows that different VM capacities (serving 5, 10, 15 objects) in case 2 consume the same amount of power.

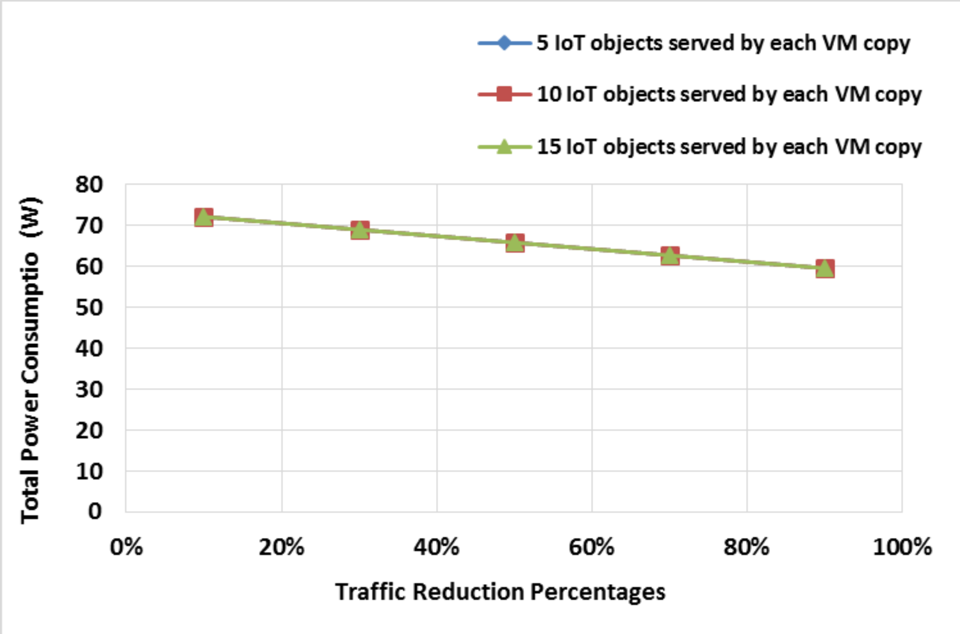

Figure 3.7 power consumption considering capacitated VMs in case 2

3.4 EEVIN Heuristic

3.4.1 Case 1 heuristic (EEVIN1)

In this section, we validated the MILP model operation by developing the EEVIN heuristic mentioned. This heuristic mimics the behaviour of the MILP model in real time. Figure 3.8 shows the pseudo code of OPS-EEVIN1 heuristic. Similar to the MILP model, restricting the cloudlets to be placed in the gateway layer only leads to implementing the GPS in the heuristic. The total power consumption of the EEVIN1 heuristic is a function of the number and the optimum placement of the cloudlets that will serve the IoT objects by the hosted VMs according to the serving constraints of each hosted VM in each cloudlet. The serving constraints can be summarised as follows:

i. The intended VM v that is requested by IoT object o should not be hosted by more than one cloudlet.

ii. There should be sufficient processor capacity in each candidate cloudlet to accommodate the required VM workload.

For each candidate cloudlet, the heuristic checks all the candidate VMs as a result of IoT object requests to be served. If all the serving constraints are met then the intended VM will be hosted in the candidate cloudlet. The heuristic packs one cloudlet with VMs until it becomes full before packing the next cloudlet. Depending on the value of the binary indicator that specifies which VMs are hosted by cloudlet c, the heuristic calculates the total workload of each cloudlet. The processing induced power consumption of all the networking elements (relays, coordinator and gateway) is calculated based on the value of the workload of the hosting cloudlet placed in these networking elements as shown in steps 11 to 25. After calculating all the processing induced power consumption of all the nodes in the three upper layers of the proposed network, the end-to-end traffic that flows through the network is calculated. The traffic passing through each node in EEVIN1 comprises of two basic parts. The first part is the traffic generated by IoT object requests of VM. This traffic is destined to the hosting cloudlet. The second part results from the reduced traffic after it is processed by VMs hosted by the cloudlet and is sent to the gateway in the proposed network (this traffic does not pass through the IoT objects as the cloudlets are placed in the three upper layers). For the first part, the heuristic tries to route the traffic between the requesting IoT object and the node hosting the serving VM by using the minimum hop algorithm in order to minimise the traffic induced power consumption of each node. The same approach is used for the second part of the traffic. It is also routed by the heuristic based on the minimum hop algorithm for the same reason.

The purpose of calculating the end to end traffic (steps 26 to 35) is to calculate the intermediate traffic between each pair of nodes. This is used in the calculating the traffic induced power consumption of each node in the proposed network (steps 36 to 46).

Finally, the EEVIN1 heuristic calculates the total power consumption TPC by summing all the processing and traffic induced power consumption of all nodes.

Inputs: **VM** $= \{\mathbf{1} \dots \boldsymbol{NVM}\}$

$\boldsymbol{CN} = \{\mathbf{1} \dots \boldsymbol{NCN}\}$

$\boldsymbol{O} = \{\mathbf{1} \dots \boldsymbol{NO}\}$

$\boldsymbol{R} = \{\mathbf{1} \dots \boldsymbol{NR}\}$

$\boldsymbol{C} = \{\mathbf{1} \dots \boldsymbol{NC}\}$

$\boldsymbol{G} = \{\mathbf{1} \dots \boldsymbol{NG}\}$

Output: No. of Served Objects

Total Power Consumption (TPC)

1. **For** each candidate cloudlet that can host a required VM $c \in CN$ **Do**
2. **For** each Virtual Machine required by an object $v \in VM$ **Do**
3. **If** $U_{ov} > 0$ **Then**
4. **If** all serving constraints are met **Then**
5. Fcv(c,v)=1
6. Calculate the workload of the hosting cloudlet $TCWc$ without considering the number of served IoT objects
7. **End If**
8. **End If**
9. **End For**

End For

For Each relay ($j \in R$) **Do**

  If the hosting cloudlet is placed in relay layer R $j \in CN$ **Do**

    Calculate PC_j^{rp}

  End If

End For

For Each coordinator ($j \in C$) **Do**

  If the hosting cloudlet is placed in coordinator layer $j \in CN$ **Do**

    Calculate PC_j^{cp}

  End If

End For

For Each gateway ($j \in G$) **Do**

  If the hosting cloudlet is placed in gateway layer $j \in CN$ **Do**

    Calculate PC_j^{gp}

  End If

End For

For each IoT object served by a cloudlet $o \in O$ **Do**

  For each hosting cloudlet $c \in CN$ **Do**

    Calculate end to end traffic that flows from each object to each cloudlet that serves this object

  End For

End For

For Each hosting cloudlet $c \in CN$ **Do**

  For Each gateway ($g \in G$) **Do**

    Calculate the end to end reduced traffic from the cloudlet to the gateway

34. **End For**

35. **End For**

36. **For** each IoT object $a \in O$ **Do**

37. Calculate PC_a^{otr}

38. **End For**

39. **For** each relay ($a \in R$)

40. Calculate PC_a^{rtr} based on minimum hop path between node pair (x,y)

41. **End For**

42. **For** each coordinator ($a \in C$)

Calculate PC_a^{ctr} based on minimum hop path between node pair (x,y)

43. **End For**

44. **For** each gateway ($a \in G$)

45. Calculate PC_a^{gtr} based on minimum hop path between node pair (x,y)

46. **End For**

47. Calculate total power consumption

$$TPC = \sum_{j \in R} PC_j^{rp} + \sum_{j \in C} PC_j^{rc} + \sum_{i \in G} PC_j^{rg} + \sum_{a \in O} PC_a^{otr} + \sum_{a \in R} PC_a^{rtr} + \sum_{a \in C} PC_a^{ctr} + \sum_{a \in G} PC_a^{gtr}$$

Figure 3.8 EEVIN1 Heuristic

3.4.2 Case 2 heuristic (EEVIN2)

In this section, the EEVIN2 heuristic associated with the MILP model- case 2 is represented in the form of the pseudo code shown in Figure 3.9. As in Section 3.3.1 the EEVIN2 heuristic in this section is also based on OPS. As in the MILP model, the total workload of each cloudlet in case 2 depends not only on the number of VM copies hosted in the cloudlet but also it is a function of the number of served IoT objects. Calculating the total power consumption and the serving constraints in EEVIN2 is based on the same concepts as in EEVIN1. Considering blocking in this case results in a third strand of traffic generated by the IoT object passing through the upper three layers reaching the gateway. This traffic is the unserved traffic by any cloudlet due to blocked. Therefore, the IoT objects send their unserved requests to the gateway as all traffic passing through the network is sent to the gateway at the end as it is the upper layer in the network. This traffic is calculated by the heuristic according to the steps 36 to 42.

Inputs: $\mathbf{VM} = \{\mathbf{1} \ldots \boldsymbol{NVM}\}$

$\boldsymbol{CN} = \{\mathbf{1} \ldots \boldsymbol{NCN}\}$

$\boldsymbol{O} = \{\mathbf{1} \ldots \boldsymbol{NO}\}$

$\boldsymbol{R} = \{\mathbf{1} \ldots \boldsymbol{NR}\}$

$\boldsymbol{C} = \{\mathbf{1} \ldots \boldsymbol{NC}\}$

$\boldsymbol{G} = \{\mathbf{1} \ldots \boldsymbol{NG}\}$

Output: No. of Served Objects

Total Power Consumption (TPC)

1. **For** each candidate cloudlet that can host a required VM $c \in CN$ **Do**
2. **For** each Virtual Machine required by an object $v \in VM$ **Do**

For each object that requires a VM $o \in O$ **Do**

If all serving constraints are met **Then**

Bcov(c,o,v)=1

Calculate the workload of the hosting cloudlet $TCWc$ considering the number of served IoT objects

End If

End For

End For

End For

For Each relay $(j \in R)$ **Do**

If the hosting cloudlet is placed in relay layer R $j \in CN$ **Do**

Calculate PC_j^{rp}

End If

End For

For Each coordinator $(j \in C)$ **Do**

If each hosting cloudlet is placed in coordinator layer C $j \in CN$ **Do**

Calculate PC_j^{cp}

End If

End For

For Each gateway $(j \in G)$ **Do**

If the hosting cloudlet is placed in gateway layer G $j \in CN$ **Do**

Calculate PC_j^{gp}

End If

End For

For each IoT object served by a cloudlet $o \in O$ **Do**

For each hosting cloudlet $c \in CN$ **Do**

Calculate end-to-end traffic that flows from each object to each cloudlet that serves this object

End For

End For

For Each hosting cloudlet $c \in CN$ **Do**

For Each gateway ($g \in G$) **Do**

Calculate the reduced traffic from the cloudlet to the gateway

End For

End For

For each IoT object $o \in O$ **Do**

For Each gateway ($g \in G$) **Do**

If Bcov(c,o,v)=0

Calculate the end to end traffic from each unserved object by any cloudlet to the gateway

End If

End For

End For

For each IoT object $a \in O$ **Do**

Calculate PC_a^{otr}

End For

For each relay ($a \in R$)

Calculate PC_a^{rtr} based on minimum hop path between node pair (x,y)

End For

49. **For** each coordinator ($a \in C$)

50. Calculate PC_a^{ctr} based on minimum hop path between node pair (x,y)

51. **End For**

52. **For** each gateway ($g \in G$)

53. Calculate PC_a^{gtr} based on minimum hop path between node pair (x,y)

54. **End For**

55. Calculate total power consumption

$$TPC = \sum_{j \in R} PC_j^{rp} + \sum_{j \in C} PC_j^{rc} + \sum_{i \in G} PC_j^{rg} + \sum_{a \in O} PC_a^{otr} + \sum_{a \in R} PC_a^{rtr} + \sum_{a \in C} PC_a^{ctr} + \sum_{a \in G} PC_a^{gtr}$$

Figure 3.9 EEVIN2 Heuristic

3.5 EEVIN heuristic results

Figures 3.10 and 3.11 show that both MILP model and EEVIN heuristic in OPS in both cases consume approximately the same amount of total power except for a small difference. This small difference is due to the impact of the network power consumption and specifically due to the traffic induced power consumed by the relays as shown in Figures 3.12 and 3.13. Figures 3.12 and 3.13 show that the relays in the heuristic consume much more power than in the MILP model in both cases which is due to bin packing which resulted in a lower number of cloudlets placed in the network. The heuristic sequentially placed the cloudlets in the upper three layers in the proposed network starting with the first candidate location which is contrary to the MILP as it optimised choosing the cloudlet location according to the distance between the cloudlets and the served IoT objects in order to reduce the network power consumption. Placing cloudlets by the heuristic in such a way made the traffic generated by the IoT objects flow through more relays to reach the destined VM type and that resulted in higher network power consumption. The average number of the placed cloudlets and their average utilisation in the EEVIN heuristic and MILP for both cases are listed in Table 3.4.

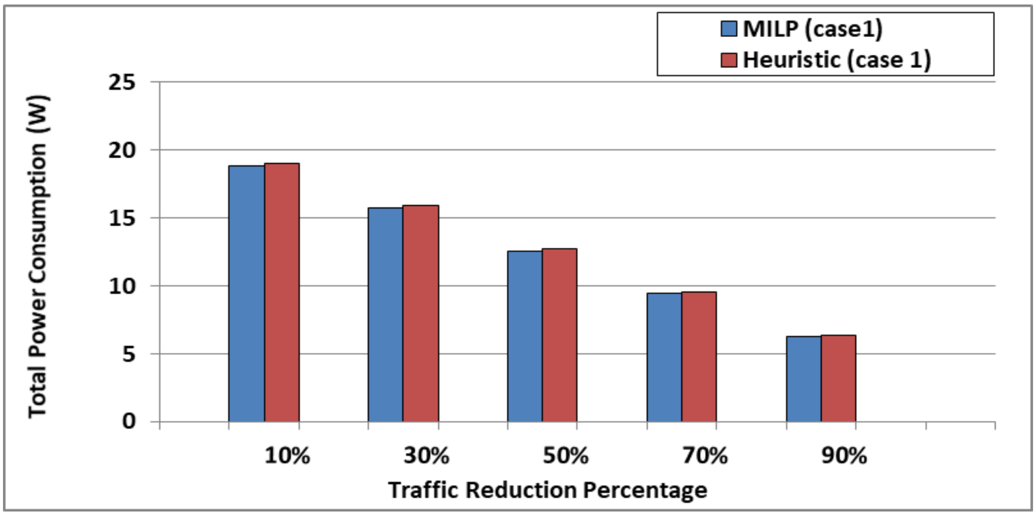

Figure 3.10 Total power consumption with MILP optimisation (case 1) and under EEVIN1 heuristic

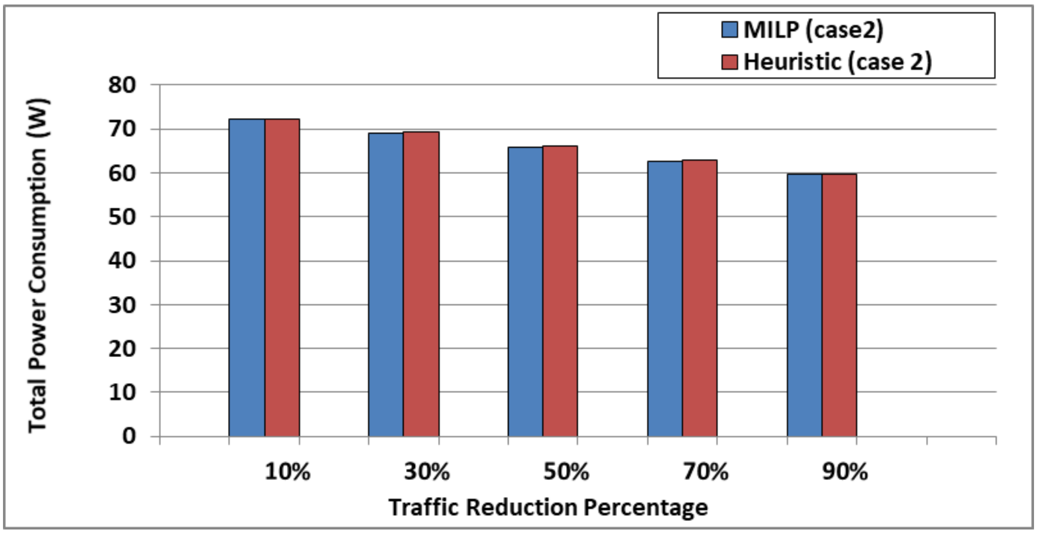

Figure 3.11 Total power consumption with MILP optimisation (case 2) and under EEVIN2 heuristic

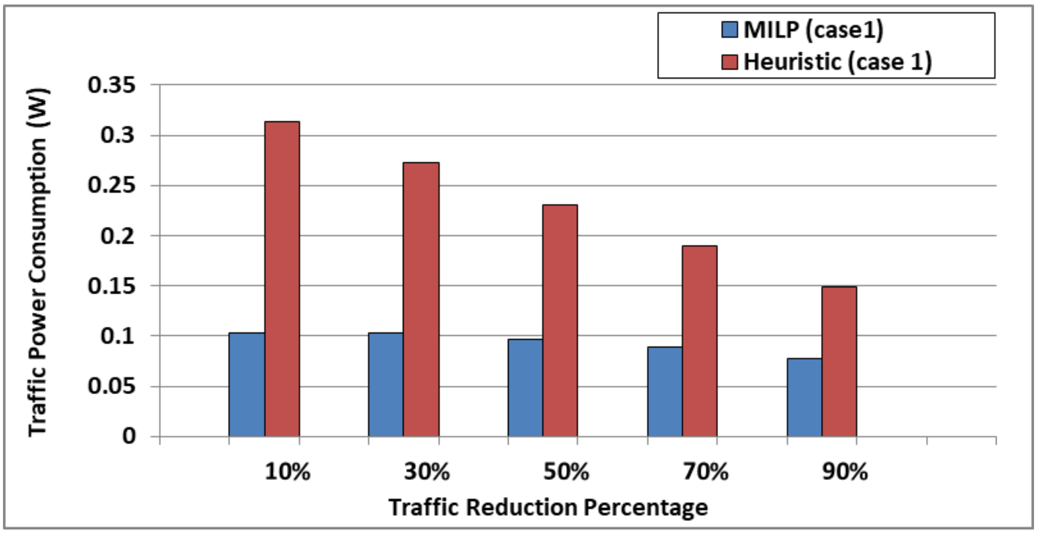

Figure 3.12 Traffic induced power consumption by relays in the MILP model (case 1) and EEVIN1 heuristic

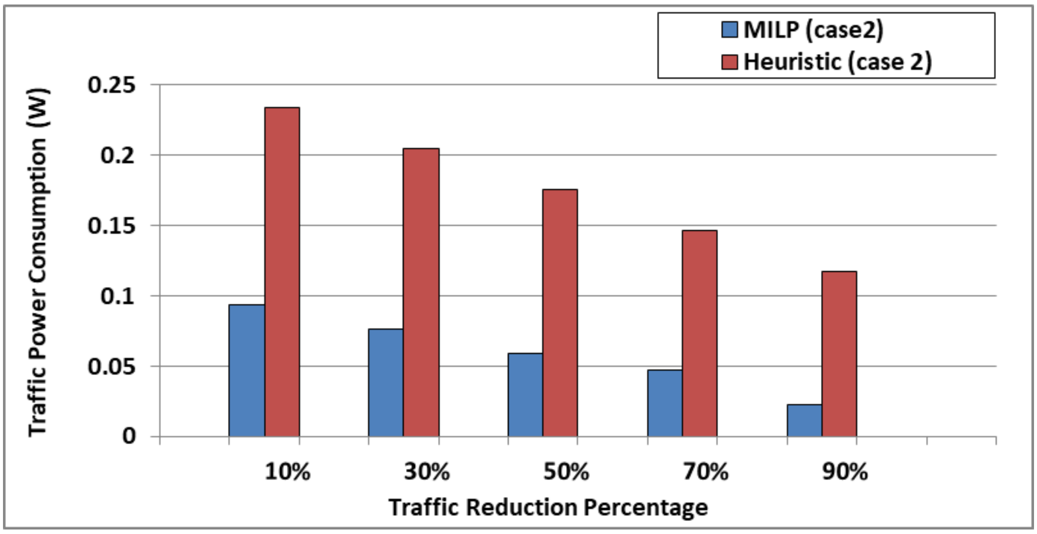

Figure 3.13 Traffic induced power consumption by relays in the MILP model (case 2) and EEVIN2 heuristic

Table 3-4 Average number of cloudlets and their average utilisation

	Case 1		**Case 2**	
	MILP	**Heuristic**	**MILP**	**Heuristic**
Average no. of the cloudlets	2	1	22	14
Average utilisation of the cloudlets	0.5	1	0.548	0.89

3.6 Summary

In this chapter, we have introduced an energy efficient edge computing platform for IoT networks. In our design, the IoT network consists of four layers. Layer one consists of IoT devices and the networking elements: Relays, coordinators and gateways, are located within the upper three layers, respectively. These networking elements perform the tasks of data aggregation and processing of the traffic produced by IoT devices. The processing of IoT traffic is handled by Virtual Machines (VMs) hosted by distributed cloudlets and located within the IoT networking elements. We have developed a MILP optimisation model, which optimises the number of cloudlets, their location and the placement of VMs with the objective of reducing the total power consumption induced by traffic aggregation and data processing.

We have also showed the impact of blocking IoT object requests if the processing resources in the network are not sufficient to handle these requests by introducing another MILP model. Here the VMs' CPU utilisation depends on the number of served IoT objects.

We investigated and compared the power consumption of two cases, developing two MILP models mentioned above: Case 1 where the MILP model minimises the total power consumption and case 2 where the MILP model maximises the number of served IoT objects in addition to the original objective of reducing the total power consumption.

Our results showed that the optimal distribution of cloudlets in the IoT network in case 1 can yield a total power savings of up to 38% compared to processing IoT data in a single cloudlet located at the gateway layer. While in case 2, locating the cloudlet at the gateway layer can yield a total power savings of up to 47% compared to the

OPS but the blocking rate reached 50% of the IoT objects' requests as the CPU utilisation capacity of the gateway was not enough to satisfy all the IoT objects. The results also revealed that the processing induced power consumption in case 2 were much higher than in case 1.

For real time implementation and based on the insights of the above MILP models, two heuristics were developed. Very comparable power savings were achieved with a small increase in the traffic induced power consumed by the relays. The heuristics are independent of the MILP optimisation and therefore provide verification for the MILP results.

Chapter 4 : Energy Efficient IoT Virtualisation with Passive Optical Access Network

4.1 Introduction

With the rapid growth in data that accompanied the explosion in the number of connected devices in IoT, serious concerns are raised about the energy cost of transporting such huge data through the Internet to be accessible by anyone anywhere. The connection between the IoT objects and the Internet is facilitated by access networks. One of the most favourable access networks in term of high bandwidth, long access distance and power consumption is passive optical networks (PON). In this chapter we design a framework for an energy efficient edge computing platform for IoT supported by a PON. The design of the proposed network is evaluated using a MILP model. The IoT network consists of four layers. The first layer represents IoT objects and the three other layers host relays, a coordinator and a gateway, respectively. The PON consists of two layers ONUs and OLT. Equipment at all layers, except the object layer, can aggregate and process the traffic generated by IoT objects. The processing is performed using distributed cloudlets that host different types of Virtual Machines (VMs). The cloudlets can be located in the three upper layers of the IoT network and the two layers of PON. We aim to reduce the total power consumption resulting from the traffic delivery and data processing at the different layers. Improvement in the energy efficiency can be achieved by optimising the location and the number of the cloudlets and VMs and by utilising energy efficient routes. We develop an Energy Efficient PON supported IoT Virtualisation (EEPIV) heuristic to enable the implementation of the MILP model concepts in real time environments.

4.2 MILP model

Our MILP model considers the architecture shown in Figure 4.1. This architecture consists of two separated IoT networks connected by a PON in order to deliver the aggregated processed traffic to the upper core network. In our framework, each IoT network is constructed from four layers. The first / lower layer is comprised of IoT objects. The second layer contains the relay elements. The objective of relays is the traffic aggregation from the IoT objects. The third layer hosts one coordinator element that aggregates traffic from the relay elements. The last layer in the IoT network consists of one gateway element. The task of the gateway is to aggregate the coordinator traffic and upload it to the access network (PON). The access network consists of two layers. The ONU layer that hosts two ONU entities and the OLT layer that hosts one OLT entity. Each ONU is connected to one of the IoT networks. ONUs aggregate and deliver IoT networks traffic to the OLT that in turn transports the traffic to the core network.

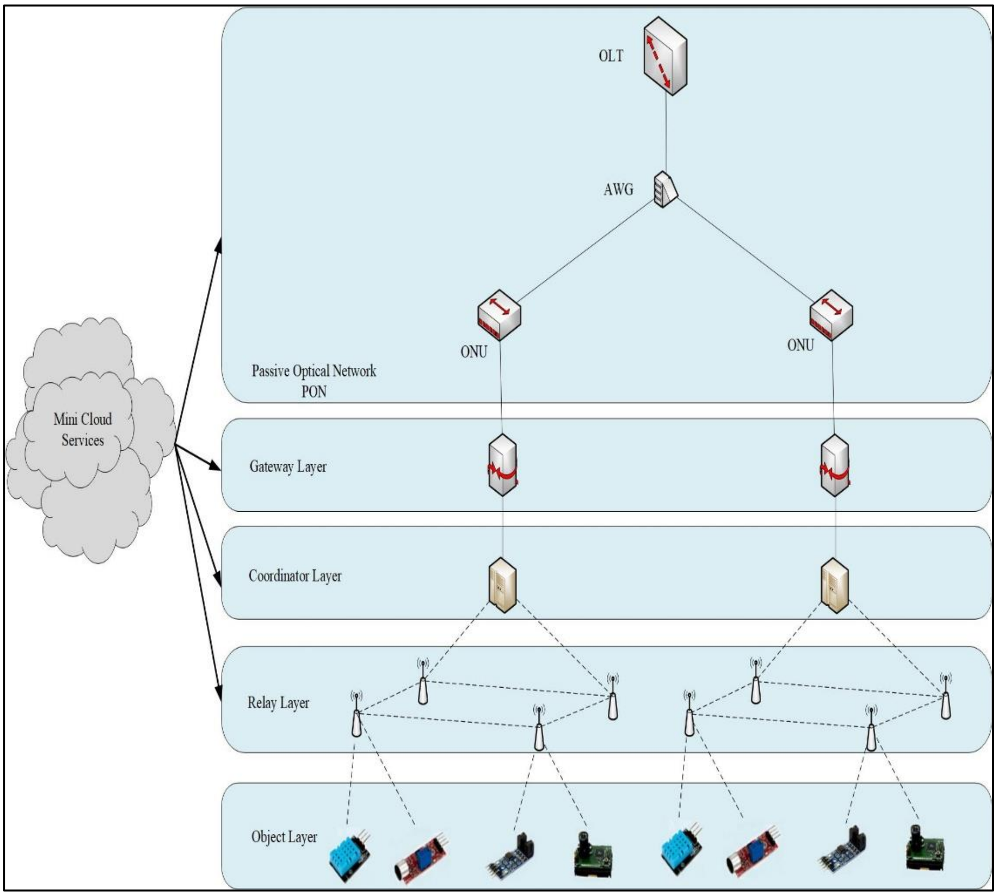

Figure 4.1 The evaluated architecture

In our framework, the capability of hosting VMs is allowed at each IoT element in the three upper layers of the IoT network in addition to the PON access network layers. Hosting VMs in IoT elements and PON entities gives these VMs the capability of processing the aggregated traffic. We modelled different VM types that correspond to different applications. Each IoT object demands one VM type. By processing the incoming raw data, VMs reduce the traffic at different percentages to generate useful information. The objective of our MILP is to minimise the total power consumption. There are two basic components of the total power consumption, the power consumption due to traffic in all IoT and PON layers and the power consumption due to VMs processing in the three upper layers of the IoT network and the two layers of the PON. The MILP power minimisation is subject to several constraints. These constraints are concerned with the optimal VMs placement, cloudlet placement, controlling traffic direction and the flow conservation for unprocessed and processed IoT traffic. For more clarity in the MILP expression and notations, we have used superscripts to index the type of variables and the parameters while we have used the subscripts as indices of these variables and parameters. Table 4.1 defines the parameters used in the MILP model:

Table 4-1List of parameters and their definitions

Notation	Description
O	Set of IoT objects
R	Set of relays
C	Set of coordinators
G	Set of gateways
ONU	Set of ONUs
OLT	Set of OLTs
TN	Set of all IoT network nodes ($TN = O \cup R \cup C \cup G \cup ONU \cup OLT$)
N_x	Set of neighbours of node x ($N_x, x \in TN$)
CN	Set of candidates nodes for the cloudlet placement ($CN = R \cup C \cup G \cup ONU \cup OLT$)
VM	Set of virtual machines types
λ_{ov}^{upt}	Un-processed traffic from the IoT object o to the virtual machine v, in kbps
d_{xy}	Distance between the node pair (x,y) in the IoT network, in meters
ϵ	Transmission amplifier power coefficient, in joule/(bit.m^2)
E^{ot}	IoT object energy per bit for transmission, in joule/bit
E^{rt}	Relay energy per bit for transmission, in joule/bit
E^{rr}	Relay energy per bit for receiving, in joule/bit
E^{ct}	Coordinator energy per bit for transmission, in joule/bit
E^{cr}	Coordinator energy per bit for receiving, in joule/bit
E^{gr}	Gateway energy per bit for receiving, in joule/bit

E^{gt}	Gateway energy per bit for transmitting, in joule/bit
E^{u}	ONU energy per bit, in joule/bit
E^{l}	OLT energy per bit, in joule/bit
W_{vc}	Normalised workload of the virtual machine v in cloudlet c
RMP	Maximum processing power consumption at relay elements
CMP	Maximum processing power consumption at coordinator elements
GMP	Maximum processing power consumption at gateway elements
UMP	Maximum processing power consumption at ONU entities
LMP	Maximum processing power consumption at OLT entities
γ, β	Large enough numbers
F	Traffic reduction factor
A	Networking elements scaling factor

Table 4.2 defines the variables in the MILP model:

Table 4-2 List of variables and their definitions

Notation	**Description**
λ_{ovc}^{upt}	Un-processed traffic from the IoT object o to the virtual machine v placed at the cloudlet c
λ_{oc}^{upt}	Un-processed traffic from IoT object o to cloudlet c placed in the candidate networking element
$\lambda_{\mathrm{ocxy}}^{\mathrm{upt}}$	Un-processed traffic from the IoT object o to cloudlet c placed in the candidate networking element passing through the link between the nodes pair (x,y)
λ_{xy}^{upt}	Un-processed traffic between the nodes pair (x,y)
λ_{xy}^{pt}	Processed traffic between the nodes pair (x,y)
λ_{cl}^{pt}	Processed traffic from cloudlet c placed in the candidate networking element to the OLT l

λ_{clxy}^{pt}	Processed traffic from cloudlet c placed in the candidate networking element to the OLT l passing through the link between the nodes pair (x,y)
I_{vc}	$I_{vc} = 1$ if the virtual machine v is placed in the cloudlet c, otherwise $I_{vc} = 0$
H_c	$H_c = 1$ if a cloudlet c is built at the candidate networking element, otherwise $H_c = 0$
TW_c	Total normalised workload of the cloudlet c built at candidate networking element
PC^{rp}	Total processing induced power consumption of the relays
PC^{cp}	Total processing induced power consumption of the coordinators
PC^{gp}	Total processing induced power consumption of the gateways
PC^{up}	Total processing induced power consumption of the ONUs
PC^{lp}	Total processing induced power consumption of the OLTs
PC^{otr}	Total traffic induced power consumption of the IoT objects
PC^{rtr}	Total traffic induced power consumption of the relays
PC^{ctr}	Total traffic induced power consumption of the coordinators
PC^{gtr}	Total traffic induced power consumption of the gateways
PC^{utr}	Total traffic induced power consumption of the ONUs
PC^{ltr}	Total traffic induced power consumption of the OLTs

The total IoT processing induced power consumption is composed of:

1) The processing induced power consumption of each relay:

$$PC^{rp} = TW_c \cdot RMP \tag{4-1}$$

$$\forall c \in R$$

2) The processing induced power consumption of each coordinator:

$$PC^{cp} = TW_c \cdot CMP \quad (4\text{-}2)$$

$$\forall c \in C$$

3) The processing induced power consumption of each gateway:

$$PC^{gp} = TW_c \cdot GMP \quad (4\text{-}3)$$

$$\forall c \in G$$

4) The processing induced power consumption of each ONU:

$$PC^{up} = TW_c \cdot UMP \quad (4\text{-}4)$$

$$\forall c \in ONU$$

5) The processing induced power consumption of the OLT:

$$PC^{lp} = TW_c \cdot LMP \quad (4\text{-}5)$$

$$\forall c \in OLT$$

The processing induced power consumption of all processing elements in our proposed network (relays, coordinators, gateways, ONUs and OLT) are evaluated in equations (4-1) to (4-5). The processing induced power of each element is a function of its CPU maximum power and total normalised workload utilisation of the cloudlet placed in the element.

The total IoT traffic induced power consumption is composed of:

1) The traffic induced power consumption of each IoT object :

$$PC^{otr} = \sum_{y \in R} \lambda_{xy}^{upt} \cdot \left(E^{ot} + \epsilon \cdot d_{xy}^{2}\right) \quad (4\text{-}6)$$

$$\forall x \in O$$

2) The traffic induced power consumption of each relay:

$$PC^{rtr} = \sum_{y \in R \cup C: y \neq x} \left(\lambda_{xy}^{upt} + \lambda_{xy}^{pt}\right) \cdot \left(E^{rt} + \epsilon \cdot d_{xy}^{2}\right) + \sum_{y \in O \cup R: y \neq x} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{rr} \quad (4\text{-}7)$$

$$\forall\, x \in R$$

3) The traffic induced power consumption of each coordinator:

$$PC^{ctr} = \sum_{y \in G} \left(\lambda_{xy}^{upt} + \lambda_{xy}^{pt}\right) \cdot \left(E^{ct} + \epsilon \cdot d_{xy}^{2}\right) + \sum_{y \in R} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{cr} \quad (4\text{-}8)$$

$$\forall\, x \in C$$

4) The traffic induced power consumption of each gateway:

$$PC^{gtr} = \sum_{y \in ONU} \left(\lambda_{xy}^{upt} + \lambda_{xy}^{pt}\right) \cdot E^{gt} + \sum_{y \in C} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{gr} \quad (4\text{-}9)$$

$$\forall\, x \in G$$

5) The traffic induced power consumption of each ONU:

$$PC^{utr} = \sum_{y \in OLT} \left(\lambda_{xy}^{upt} + \lambda_{xy}^{pt}\right) \cdot E^{u} + \sum_{y \in G} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{u} \quad (4\text{-}10)$$

$$\forall\, x \in ONU$$

6) The traffic induced power consumption of the OLT:

$$PC^{ltr} = \sum_{y \in ONU} \left(\lambda_{yx}^{upt} + \lambda_{yx}^{pt}\right) \cdot E^{l} \quad (4\text{-}11)$$

$$\forall\, x \in OLT$$

Traffic induced power consumption components of our proposed network are represented by equations (4-6) to (4-11). The general structure of these equations is based on radio energy dissipation equation (Friis free-space equation) used in [31]. These equations are comprised of two basic parts the sending part and receiving part. Both parts are based on bit rate times the propagation energy per bit. Equation (4-6) represents the traffic induced power consumption of the IoT objects. This equation considers the sending traffic only because the traffic received by the IoT objects is considered in this model as signalling messages with small data size that can be ignored. On the other hand, equation (4-11) considers only the receiving traffic induced power consumption of OLT as the OLT layer is the highest layer in the model.

Objective: Minimise

$$\sum_{c\in R} PC^{rp} + \sum_{c\in C} PC^{cp} + \sum_{c\in G} PC^{gp} + \sum_{\forall c\in ONU} PC^{up} + \sum_{\forall c\in OLT} PC^{lp} + \sum_{x\in O} PC^{otr} + A \cdot \left(\sum_{x\in R} PC^{rtr} + \sum_{x\in C} PC^{ctr} + \sum_{x\in ONU} PC^{utr} + = \sum_{x\in OLT} PC^{ltr} \right) + \sum_{x\in G} PC^{gtr} \quad (4\text{-}12)$$

The model objective is to minimise the PON and IoT network power consumption due to traffic processing and aggregation as presented in equation (4.12). The scaling factor A is introduced to examine the case where the traffic induced power consumption in the networking elements is comparable to their processing induced power consumption.

Subject to:

1) IoT network un-processed traffic constraints

$$\sum_{\forall\, c\, \in CN} \lambda_{ovc}^{upt} = \lambda_{ov}^{upt} \tag{4-13}$$

$$\forall\, o \in O, \forall\, v \in VM$$

$$\lambda_{oc}^{upt} = \sum_{\forall\, v \in VM} \lambda_{ovc}^{upt} \tag{4-14}$$

$$\forall\, o \in O, \forall c \in CN$$

$$\sum_{y \in N_x} \lambda_{ocxy}^{upt} - \sum_{y \in N_x} \lambda_{ocyx}^{upt} = \begin{cases} \lambda_{oc}^{upt} & if\ x = o \\ -\lambda_{oc}^{upt} & if\ x = c \\ 0 & otherwise \end{cases} \tag{4-15}$$

$$\forall\, o \in O, \forall c \in CN, \forall x \in TN$$

$$\lambda_{xy}^{upt} = \sum_{c\, \in\, CN} \sum_{o\, \in\, O} \lambda_{ocxy}^{upt} \tag{4-16}$$

$$\forall\, x \in TN, \forall y \in N_x$$

Constraint (4.13) distributes the unprocessed traffic from IoT objects (*o*) over a number of VM (*v*) instances that are hosted in different mini cloudlets (*c*). It ensures that the total un-processed traffic flows from the IoT object (*o*) to all VM (*v*) instances in different mini cloudlets (*c*) equals to the traffic between that object (*o*) and the VM (*v*). Constraint (4-14) calculates the traffic flowing from IoT objects to each networking element. It ensures that the total un-processed traffic from the IoT object o to all the virtual machines v placed in cloudlet c is equal to the un-processed traffic from the object o to cloudlet c

placed in candidate networking element. Constraint (4-15) represents the flow conservation for the un-processed traffic from the IoT object o to cloudlet c located in candidate networking element. It ensures that the total un-processed outgoing traffic is equal to the total un-processed incoming traffic for each IoT node except for the source and the destination. Constraint (4-16) represents the total unprocessed traffic between any IoT node pair (x,y).

2) IoT network processed traffic constraints

$$\sum_{\forall l \in OLT: c \notin OLT} \lambda_{cl}^{pt} = F \cdot \sum_{\forall o \in O} \lambda_{oc}^{upt} \tag{4-17}$$

$$\forall c \in CN$$

$$\sum_{y \in N_x \cap CN} \lambda_{clxy}^{pt} - \sum_{y \in N_x \cap CN} \lambda_{clyx}^{pt} = \begin{cases} \lambda_{cl}^{pt} & if\ x = c \\ -\lambda_{cl}^{pt} & if\ x = l \\ 0 & otherwise \end{cases} \tag{4-18}$$

$$\forall c \in CN, \forall l \in OLT, \forall x \in CN: c \neq l$$

$$\lambda_{xy}^{pt} = \sum_{c \in CN} \sum_{l \in OLT: c \neq l} \lambda_{\text{clxy}}^{pt} \tag{4-19}$$

$$\forall x \in CN, \forall y \in N_x \cap CN$$

Constraint (4-17) calculates the reduced traffic flowing from the candidate networking element hosted in cloudlet c to the OLT l. Constraint (4-18) represents the flow conservation for the processed traffic from the candidate networking element hosted cloudlet c to the OLT l. It ensures that the total processed outgoing traffic is equal to the total processed incoming traffic for each IoT and PON node except for the source and the destination. Constraint

(4-19) represents the total processed traffic between any IoT and PON node pair (x,y).

3) Virtual machine placement and workload constraints

$$\sum_{o \in O} \lambda_{ovc}^{upt} \geq I_{vc} \tag{4-20}$$

$$\forall v \in VM, \forall c \in CN$$

$$\sum_{o \in O} \lambda_{ovc}^{upt} \leq \beta \cdot I_{vc} \tag{4-21}$$

$$\forall v \in VM, \forall c \in CN$$

$$\sum_{v \in VM} I_{vc} \geq H_c \tag{4-22}$$

$$\forall c \in CN$$

$$\sum_{v \in VM} I_{vc} \leq \gamma . H_c \tag{4-23}$$

$$\forall c \in CN$$

$$TW_c = \sum_{v \in VM} W_{vc} \cdot I_{vc} \tag{4-24}$$

$$\forall c \in CN$$

Constraints (4-20) and (4-21) place the virtual machine v in the cloudlet c if the cloudlet c is serving some IoT objects requests for this virtual machine. β is a large enough number with units of bps to ensure that I_{vc}= 1 when $\sum_{o \in O} \lambda_{ovc}^{upt}$ is greater than zero, otherwise I_{vc}= 0. Constraints (4-22) and (4-23) build a cloudlet c in the candidate networking element if this networking element is chosen to host at least one virtual machine v, where γ is a large enough unitless number to ensure that H_c= 1 if $\sum_{v \in VM} I_{vc}$ is greater than zero, otherwise H_c= 0. Constraint (4-24) calculates the total normalised workload of each built cloudlet c.

4.3 MILP evaluation and results

As mentioned earlier, we have considered two separated IoT networks connected to a PON access network. Each IoT network consists of 50 IoT objects, 25 relays, one coordinator and one gateway. In addition, each IoT network is connected to an ONU, both ONUs are connected to one OLT. The IoT objects, relay elements and the coordinator in each IoT network are distributed through 30m×30m area with same distribution as in Chapter 3 (shown in Figure 3.2). The gateway is placed 100m away from the coordinator. The distribution of IoT objects is random and uniform while relay elements are located every 6m. All devices in the IoT network communicate using the Zigbee protocol. On the other hand, the gateway is connected to the ONU through Gigabit Ethernet link and the ONU is connected to the OLT through an optical fiber. Only the uplink direction has been considered as it carries the highest amount of traffic. Consequently traffic is not allowed to pass from one IoT network to another through the OLT. Our model accounts for the traffic induced power consumption in PON entities as well as in the receiving and transmitting components of the IoT network (including propagation losses and the power amplification) [89]. The input parameters of the model are listed in Table 4.3. In terms of power consumption, two parts are considered for each network element in the proposed network; namely the communication and processing parts. The specifications of communication part used in objects, relays and coordinator are based on [90] while we used Cisco 910 industrial router [91] for the communication part of the gateway. In addition we used FTE7502 EPON ONU [96] and FSU7100 EPON OLT [97] as the ONU and OLT elements in the proposed network. The relays, coordinator, gateway, ONU and OLT elements are equipped with Intel Atom Z510 CPU [92] used for processing. We have considered a range of traffic reduction percentages after processing in order to investigate different impacts of processing applications.

Table 4-3 List of input parameters

Parameter Name	**Value**
Traffic sent from IoT object to a VM type (λ_{ov}^{upt})	5kbps [93]
CPU maximum power consumption (RMP, CMP, GMP)	4.64 W[92]
Number of CPUs used in a relay, coordinator, gateway, ONU and OLT.	1, 2, 4, 4, 10
IoT object, relay and coordinator transmitting energy per bit (E^{ot}, E^{rt},E^{ct})	50nJ/bit [90]
Relay and coordinator receiving energy per bit (E^{or}, E^{rr},E^{cr})	50nJ/bit [90]
Gateway receiving energy per bit (E^{gr})	60µJ/bit [91]
Gateway sending energy per bit (E^{gt})	15nJ/bit [91]
ONU energy per bit (E^{u})	7.5 nJ/bit [96]
OLT energy per bit (E^{l})	225.6 pJ/bit [97]

Transmission amplifier power coefficient (ϵ)	255 pJ/(bit.m^2)[90]
VM type 1 normalised workload in relay, coordinator, gateway, ONU and OLT elements (W_{1c})	0.1, 0.05, 0.025, 0.025, 0.01 [94]
VM type 2 normalised workload in relay, coordinator, gateway, ONU and OLT elements (W_{2c})	0.2, 0.1, 0.05, 0.05, 0.02 [94]
VM type 3 normalised workload in relay, coordinator, gateway, ONU and OLT elements (W_{3c})	0.3, 0.15, 0.075, 0.075, 0.03 [94]
VM type 4 normalised workload in relay, coordinator, gateway, ONU and OLT elements (W_{4c})	0.4, 0.2, 0.1, 0.1, 0.04 [94]
Traffic reduction percentage (F)	{10, 30, 50, 70, 90}%
Distance between node pair (x, y) in the IoT network, in meters (d_{xy})	Within 30m × 30m [95]
γ, β, A	50, 10000000 bps, 5

The CPU utilisation of the VMs belonging to a certain type is assumed to be independent of both the number of served IoT objects and the different traffic reduction percentages. We have considered three scenarios. In the first scenario, we have considered four VM types with heterogeneous VMs CPU demands ranging from 10% to 40% CPU utilisation. The second scenario considered four

VM types with high homogeneous CPU requirements of 40%. Finally, the third scenario considered four VM types with homogeneous CPU requirements of 40%, similar to scenario 2, however the OLT is equipped with a CPU with lower energy efficiency (9.28W power consumption, but similar processing capability). This setting allows us to assess the framework at different CPU demands and energy efficiency levels. Figures 4.2, 4.4 and 4.5 show the three scenarios processing, traffic and total power consumption respectively, while Figures 4.5, 4.6 and 4.7 show the VMs placement for the three scenarios.

Scenario 1 produces the lowest processing induced power consumption at low reduction percentage (10%, Figure 4.2) as it evaluates heterogeneous VMs and is able to place some of these VMs in the OLT (10%, Figure 4.5). This placement reduces the total number of VM copies needed; as placing VMs in any other layer duplicates them because the two IoT networks are not allowed to pass traffic between them due to the downlink restriction. Scenario 2 places more VMs at the OLT as it considers VMs with high and homogeneous CPU utilisation at low reduction percentages (10%, Figure 4.6). It however, still consumes higher CPU induced power compared to scenario 1 as all VMs consumes high power (10%, Figure 4.2). Scenario 3 results in the highest CPU induced power consumption at low reduction percentage (10%, Figure 4.2) as the OLT is equipped with energy inefficient CPU, resulting in placing the VMs in the lower layers as shown (10%, Figure 4.7). Note that all scenarios place VMs at the relay layer for both IoT networks at high reduction percentages (50% - 90%, Figures 4.5, 4.6, 4.7) as this leads to the minimum traffic induced power consumption at upper layers. As Scenario 1 considers heterogeneous VMs. It continues to produce the lowest CPU induced power consumption compared to the other two scenarios which have similar CPU induced power consumption (30% - 90%, Figure 4.2) as both serve VMs with similar CPU utilisation of 40% at the relay element.

As shown in Figure 4.3, we notice a general trend towards lower network power consumption with higher reduction percentages. This is attributed to the lower traffic pushed in the network as useful extracted knowledge has lower data rate compared to the raw unprocessed traffic. Scenario 3 produces the lowest traffic induced power consumption at low reduction percentage (10%, Figure 4.3) as it is able to place more VMs at the coordinator compared to other scenarios (10%, Figures 4.5, 4.6 and 4.7), allowing less knowledge traffic to pass though the upper layers. However, this saving in network induced power consumption is masked by the increase in CPU induced power consumption at low reduction percentages, leading to an overall high power consumption for scenario 3 compared to the other two scenarios (10%, Figure 4.4). Scenario 1 comes next in terms of traffic induced power consumption at low reduction percentages (10%, Figure 4.3) as it is able to place some VMs at lower layers (10%, Figure 4.5) compared to scenario 2 which prefers to place all VMs at the OLT layer (10%, Figure 4.6) resulting in the highest traffic induced power consumption (10%, Figure 4.3). In addition, scenario 1 results in slightly higher traffic induced power consumption at reduction percentage of 70% (Figure 4.3) as it is able to place more cloudlets in the relay layer than the other scenarios (70%, Figures 4.5, 4.6 and 4.7). Note that all scenarios consume the same traffic-induced power for 30%, 50% and 90% traffic reduction percentages as shown in Figure 4.3. This is influenced by the similar distribution of VMs copies for all these cases as shown in Figures 4.5, 4.6, 4.7. This identical distribution results from high reduction in traffic after processing by VMs, thus, the VMs are placed in relay elements as close as possible to the IoT objects. However, scenario 1 is the most energy efficient scenario considering total power consumption at all reduction percentages (Figure 4.4) as it has the lowest processing induced power consumption compared to the other two scenarios which compensates for the lower traffic energy efficiency. This results in about 17% and 19% of power saving for scenario 1 compared to scenario 2 and 3, respectively.

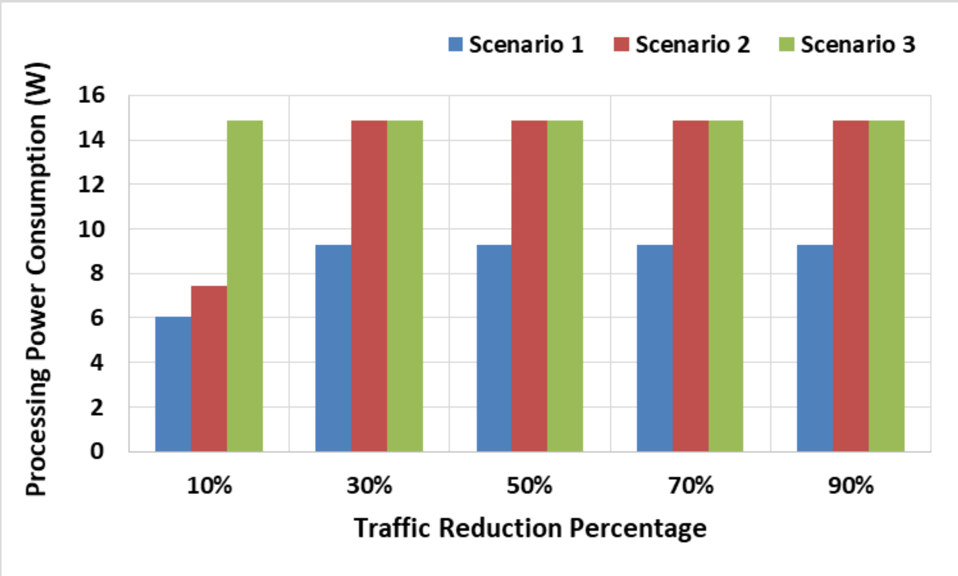

Figure 4.2 Processing power consumption of the three scenarios

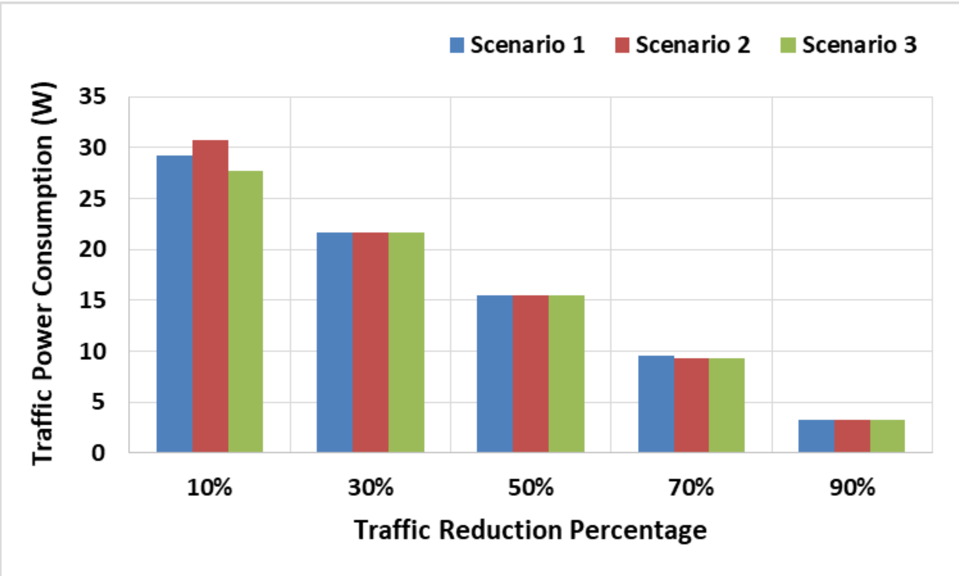

Figure 4.3 Traffic power consumption of the three scenarios

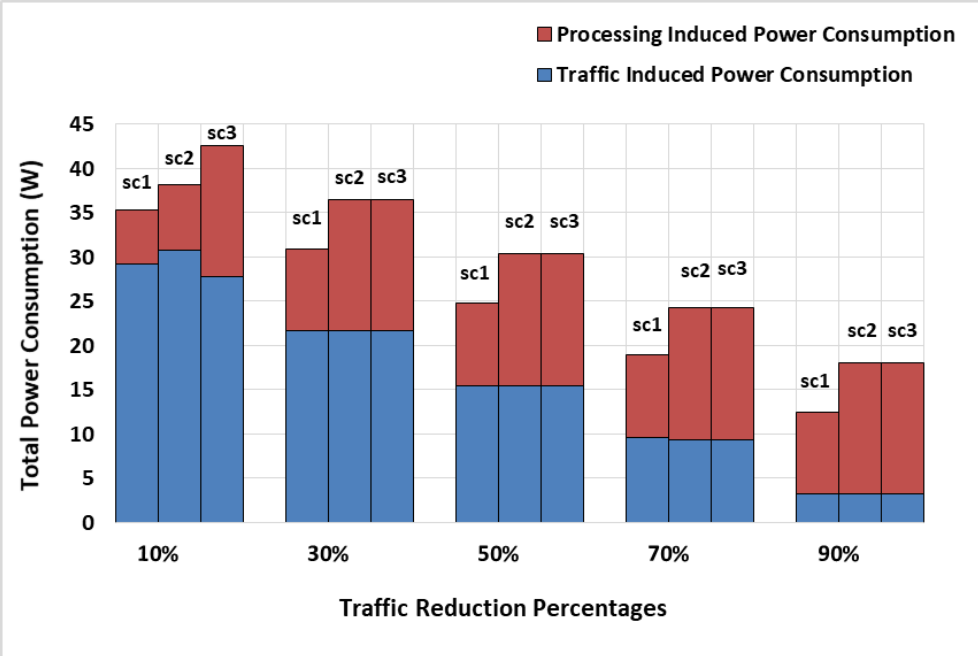

Figure 4.4 Total power consumption of the three scenarios

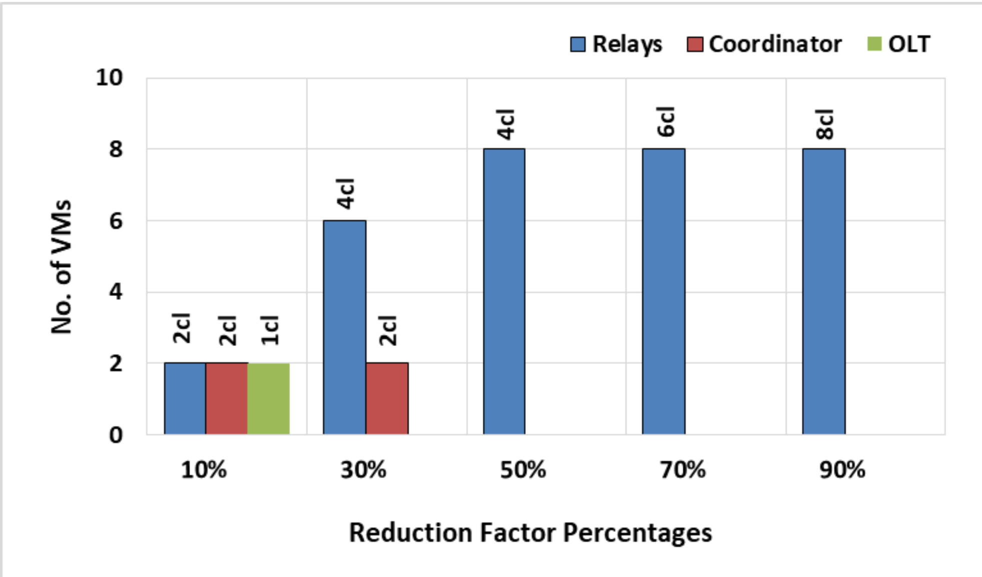

Figure 4.5 VMs placement in different cloudlets (cl) in scenario 1

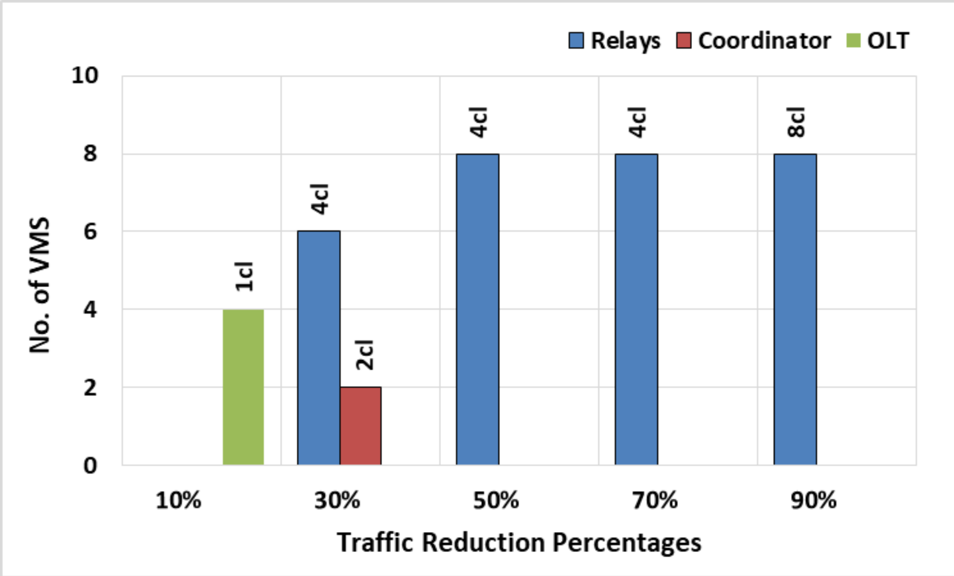

Figure 4.6 VMs placement in different cloudlets (cl) in scenario 2

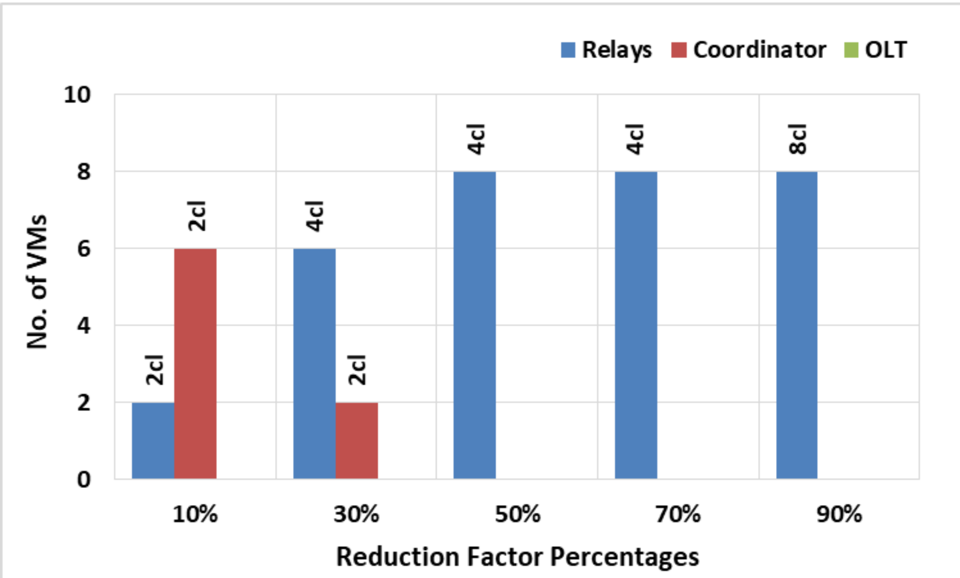

Figure 4.7 VMs placement in different cloudlets (cl) in scenario 3

4.4 EEPIV Heuristic

This section validates the MILP model results by presenting the EEPIV heuristic that mimics the MILP model behaviour. The pseudo code of EEPIV heuristic is presented by Figure 4.8. The heuristic shown in Figure 4.8 covers all the scenarios of our MILP model as implementing these scenarios relies on changing the input parameters not the constraints that the model is subjected to.

The heuristic calculates the total power consumption (TPC) of the network according the optimum place and number of the mini cloudlets that serve the IoT objects through the hosted VMs. Serving IoT objects by VMs is subject to the limited capabilities of the serving host VM in each cloudlet as below:

i. There should be sufficient processor capacity in each candidate cloudlet to accommodate the hosted VM workload.

ii. The intended VM v that is requested by IoT object o in each network should not have been hosted by any other cloudlet in this network before.

If all the serving constraints above are met, then heuristic hosts the intended VM by the candidate cloudlet to satisfy the IoT object request and sets the binary indicator F_{cv} accordingly. The total workload of each hosted cloudlet in the candidate place is calculated depending on the binary indicator F_{cv}.

Since the processing induced power consumption of each processing element is a function of the total workload of the cloudlet place in this element, the heuristic calculates the processing induced power consumption of all the processing elements in the proposed network (relays, coordinators, gateways, ONUs and OLT) as shown in steps 11 to 35 in Figure 4.8. The end-to-end traffic generated by the IoT objects' requests is next calculated by the heuristic. The traffic passes through two stages: the first stage flows from the generator (IoT object) to the destined VM in the hosting cloudlet which is represented by λ_{oc}^{upt} (unprocessed traffic). The second stage comes after the processing stage. In this stage, the processed traffic λ_{cl}^{pt} (reduced traffic) flows from the cloudlet to the last layer in the network which is represented in our proposed network by the OLT layer. The intermediate traffic between each node pair in the network is calculated by the heuristic model based on the end to end traffic. The heuristic routes the traffic through these intermediate nodes from the source to the destination using a minimum hop algorithm to reduce the traffic induced power consumption. Finally, the heuristic calculates the total power consumption TPC by summing all the processing and traffic induced power consumption of all nodes.

Inputs: $\mathbf{VM} = \{\mathbf{1} \ldots \boldsymbol{NVM}\}$

$\boldsymbol{CN} = \{\mathbf{1} \ldots \boldsymbol{NCN}\}$

$\boldsymbol{O} = \{\mathbf{1} \ldots \boldsymbol{NO}\}$

$\boldsymbol{R} = \{\mathbf{1} \ldots \boldsymbol{NR}\}$

$\boldsymbol{C} = \{\mathbf{1} \ldots \boldsymbol{NC}\}$

$\boldsymbol{G} = \{\mathbf{1} \ldots \boldsymbol{NG}\}$

$\boldsymbol{ONU} = \{\mathbf{1} \ldots \boldsymbol{NONU}\}$

$\boldsymbol{OLT} = \{\mathbf{1} \ldots \boldsymbol{NOLT}\}$

Output: No. of Served Objects

Total Power Consumption (TPC)

1. **For** each candidate cloudlet that can host a required VM c $\in CN$ **Do**
2. **For** each Virtual Machine required by an object $v \in VM$ **Do**
3. **If** $U_{ov} > 0$ **Then**
4. **If** all serving constraints are met **Then**
5. $F_{cv}(c, v) = 1$
6. Calculate the workload of the hosting cloudlet $TCWc$

without considering the number of served IoT objects

End If

End If

End For

End For

For Each relay ($r \in R$) **Do**

If the hosting cloudlet is placed in relay layer R $c \in CN$ **Do**

Calculate R_PPC

End If

End For

For Each coordinator ($c \in C$) **Do**

If the hosting cloudlet is placed in coordinator layer C $c \in CN$ **Do**

Calculate C_PPC

End If

End For

For each gateway ($g \in G$) **Do**

If the hosting cloudlet is placed in gateway layer $c \in CN$ **Do**

Calculate G_PPC

End If

End For

For each ONU ($u \in ONU$)

If the hosting cloudlet is placed in ONU layer $c \in CN$ **Do**

Calculate PC^{up}

End If

End For

For each OLT ($l \in OLT$)

If the hosting cloudlet is placed in OLT layer $c \in CN$ **Do**

Calculate PC^{up}

End If

End For

For each IoT object served by a cloudlet $o \in O$ **Do**

For each hosting cloudlet $c \in CN$ **Do**

Calculate end to end traffic that flows from each object to the cloudlet that serves this object

End For

End For

For Each hosting cloudlet $c \in CN$ **Do**

For Each OLT ($l \in OLT$) **Do**

Calculate the end to end reduced traffic from the cloudlet to the OLT

End For

End For

For each IoT object $o \in O$ **Do**

Calculate TO_tr

End For

For each relay ($r \in R$)

Calculate TR_tr based on minimum hop path between node pair (x,y)

End For

For each coordinator ($c \in C$)

Calculate TC_tr based on minimum hop path between node pair (x,y)

End For

For each gateway ($g \in G$)

Calculate TG_tr based on minimum hop path between node pair (x,y)

End For

For each ONU ($u \in ONU$)

Calculate ONU_tr based on minimum hop path between node pair (x,y)

End For

For each OLT ($l \in OLT$)

Calculate ONU_tr based on minimum hop path between node pair (x,y)

End For

Calculate total power consumption

$$TPC = \sum_{r \in R} R\_PPC + \sum_{c \in C} C\_PPC + \sum_{g \in G} G\_PPC + \sum_{u \in ONU} ONU\_PPC + \sum_{o \in O} TO\_tr + \sum_{r \in R} TR\_tr + \sum_{c \in C} TC\_tr + \sum_{g \in G} TG\_tr + \sum_{u \in ONU} ONU\_tr + \sum_{l \in OLT} OLT\_tr$$

Figure 4.8 pseudo code of EEPIV heuristic

4.5 EEPIV Heuristic results

We used the same inputs in Table 4.3 for the heuristic. The heuristic results show close agreement with the MILP results comparing Figure 4.9 with Figure 4.4.

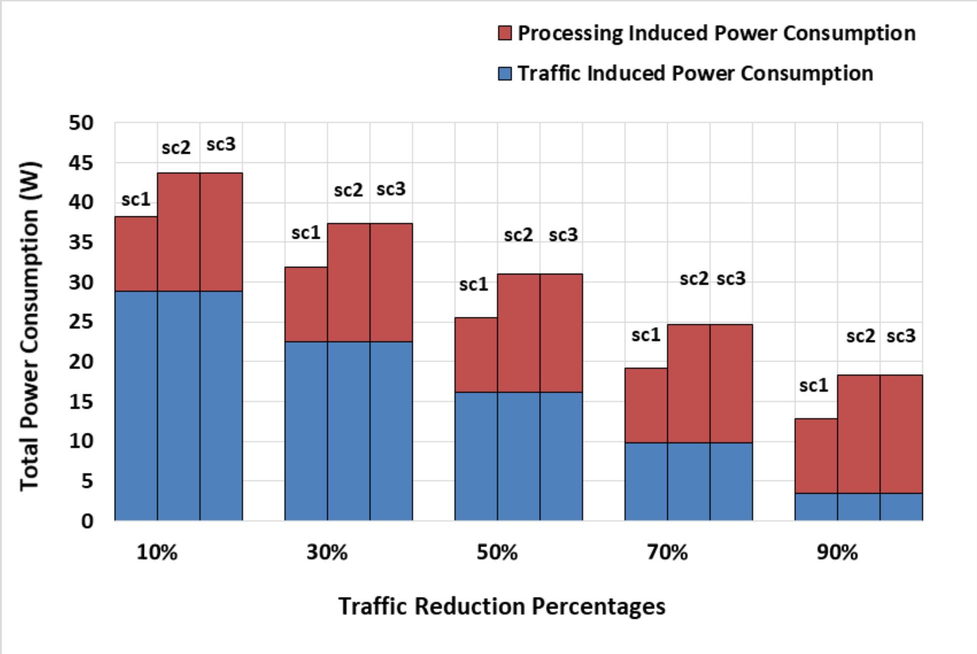

Figure 4.9 Total power consumption of the three scenarios in the heuristic

Scenarios 1 and 2 result in lower processing induced power consumption in MILP than in heuristic at low reduction percentage (10%, Figures 4.2 and 4.10). This results from placing/using more VM copies in the heuristic (8 VMs) than in the MILP as shown by Figures 4.5 and 4.6.

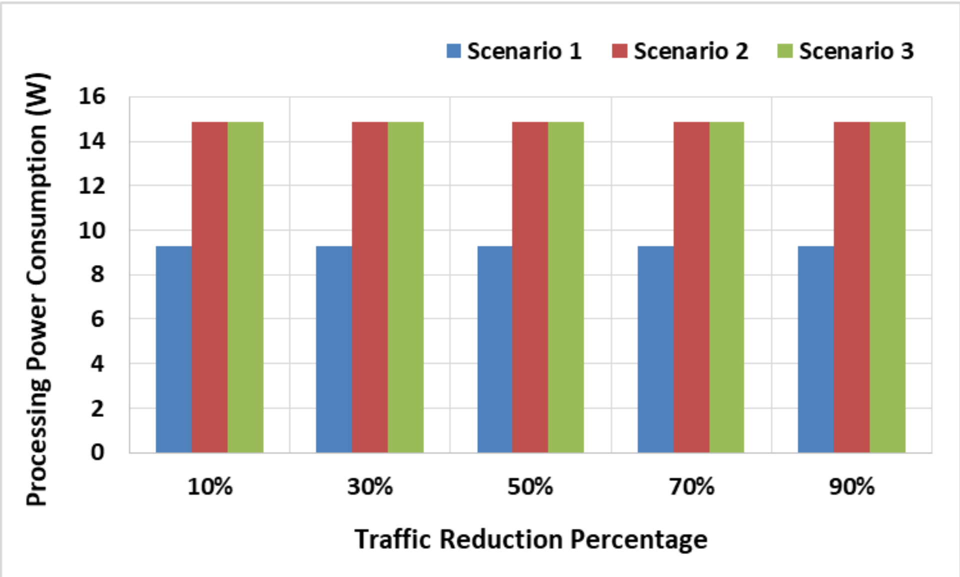

Figure 4.10 Processing power consumption of the three scenarios in the heuristic

Scenarios 1 and 2 in heuristic result in lower traffic induced power than in MILP at traffic reduction percentages of 10% (Figures 4.3 and 4.11). This results from the MILP placing the serving VMs in cloudlets at higher layers (Figures 4.5 and 4.6) while all the cloudlets in the heuristic are distributed throughout the lower layer (relay layer). Placing cloudlets in higher layers results in sending more unprocessed traffic (unreduced traffic) to higher layers which in turn results in higher traffic induced power consumption. However, all the scenarios in the heuristic consume higher traffic induced power than in MILP for the rest of the reduction percentage values as a result of the different distribution of the cloudlets in the proposed network. Since for each cloudlet, the heuristic attempts to place it in the first network element that can accommodate this cloudlet, the heuristic placed all the cloudlets in the relay layer without consideration of the closeness of the cloudlet to the IoT objects. On other side, the MILP places the cloudlets in an optimum way to minimise the traffic and processing induced power consumed by all elements of the proposed network.

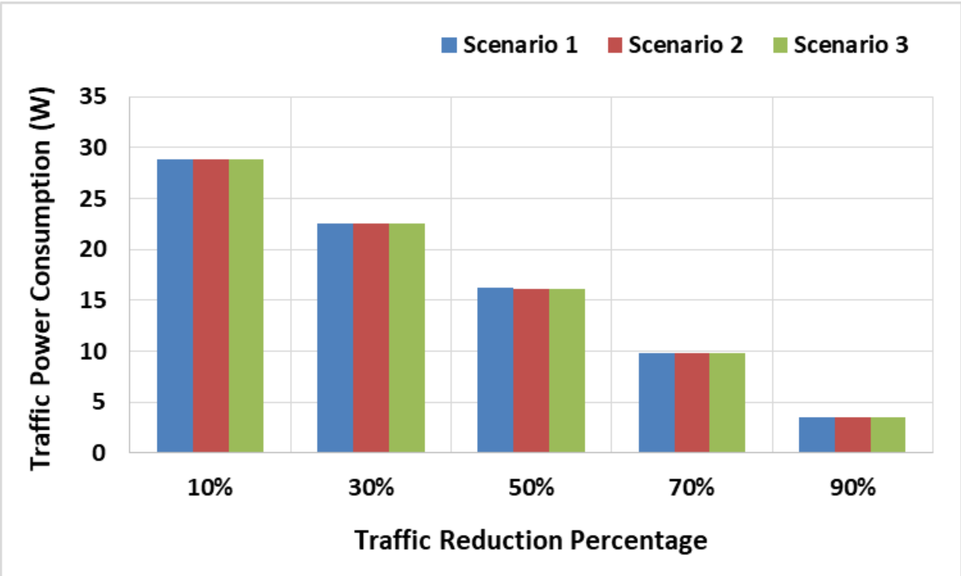

Figure 4.11 Traffic power consumption of the three scenarios in the heuristic

The number of cloudlets placed using the heuristic (2 cloudlets in scenario 1, and 4 cloudlets in scenarios 2 and 3 for all traffic reduction percentage values with 8 VMs in all scenarios) is less than in the MILP as one of main processes in the heuristic is bin packing where VMs must be packed into a finite number of bins (cloudlets) in a way that minimises the number of bins used.

4.6 Summary

In this chapter, we have investigated the energy efficiency of edge computing platforms for IoT networks connected to a PON. To achieve this, we have developed a MILP model, which optimises the placement and number of the cloudlets and VMs and utilises energy efficient routes with the objective of minimising the power consumption. Our results indicate that concentrating the VMs placement at the OLT connecting several IoT networks can help in saving power consumption when VMs process raw data at low traffic reduction percentages. On the other hand, VMs should be placed in lower layer relays at high traffic reduction ratios. Our results indicate that up to 19% of the total power can be saved while utilising PONs and serving heterogeneous VMs. For real time implementation, a heuristic is developed based on the MILP model insights with very comparable MILP-heuristic power consumption values. Scenario 1 in the heuristic achieves power savings of 17% (MILP 17%), and 17% (MILP 19%) compared to scenario 2 and scenario 3 respectively.

Chapter 5 : Energy Efficient Virtualised IoT P2P Networks

5.1 Introduction

Peer-to-peer (P2P) overlay networks can improve the communication performance of IoT networks. Energy efficiency is one of the main advantages that can be brought about by using P2P communication systems. In this chapter, an energy efficient virtualised IoT framework with P2P networking and edge computing is proposed. The proposed network encompasses IoT objects and relay devices. In this network, the IoT request for a processing task is served by peers. The peers in our work are represented by IoT objects and the virtual machines (VM) hosted in a device. We have considered three scenarios to investigate the saving in power consumption and the system capabilities in terms of tasks processing. The first scenario is the VMs only scenario, where the task requests are processed using VMs hosted in relays only. The second scenario is the objects only scenario, where the task requests are processed using the IoT objects only. The last scenario is the hybrid scenario, where the task requests are processed using both IoT objects and VMs. We have developed a MILP model to maximise the number of tasks deployed by the system and minimise the total power consumed by the IoT network. We investigated our model under the impact of VMs placements, fairness constraints between the objects, tasks number limitations, uplink and downlink capacities, and processing capability limitations. Based on the MILP model principles, we developed an energy efficient virtualised IoT P2P networks heuristic (EEVIPN) with comparable results in terms of energy efficiency and tasks processing.

5.2 Energy Efficient MILP for P2P IoT Networks

The MILP model developed considers the architecture shown in Figure 5.1. The proposed architecture is constructed of two typical layers. The first layer is represented by the IoT objects. The upper layer consists of the relay devices that realise traffic transportation between peers. In our framework, each object is capable of processing three types of tasks that are required by other objects. The task processing capabilities and task requirements for the IoT objects are specified by the MILP model parameters. Each relay node has the capability of hosting VMs in order to process the tasks requested by IoT objects. The number of possible VMs locations is limited to 10 out of 25 possible locations. These VMs have the ability to handle all task types.

Figure 5.1 illustrates all cases of processing requests that we have considered in our P2P platform. Internal processing is shown in case a, where the object has the ability to process its own request. Consequently, the network power consumption associated with sending the task request to another object or VM or receiving a task result from them will be eliminated. One application of this case might be in smart night lights. In case b, the object sends its task request to the object's neighbour (the directly connected relay device) to be processed by the hosted VM, for example a healthcare device. Some of the objects in our model have the ability to process task requests generated by other objects but considering fairness constraint limitations. The fairness constraint states that each object should reciprocate equally to other objects choosing it to process its requested task. Object to object communication such as two Arduino devices with different capabilities is illustrated in case c. The last task processing case is case d. In this case, none of the objects themselves or the other objects or even the VM hosted by its directly connected relay have the ability to process the requested tasks. In spite of that, VMs can process all types of tasks, but the capacity of each VM-processor is limited to a specific maximum workload. So, in order to process this task, the relay sends the task request to other relays to be processed by the nearest possible VM (keep in mind that not all the relays host VMs) such as a smart camera sending small size images to be processed.

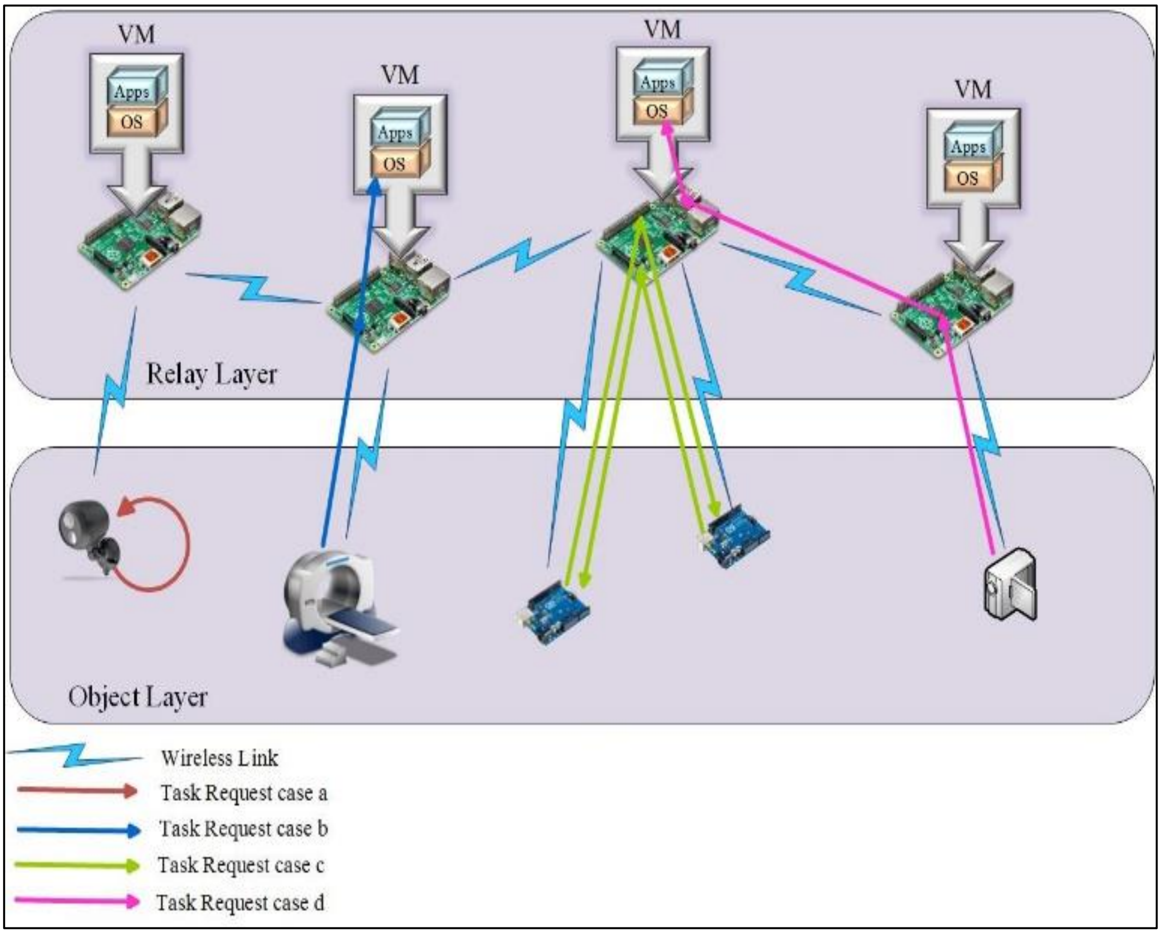

Figure 5.1The proposed architecture with P2P communication

The MILP model objective consists of two main parts. The first part maximises the number of logical end-to-end connections between objects and between both VMs and other objects. Maximising this number means maximising the number of served tasks. The second part of the objective considers minimising the total power consumption of all elements in our network. The total power consumption in our model falls into two parts. The first part is the traffic induced power consumption in objects and relays caused by uplink and downlink traffic flow through the network. The uplink traffic is generated by the task requests (the row data) while the downlink traffic is the reduced traffic generated after task processing (the information). The second part of the power consumption equation represents the processing induced power consumption in objects and relays produced by the tasks processing in objects and hosted VMs. Note that the processing in rest of our paper is indicated as VMs not the hosting relays. For example, we point out the processing induced power consumed by relays as the processing induced power consumption in VMs.

The MILP model objective is subject to many constraints. These constraints are related to VMs placements, fairness constraint between the objects, tasks number limitations, uplink and downlink capacities, and processing capability limitations.

For more clarity in the MILP expression and notations, we have used superscripts to index the type of variables and the parameters while we have used the subscripts as indices of these variables and parameters.

First, the sets, parameters, and variables of our P2P IoT MILP model are defined in Tables 5.1 and 5.2:

Table 5-1 List of parameters and their definitions

Notation	Description
O	Set of objects
VM	Set of virtual machines
R	Set of relays
P	Set of peers $(O \cup VM)$
TN	Set of all IoT network nodes ($TN = P \cup R$)
P_r^R	Set of peers of relay r
N_a	Set of neighbours of node a
K	Set of tasks
K^p	Subset of tasks that can be served by each peer p. $K^p \subset K$

R_p^P	Neighbour relay of peer p if the peer is object or hosting relay of peer p if it is VM
Q_{ik}	Task *k* required by object *i*
W_k	The workload required by each task *k* (GHz)
B^V	The number of possible locations occupied by VMs
ψ_j^V	The processor capacity of each virtual machine j (GHz)
Ω_j^V	The maximum power consumed by each virtual machine j (W)
ψ_j^O	The processor capacity of each object j (GHz)
Ω_j^O	The maximum power consumed by the processor used in each object j (W)
M_k	The traffic demand of each task *k* (bit/s) (row data)
C_k	The traffic resulting after processing each task *k* (bit/s) (information)
L^{DO}	Maximum traffic that can be downloaded by each object (bits/s)
L^{DV}	Maximum traffic that can be downloaded by each VM (bits/s)
L^{UO}	Maximum traffic that can be uploaded by each object (bits/s)
L^{UV}	Maximum traffic that can be uploaded by each VM (uploading tasks results) (bits/s)
X_i	Maximum number of upload slots for each object i
E^{elec}	Energy consumed per bit by the electronics of the transceiver (Joules/bit)
D_{mn}	Distance between any node pair in the IoT network (*m*,*n*) (meter) $m,n \in TN$
ϵ	Transmission amplifier power coefficient (Joule/(bit.m^2))
F	Task weight factor

Table 5-2 List of variables and their definitions

Notation	Description
U_{ijk}	Binary variable which is set to 1 if peer j processes task k requested by object i, otherwise it is set to 0
V_j	Binary variable which is set to 1 if there is a virtual machine in that location otherwise it is set to 0
I_j^{DM}	Download rate (downloading task request) for each peer j (kbps)
I_i^{DC}	Download rate (downloading task result) for each object i (kbps)
I_i^{UM}	Upload rate (uploading task request) for each object i (kbps)
I_j^{UC}	Upload rate (uploading task result) for each peer j (kbps)
λ_{xy}^Q	Total traffic passing from relay x (neighbour of source object) to relay y (neighbour of destination peer or hosting the destination peer)
λ_{xy}^S	Total traffic (tasks result traffic) passing from the relay x (neighbour of

	source peer or hosting the destination peer) to relay y (neighbour of destination object)
λ_{xyab}^{Q}	Relay to relay traffic (x, y) passing through the link between the intermediate relays pair (a, b)
λ_{xyab}^{S}	Relay to relay traffic (x, y) (tasks results traffic) passing through the link between the intermediate relays pair (a, b)
λ_{ab}^{Q}	Intermediate traffic between any two relays pair (a, b)
λ_{ab}^{S}	Intermediate traffic (tasks results traffic) between any two relays pair (a, b)

The total IoT network power consumption is composed of:

1. The processing induced power consumption of each peer can be calculated by summing the workloads of all processed tasks by the peer and multiplying the summation by the energy per processed bit. The processing power in our work is composed of two parts:
   a) Processing induced power consumption of each object:

$$P_j^{op} = \left(\sum_{i \in O, k \in K_j^P} U_{ijk} \,.\, W_k \right) \cdot \frac{\Omega_j^O}{\psi_j^O} \qquad \text{(5-1)}$$

$$\forall\, j \in O$$

   b) Processing induced power consumption of each VM:

$$P_j^{vp} = \left(\sum_{i \in O, k \in K_j^P} U_{ijk} \,.\, W_k \right) \cdot \frac{\Omega_j^V}{\psi_j^V} \qquad \text{(5-2)}$$

$$\forall\, j \in VM$$

2. The traffic induced power consumption equations consist of two basic parts, the sending part and the receiving part. Both parts are based on radio energy

dissipation equation (Friis free-space equation) used in [31]. The power consumption is equal to the bit rate times the propagation energy per bit.

a) The traffic induced power consumption of each object:

$$P_i^{otr} = \sum_{j \in P} \sum_{k \in K_j^P : i \neq j} U_{ijk} \, . \, M_k . \, \left(E^{elec} + \epsilon \, . \, D_{ig}^2\right)$$

$$+ \left(\sum_{j \in O: i \neq j} \sum_{k \in KP_j} U_{jik} \, . \, C_k . \, \left(E^{elec} + \epsilon \, . \, D_{ig}^2\right) \right)$$

$$+ \sum_{j \in O: i \neq j} \sum_{k \in K_j^P} U_{jik} \, . \, M_k \, . \, E^{elec}$$

$$+ \left(\sum_{j \in P: i \neq j} \sum_{k \in KP_i} U_{ijk} \, . \, C_k \, . \, E^{elec} \right) \quad (5\text{-}3)$$

$$g = R_i^P$$

$$\forall \, i \in O$$

The first two terms represent the sending power while the third and fourth parts represent the receiving power. The first term calculates the power consumed by each object in sending its requests to other peers in order to process them. The second part represents the power consumed by each object in sending back the results of the tasks processed by itself to the original request generator. The third part represents the power consumed by each object in receiving the task requests from other objects. The last part shows the power consumed by each object in receiving the results of its task requests.

b) Traffic induced power consumption of each relay:

$$P_a^{rtr} = \sum_{b \in N_a \cap R : a \neq b} \left(\lambda_{ab}^{Q} \, . \, (E^{elec} + \, \epsilon \, . \, D_{ab}^2)\right)$$

$$+ \sum_{b \in N_a \cap R : a \neq b} \left(\lambda_{ab}^{S} \, . \, (E^{elec} + \, \epsilon \, . \, D_{ab}^2)\right)$$

$$+ \sum_{j \in P_a^R \cap O} \left(I_j^{DM} . \, \left(E^{elec} + \, \epsilon . D_{aj}^2 \,\right)\right)$$

$$+ \sum_{i \in P_a^R \cap O} \left(I_i^{DC} . \, (E^{elec} + \, \epsilon . D_{ai}^2 \,)\right)$$

$$+ \sum_{b \in N_a \cap R : a \neq b} \left(\lambda_{ba}^{Q} \, . \, E^{elec}\right) \qquad (5\text{-}4)$$

$$+ \sum_{b \in N_a \cap R : a \neq b} (\lambda_{ba}^{S} \, . \, E^{elec})$$

$$+ \sum_{i \in P_a^R \cap O} I_i^{UM} . E^{elec} \, + \sum_{j \in P_a^R \cap O} I_j^{UC} . E^{elec}$$

$$\forall \, a \in R$$

Traffic induced power consumption of the relays P_a^{rtr} consists of 8 terms. The first four terms represent the sending power and the last four terms represent the receiving power. The first and second terms represent the power consumed in sending the task requests and task results respectively from a relay to another relay. The third and fourth terms calculate the power consumed in sending the task requests and task results respectively to the objects directly connected to that relay. The fifth and sixth terms describe the power consumed in receiving task requests and task results respectively by each relay from another neighbuor relay. The seventh term calculates the power consumed by each relay in receiving the task requests from the directly connected object while the last term represents the power consumed by each relay in receiving the task results from other peers (directly connected object to the relay or VM hosted by this relay).

Objective: **Maximise**

$$\left(\sum_{i\in O, j\in P, k\in K_j^P} F\ .\ U_{ijk}\right) - \left(\sum_{j\in O} P_j^{op} + \sum_{j\in VM} P_j^{vp}\right)$$

$$-\left(\sum_{i\,\in\, O} (P_i^{otr}) + \sum_{a\in R} (\ P_a^{rtr})\right) \tag{5-5}$$

Equation (5-5) gives the model objective where the number of logical end-to-end connections between objects and other peers is maximised while the network power consumption and the processing power consumption are minimised. The parameter F takes care of the units and is also used to scale the number of connections so that they become comparable in magnitude to the consumed power.

Subject to:

1. U indicator setting constraints

$$\sum_{j\in P} U_{ijk} \leq Q_{ik} \tag{5-6}$$

$$\forall\ i \in O, \forall k \in K$$

Constraint (5-6) ensures that only one peer (one object or one VM) can serve each request of each object.

2. Fairness constraints

$$\sum_{k \in K_j^P} U_{ijk} = \sum_{k \in K_i^P} U_{jik} \qquad (5\text{-}7)$$

$$\forall\, i \in O, \forall\, j \in O$$

Constraint (5-7) is the fairness constraint which ensures that each object reciprocates equally to other objects that serve a request of this object.

3. Virtual Machine Calculations constraints

$$\sum_{i \in O, k \in K_j^P} U_{ijk} \geq V_j \qquad (5\text{-}8)$$

$$\forall\, j \in VM$$

$$\sum_{i \in O, k \in K_j^P} U_{ijk} \leq A.V_j \qquad (5\text{-}9)$$

$$\forall\, j \in VM$$

$$\sum_{j \in VM} V_j = \mathrm{B}^{\mathrm{v}} \qquad (5\text{-}10)$$

Constraints (5-8) and (5-9) locate a virtual machine in an appropriate relay in order to process the requested tasks. Constraint (5-10) limits the number of selected locations occupied by the virtual machines to 10 only out of 25 possible locations.

4. Processing power consumption calculations

$$\sum_{i \in O, k \in K_j^P} U_{ijk} \,.W_k \leq \psi_j^O \qquad (5\text{-}11)$$

$$\forall j \in O$$

$$\sum_{i\in O, k\in K_j^P} U_{ijk} \, . W_k \leq \; \psi_j^V \tag{5-12}$$

$$\forall j \in \; VM$$

Constraints (5-11) and (5-12) ensure that the summation of the whole workloads of processed tasks by each object and each VM respectively do not exceed its maximum processing workload capability

5. Traffic calculations and capacity constraints

$$\lambda_{xy}^{Q} = \left(\left(\sum_{i\in P_x^R \cap O} \left(\sum_{j\in P_y^R : i\neq j} \left(\sum_{k\in K_j^P} U_{ijk} \; . \; M_k \right) \right) \right) \right) \tag{5-13}$$

$$\forall \; x \in R, \forall \; y \in R: x \neq y$$

$$\lambda_{xy}^{S} = \left(\left(\sum_{j\in P_x^R} \left(\sum_{i\in P_y^R \cap O : i\neq j} \left(\sum_{k\in K_j^P} U_{ijk} \; . \; C_k \right) \right) \right) \right) \tag{5-14}$$

$$\forall \; x \in R, \forall \; y \in R: x \neq y$$

$$\sum_{b\in N_a \cap R : a\neq b} \lambda_{xyab}^{Q} - \sum_{b\in N_a \cap R : a\neq b} \lambda_{xyba}^{Q} = \begin{cases} \lambda_{xy}^{Q} & if \; a = x \\ -\lambda_{xy}^{Q} & if \; a = y \\ 0 & otherwise \end{cases} \tag{5-15}$$

$$\forall \; x \in R, \forall y \in R, \forall a \in R: \; x \neq y$$

$$\sum_{b\in N_a \cap R : a\neq b} \lambda_{xyab}^{S} - \sum_{b\in N_a \cap R : a\neq b} \lambda_{xyba}^{S} = \begin{cases} \lambda_{xy}^{S} & if \; a = x \\ -\lambda_{xy}^{S} & if \; a = y \\ 0 & otherwise \end{cases} \tag{5-16}$$

$$\forall \; x \in R, \forall y \in R, \forall a \in R: \; x \neq y$$

$$\lambda_{ab}^{Q} = \sum_{x\in R} \sum_{y\in R:\, x\neq y} \lambda_{xyab}^{Q} \quad (5\text{-}17)$$

$$\forall\, a \in R, b \in N_a \cap R: a \neq b$$

$$\lambda_{ab}^{S} = \sum_{x\in R} \sum_{y\in R:\, x\neq y} \lambda_{xyab}^{S} \quad (5\text{-}18)$$

$$\forall\, a \in R, b \in N_a \cap R: a \neq b$$

Constraints (5-13) and (5-14) calculate the transient traffic between relays due to P2P traffic (task requests and the results traffic). Constraints (5-15) and (5-16) represent the flow conservation of the traffic between the source relay (requester's (object) neighbour) and the destination relay (serving peer's neighbour or host) through the intermediate relays. Constraints (5-17) and (5-18) calculate the traffic flows through each intermediate relay.

$$I_j^{DM} = \sum_{i\in O, k\in K_j^P : i\neq j} U_{ijk} \,.\, M_k \quad (5\text{-}19)$$

$$\forall\, j \in P$$

$$I_i^{DC} = \sum_{j\in P, k\in K_j^P : i\neq j} U_{ijk} \,.\, C_k \quad (5\text{-}20)$$

$$\forall\, i \in O$$

$$I_j^{DM} \leq L^{DO} \quad (5\text{-}21)$$

$$\forall\, j \in O$$

$$I_i^{DC} \leq L^{DO} \tag{5-22}$$

$$\forall\, i \in O$$

$$I_j^{DM} \leq L^{DV} \tag{5-23}$$

$$\forall\, j \in VM$$

Constraint (5-19) calculates the download rate of each peer by summing the received traffic demand of each requested task from other objects selected to serve them. Constraint (5-20) calculates the download rate of the reduced traffic (resulting information) received by each object. Constraints (5-21), and (5-22) limit the download rate of each object to its maximum value, while constraint (5-23) limits the download rate of each VM to its maximum value.

$$I_i^{UM} = \sum_{j \in P, k \in K_j^P : i \neq j} U_{ijk} \,.\, M_k \tag{5-24}$$

$$\forall\, i \in O$$

$$I_j^{UC} = \sum_{i \in O, k \in K_j^P : i \neq j} U_{ijk} \,.\, C_k \tag{5-25}$$

$$\forall\, j \in P$$

$$I_i^{UM} \leq L^{UO} \tag{5-26}$$

$$\forall\, i \in O$$

$$I_j^{UC} \leq L^{UO} \tag{5-27}$$

$$\forall j \in O$$

$$I_j^{UC} \leq L^{UV} \tag{5-28}$$

$$\forall j \in VM$$

$$\sum_{j \in P, k \in K_j^P : i \neq j} U_{ijk} \leq X_i \tag{5-29}$$

$$\forall i \in O$$

$$\sum_{i \in O, k \in K_j^P : i \neq j} U_{ijk} \leq X_i \tag{5-30}$$

$$\forall j \in O$$

Constraint (5-24) calculates the upload rate of each object by summing the uploaded task traffic demands. While constraint (5-25) calculates the upload rate of each peer that results from sending the reduced traffic (the resulting information from task processing). Constraints (5-26), and (5-27) limit the upload rate of each object to its maximum value while constraint (5-28) limits the upload rate of each VM to its maximum value. Constraints (5-29) and (5-30) limit the number of upload slots of each object.

5.3 MILP Model Results

Our IoT nodes, depicted in Figure 5.1, consist of 25 objects and 25 relays distributed over an area of 30m × 30m [95]. The objects are distributed randomly while the relays are distributed uniformly, every 6m as shown in Figure 5.2. All devices in the IoT network communicate using the Zigbee protocol.

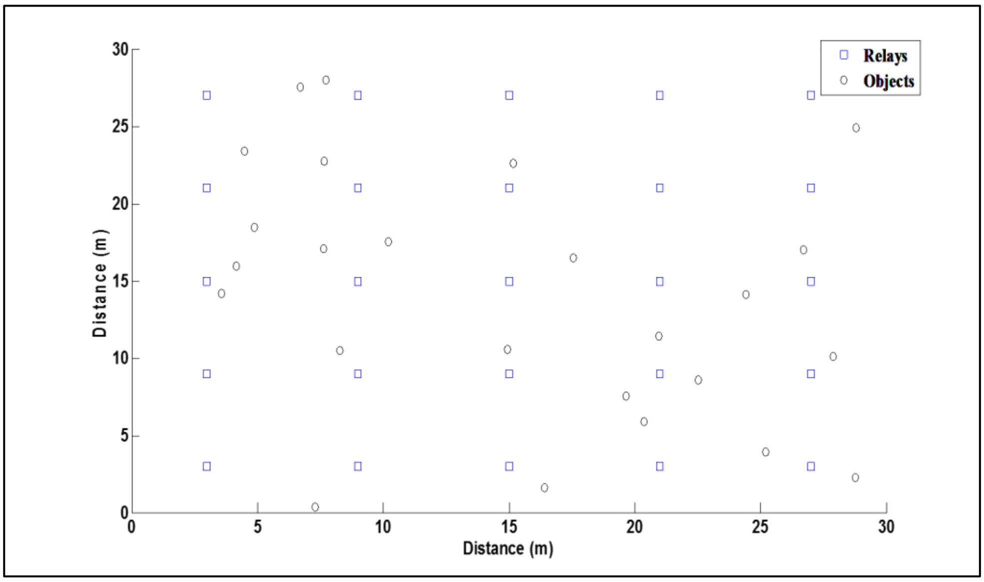

Figure 5.2 IoT distribution in space

Table 5.3 lists the model input parameters. We have used the Arduino 101 as an IoT object as it is one of the most power efficient processors with a higher clock speed compared to other types of Arduino [98]. Arduino 101 is referred to as Genuino 101 outside USA [99]. We used the Raspberry pi 3 in the relays, with processing capability of 1.2 GHz [100]. We assumed the traffic demand of the first task is 250 bit/s representing applications with small traffic volume in range of 0-250 bit/s. We assumed other values of traffic close to the first one in a consistent way to comply with the link capacity limit constraint and to be very close to practical IoT applications. The data rates thus considered were 240b/s representing a heartbeat sensor and 2.4 kb/s associated with blood glucose level sensors and temperature readings [101]. The range of traffic values considered resulted in heterogeneous tasks that have to be tackled by our optimisation model [102], [103]. In Bit Torrent the typical value for the maximum number of upload slots for each peer is 4 [104]. We have considered, in our P2P communication system, a range of different numbers of upload slots from 1 to 10 slots per object. We found that the average value of upload slots that ensures the highest percentage of executed tasks is 4.

As alluded to earlier, we have considered three scenarios. The first scenario is VMs only scenario. This restricts the processing of all requested tasks to 10 VMs out of 25 possible locations. This scenario is implemented by setting the number of end-to-end connections between the objects to zero, to ensure that no objects respond to any task request, i.e. equation (5-31):

$$\sum_{j \in O, k \in K_j^P} U_{ijk} = 0 \qquad (5\text{-}31)$$

$$\forall\, i \in O$$

The second scenario is the objects only scenario which restricts the processing of the requested tasks by the IoT to objects only. This scenario is implemented by setting the total number of VMs to zero, i.e. equation (5-32):

$$\sum_{j \in VM} V_j = 0. \qquad (5\text{-}32)$$

The last scenario allows cooperation between the VMs and the objects in order to process the requested tasks. Figure 5.3 shows the processing induced power consumption of the three scenarios. The *x* axis represents the range of different values of task weights F multiplied by the variable U as shown in (5-5) (the objective function). This range is used to scale the number of connections to be comparable to the amount of consumed power.

Table 5-3 MILP Model Input Parameters

Parameter Description	value
Energy per bit consumed by the electronics of the transceiver (E^{elec})	50 nJ/bit [90]
Maximum download rate of each peer (objects and VMs) (L^{DO} and L^{DV}) respectively	10 kb/s, 25 kb/s [101,90]
Maximum upload rate of each (objects and VMs) (L^{UO}& L^{UV}) respectively	5 kb/s, 25 kb/s [90,91]
The processor capacity of object (ψ_j^O)	32 MHz [98]
The processor Capacity of VM(ψ_j^V)	1.2 GHz [100]
CPU maximum power consumption in objects (Ω_j^O)	347 mW [98]
CPU maximum power consumption in VM (Ω_j^V)	3.7 W [93]
Transmission amplifier power coefficient (ϵ)	255 pJ/(bit.m^2)[90]
The requested workload for each task (W_k), $k \in K$	0.01 GHz, 0.012 GHz, 0.015 Hz, 0.02 GHz, 0.05 GHz, 0.1 GHz, 0.2GHz, 0.3 GHz, 0.4 GHz, 0.5GHz [94]
Traffic generated by each task request (M_k), $k \in K$	250b/s, 500b/s, 750b/s, 1000b/s, 1250b/s, 1750b/s, 2000b/s, 2250b/s, 2500b/s, 2750b/s [100-102]

Traffic generated by each task result after reduction (C_k), $k \in K$	25b/s, 100b/s, 225b/s, 400b/s, 625b/s, 1050b/s, 1400b/s, 1800b/s, 2125b/s, 2475b/s
Maximum number of upload slots for each peer (X_i)	4 [104]
IoT nodes distribution area	30m × 30m [95]
Range of task weight (F) for all scenarios	{0, 0.1, 0.2, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8}
Scale factor with large value (A)	1000000

Figure 5.3 shows that the hybrid and VMs only scenarios consume the same amount of processing induced power at task weight values in the range (*F*=0 ~ 0.9) as there are no tasks executed by the objects in the hybrid scenario at these values as shown in Figure 5.4. The inefficiency of the objects-processors used and the effect of the power optimisation at such low scale factor result in blocking the requested tasks instead of implementing them by the objects. The power inefficiency of a processor used in object processing only is clearly illustrated in Figure 5.5 at task weight values (*F*= 0.2 and *F*= 0.3). At these values, the objects only scenario executes less tasks than the other two scenarios (about half), but it consumes more processing power than both of them as shown in Figure 5.3.

Starting at *F*=0.6 to the end of the range of task weights in Figure 5.3, the objects only scenario consumes the same amount of processing induced power. The low utilisation of the P2P layer in the objects only scenario is attributed to two reasons. Firstly, the effect of the fairness constraint and secondly the most effective reason is the low capacity of the processors in the IoT objects. This low capacity is clearly seen in Table 5.4 as the objects in the objects only scenario drop tasks (5 to 10) as the workloads of these tasks are larger than the processor capacity of the objects (ψ_j^O) as shown in Table 5.3.

A general trend followed by both hybrid and the VMs only scenarios towards higher power consumption for higher task weights can be seen in Figure 5.3. Starting from task weight F=1.2 a small gap is observed between the two scenarios and this grows as the task weight increases. This gap is caused by the higher power consumption of the hybrid scenario compared to the VMs only scenario because of the internal processing of the objects in the hybrid scenario. Due to the limited number of upload slots available for each object, an object tends to process its request internally instead of using the free upload slots. Accordingly, the internal processing allows the objects to send more task requests with higher workloads to VMs' to be processed. Therefore, the VMs in the hybrid scenario consume more processing induced power than the VMs in the VMs only scenario as shown in Figure 5.6.

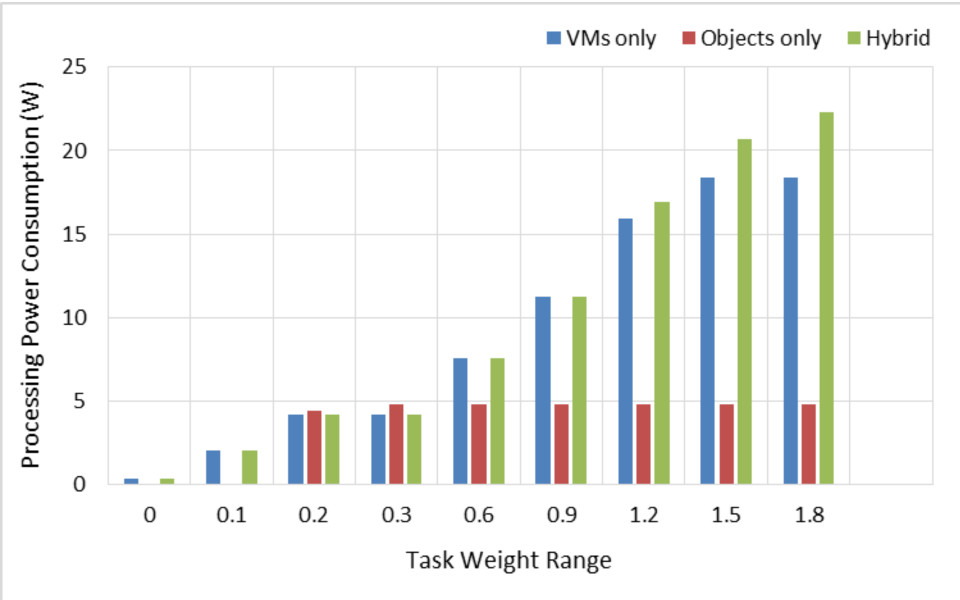

Figure 5.3 Total processing induced power consumption in the three scenarios

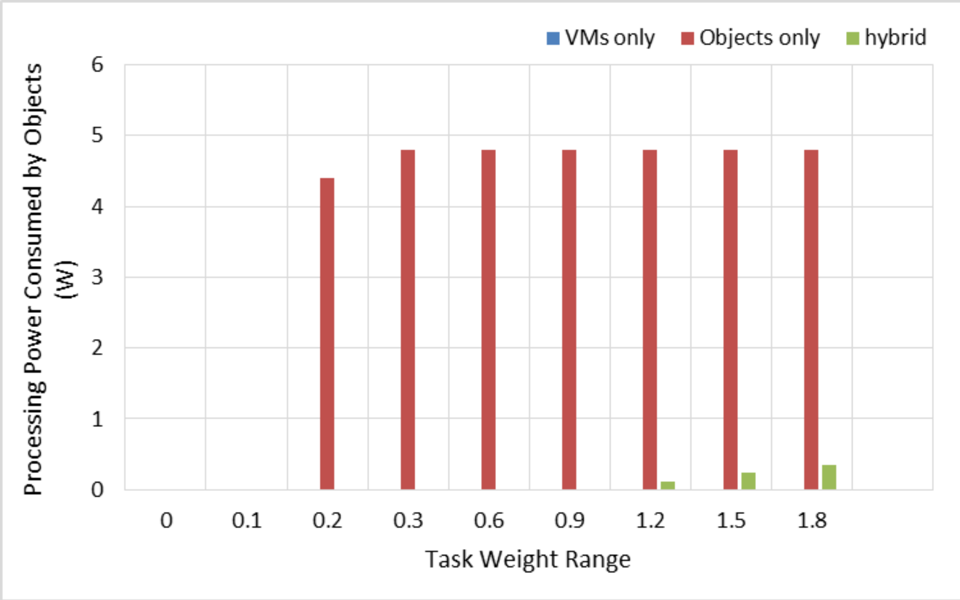

Figure 5.4 Processing induced power consumption by objects in the three scenarios

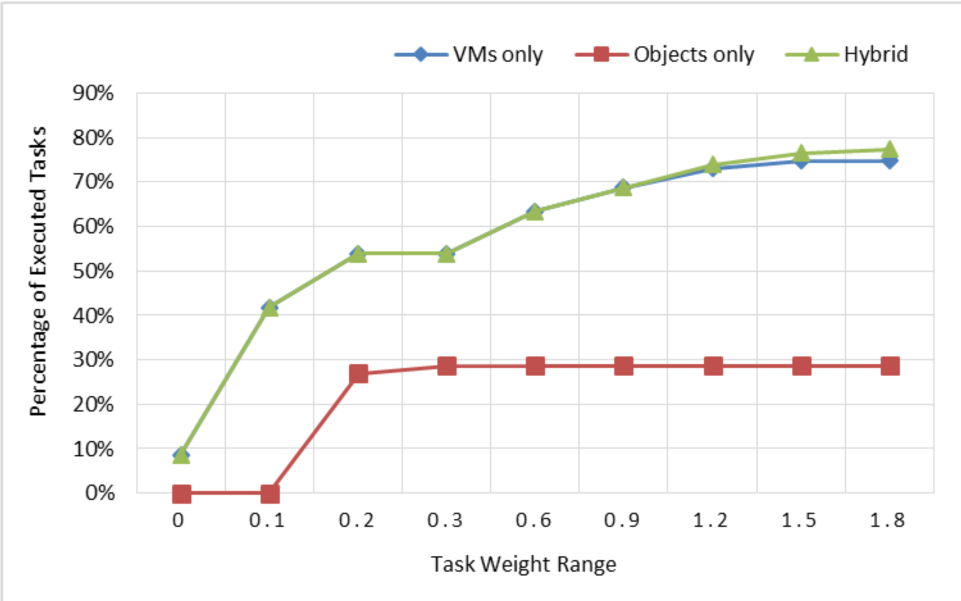

Figure 5.5 Percentage of executed tasks in the three scenarios

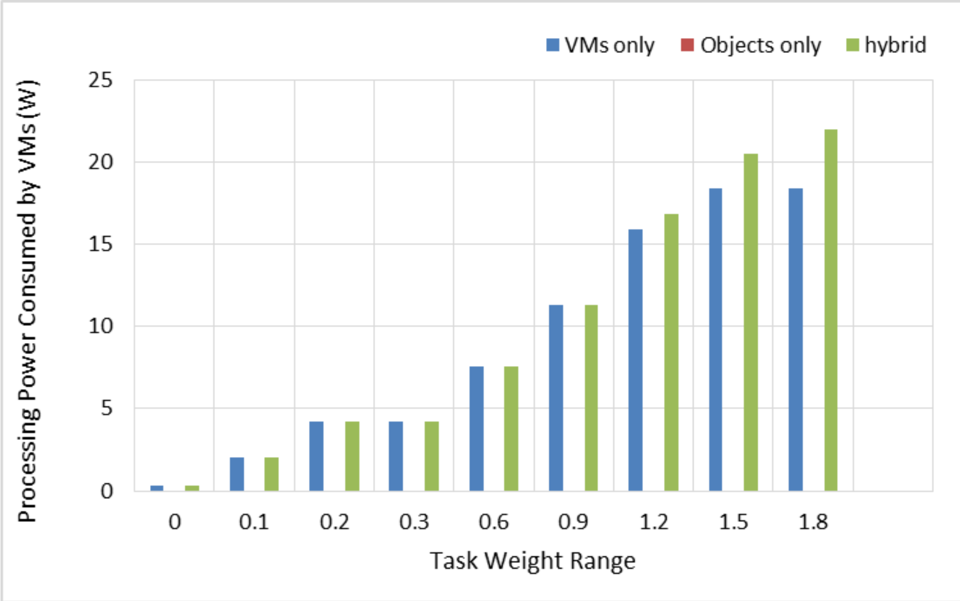

Figure 5.6 Processing induced power consumption by VMs in the three scenarios

To clarify, we consider task k_9 in Table 5.4 as an example. In the hybrid scenario, task k_9 is requested by objects no. 8, 15 and 25 and in the VMs only scenario are only requested by objects no. 15 and 25 but not by 8. This means that the request by object no. 8 is blocked. Therefore, by checking object no. 8, we notice:

1. Object 8 in the hybrid scenario processes internally task request k_2 and sends k_1, k_8 and k_9 task requests to other peers. The total generated traffic as a result of sending all these requests is 5000 b/s which is the maximum limit of the upload capacity of each object (the traffic generated by each task request is illustrated in Table 5.3).
2. Obviously, in the VMs only scenario, internal processing is not allowed, therefore object no. 8 sends requests k_1, k_2 and k_8 to VMs while task request k_9 is blocked. The total upload traffic due to requests is 3000 b/s which leaves only 2000 b/s of allowed traffic that can be uploaded by object 8. This (ie 2000 b/s) is not enough to transmit k_9 and that results in blocking this request instead of sending it to be served. In addition, blocking k_9 by object no. 8 in particular

is due to the power optimisation and its impact on the behaviour of the object. Since the object tries to send tasks with the lowest processor workload and lowest traffic demand requirements to be served by other peers, this results in blocking k_9.

Table 5-4 Tasks execution map at task weight (F=1.8)

Task ID	Total No. of Task Requests	Total No. of Served Tasks		
		Objects Only Scenario	VMs Only Scenario	Hybrid Scenario
k_1	15	12	15	15
k_2	10	6	10	10
k_3	15	10	15	15
k_4	8	5	8	8
k_5	14	0	14	14
k_6	11	0	11	11
k_7	9	0	6	6
k_8	13	0	5	6
k_9	11	0	2	3
k_{10}	9	0	0	1

As a result of the power optimisation, there is a general pattern followed by the objects in our network when they send their requests to be served by other peers. First, to make sure that the objects requests are satisfied using the lowest processing and network power consumption, objects search for the nearest available VMs starting with ones that are hosted by directly connected relays (objects' neighbours) then the circle of search is increased to include other relays starting from the nearest to the farthest. The implication is that the results in Figure 5.7 show that the traffic induced power consumed by relays is more than the power used by the objects. This difference increases with increase in the task weight in both hybrid and VMs only scenarios. In the hybrid scenario, when the model starts serving more tasks than the VMs can handle because of traffic and processing capacity constraints, the objects serve tasks using their own processors (internal processing) as shown in Figure 5.4. This starts at *F*=1.2 and continues beyond. Given that it is internal processing, it is of interest to understand the drivers behind the increase in the traffic induced power consumption in the relays. In this scenario, the internal processing affects the VMs behaviour resulting in serving more tasks with higher workload. Sending task requests with high workloads to VMs results in consuming more traffic induced power by relays. In the objects only scenario, the objects either serve task requests using their own resources if they able to, or send the requested tasks to other objects to be served. Sending tasks to other objects while satisfying the fairness constraint can lead to sending the requests to remote objects. This results in higher network power consumption in the relays. Consequently, the traffic induced power consumption of the relays in the objects only scenario is higher than the power consumption of the objects. It is even higher than the power consumption in the relays in other scenarios as illustrated in Figure 5.7 at task weight value *F*=0.3. However, as discussed earlier, the low capacity of the processor used in IoT objects results in low and constant serving tasks rate for other values of task weight range. This leads in turn to a constant consumption of traffic induced power for all devices in our network in the objects only scenario.

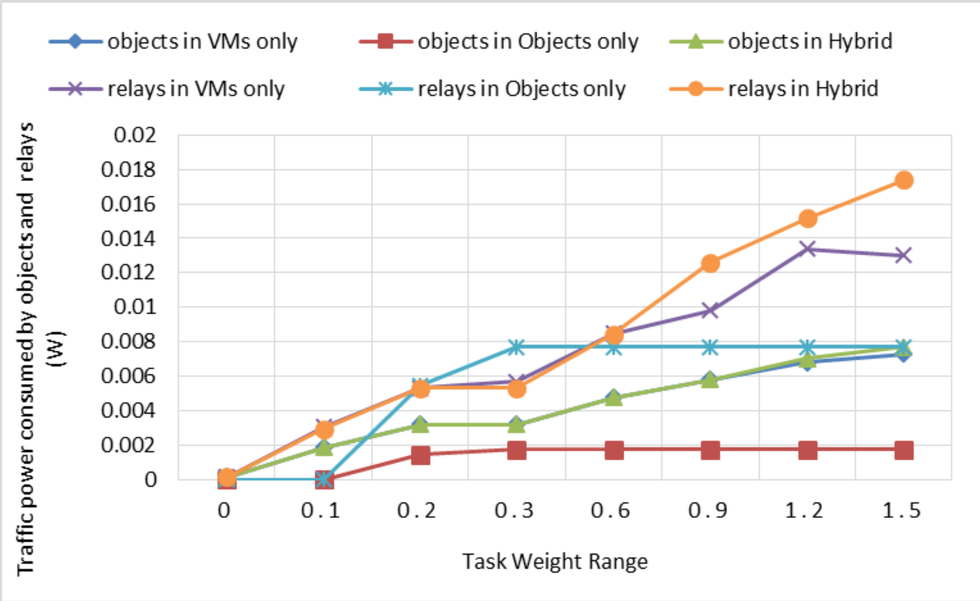

Figure 5.7 Traffic induced power consumption by objects and relays in the three scenarios

5.4 (EEVIPN) Heuristic and Results

In this section we try to mimic the behaviour of our energy efficient P2P IoT MILP model in real time by developing a P2P energy efficient IoT task processing and traffic routing heuristic. The pseudo code of EEVIPN heuristic is listed in Figure 5.8. It considers the hybrid scenario as it is the generic scenario that can be used to build other scenarios such as the VMs only and the object only scenarios. To determine the total power consumption (TPC), the heuristic determines the type and the optimum place of the peer to be used to serve the processing tasks according to the serving constraints of each peer. The serving constraints can be summarised as follows:

i. The processing task should not have been served by any other peer before.

ii. The upload traffic of each candidate peer should not exceed the maximum limit.

iii. The download traffic of each candidate peer should not exceed the maximum limit.

iv. The upload slots of each object should not exceed the specified maximum number.

v. The number of candidate VMs should not exceed the specified maximum number of serving VMs.

vi. There should be sufficient processor capacity in each candidate peer to accommodate the processing task workload.

Recall that these are the general serving constraints and could be changed according to the type of the serving peer. For example, if the candidate peer is a VM then all the serving constraints should be considered. If the candidate peer is an object (not the task requester), constraint (v) will not be applied. For the internal processing, the heuristic should check constraints (i) and (vi) because the requested task is served by the requester object internally and as a result, there will be no external data processing neither traffic flow.

For each task requested by an object, the heuristic first checks all the candidate VMs hosted by relays in the network. Starting with VMs is an attempt by heuristic to mimic the MILP model behaviour at the lowest values of task weight, by looking for candidate VMs as serving peers. The heuristic first checks VMs due to the power efficiency of their processors compared to the power efficiency of the objects processors. It also checks VMs first due to their high ability to serve all types of requested processing tasks. The serving constraints of the first candidate VM are investigated by the heuristic. If all these constraints are met, then the link between the requester and the serving VM is set. The requested task is served and the processing power P_j^{vp} of each VM is calculated. The heuristic loops for the rest of the VMs for all requested tasks by all objects. It finally calculates all the processing induced power of all serving VMs. If the requested task is served by an object, there are two cases, the first case represents internal processing. In the second case, the object serves another object. In this case, the Tit-for-Tat constraint (the fairness constraint) should be applied to guarantee equal reciprocity between the two objects intend to serve each other. In both cases, if all serving constraints are met then the link between the requester and the serving object is set. The candidate object serves the requested processing task and the processing induced power consumed by object-processer P_j^{op} is calculated. After checking all the possible serving peers for all requested tasks by all requesting objects, the traffic induced power consumption of each object P_i^{otr} is calculated. In addition, the power consumption of each relay P_a^{rtr} caused by cross traffic between the requesting objects and the serving peers is calculated. The traffic induced power consumption of each relay is composed of two basic parts. The first represents the power consumption due to traffic flowing between relays. The heuristic tries to route the traffic between node *x* (the directly connected relay to the requesting object) and node *y* (the directly connected relay to the serving object or hosting the serving VM) by using a minimum hop algorithm in order to minimise the traffic induced power consumption of each relay. The other part of P_a^{rtr} is the network power consumption due to the traffic flowing between relays and the request generator and serving objects. Finally, the heuristic calculates the number of served tasks by all peers NST and the total power consumption TPC.

Inputs: $\mathbf{VM} = \{\mathbf{1} \ldots \boldsymbol{NVM}\}$

$\boldsymbol{O} = \{\mathbf{1} \ldots \boldsymbol{NO}\}$

$\boldsymbol{K} = \{\mathbf{1} \ldots \boldsymbol{NK}\}$

$\boldsymbol{R} = \{\mathbf{1} \ldots \boldsymbol{NR}\}$

Output: No. of Served Tasks

Total Power Consumption (TPC)

For each task k $\in K$ Do

  For each object requesting a task $i \in O$ Do

    For each candidate VM that can serve a requested task $j \in VM$ Do

      If all serving constraints are met Then

        U(i,j,k)=1

        Calculate P_j^{vp}

      End If

    End For

  End For

End For

For each task k $\in K$ Do

  For each object requesting a task $i \in O$ Do

    For each candidate object that can serve a task $j \in O$ Do

      Case $(i = j)$

        If all serving constraints are met Then

          U(i,j,k)=1

          Calculate P_j^{op}

        End If

      End Case

      Case $(i \neq j)$

        If all serving constraints are met Then

Do Tit for Tat

U(i,j,k)=1

Calculate P_j^{op}

End If

End Case

End For

End For

End For

For Each IoT object $i \in O$ Do

Calculate P_i^{otr}

End For

For Each relay sending and receiving traffic to and from other relays $(a \in R)$

Calculate P_a^{rtr} based on minimum hop path between node pair (x,y)

End For

For each relay receiving task requests from objects and sending task results to objects $(a \in R)$

Calculate P_a^{rtr}

End For

Calculate no. of served tasks

$$NST = \sum_{i \in O} \sum_{j \in P} \sum_{k \in K} U(i,j,k)$$

40.	Calculate total power consumption $$TPC = \sum_{j\in O} P_j^{op} + \sum_{j\in VM} P_j^{vp} + \sum_{i\in O} P_i^{otr} + \sum_{a\in R} P_a^{rtr}$$

Figure 5.8 EEVIPN heuristic pseudo code

Figure 5.9 presents the total power consumption of both MILP and EEVIPN heuristic versus the percentage number of served tasks. It is clearly seen that the power consumption of the MILP and EEVIPN heuristic are comparable. The highest percentage of served tasks that can be achieved is 77% by the hybrid scenario in the MILP model. Therefore, we do not show results beyond 80% of served tasks as these cases will consume the same amount of power. It should be noted that the hybrid scenario the MILP model consumes higher power than the heuristic when serving higher than 70% of the requested tasks because of the higher VMs utilisation as clearly shown in Figure 5.10. The higher utilisation of VMs results from the internal processing by the objects at higher percentage of tasks execution as mentioned before in the discussion of the results in Figure 5.3. There are no tasks served by the objects in the VMs only and hybrid scenarios in the heuristic as illustrated by Figure 5.11 (processing induced power by objects =0). In VMs only scenario as alluded earlier, the objects are not allowed to process any task in this scenario. In the hybrid scenario, tasks with small workloads are served by VMs as the heuristic starts task assignment with VMs. After that, the heuristic tends to assign the remaining tasks (unserved) to objects where the tasks have workload requirements higher than the objects capabilities. As such, objects are not exploited in this scenario. Moreover, this results in both the hybrid scenario and VMs only scenario (in heuristic) following the same behaviour in executing tasks. This results in the two scenarios consuming the same amount of power as clearly shown in Figure 5.9.

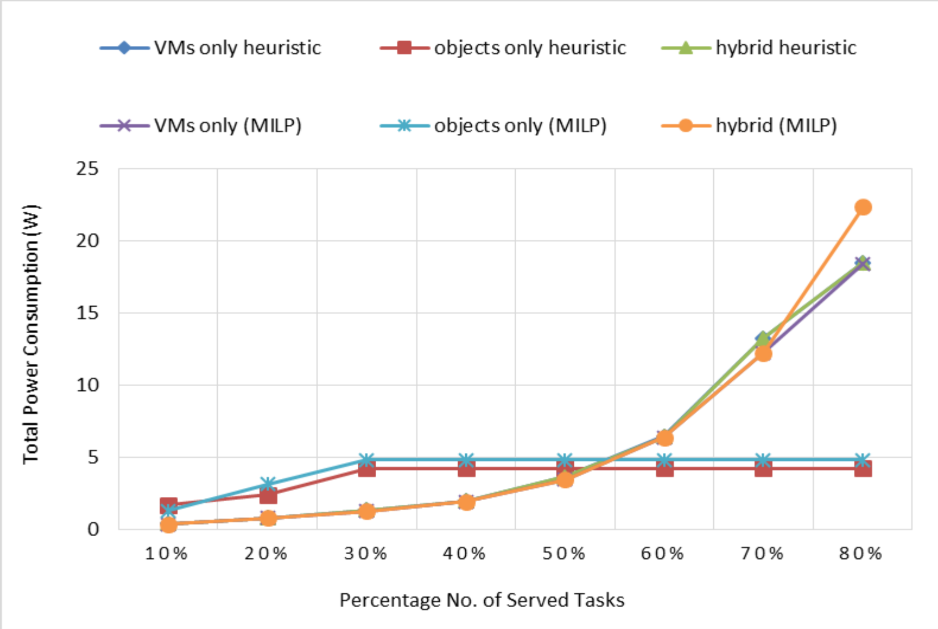

Figure 5.9 Total power consumption in EEVIPN heuristic and MILP model

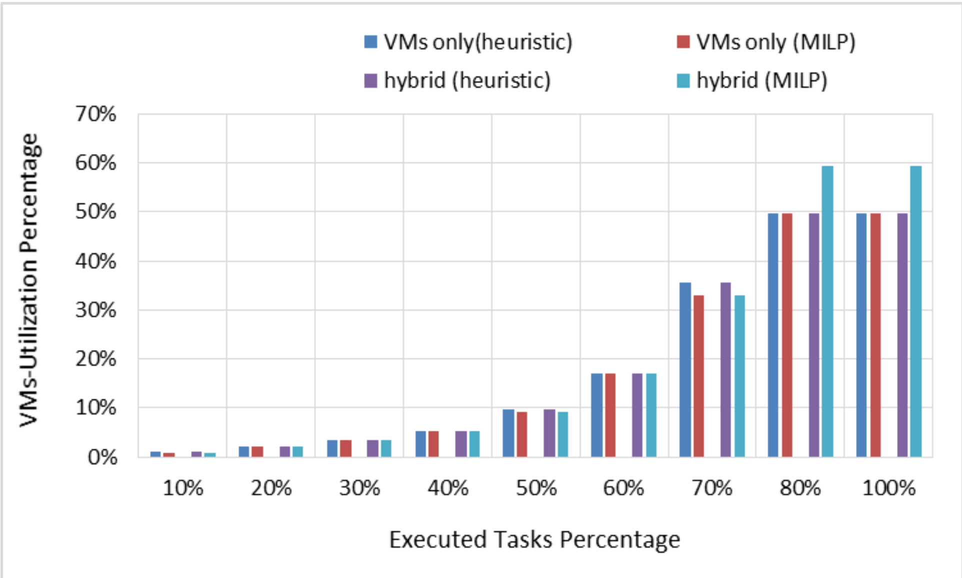

Figure 5.10 VMs- utilisation in hybrid and VMs only scenarios

Figure 5.9 shows that the objects only scenario (heuristic) consumes higher power than MILP model. This small difference is attributed to the impact of the network power consumption and specifically the power consumed by the relays as shown in Figure 5.12. In the MILP model (objects only scenario), if the tasks are not served internally by the objects then the model optimises the choice of the serving objects according to the fairness constraint in addition to the distances from the requesting objects to the serving objects in order to reduce the power consumption. In the heuristic, the search for serving objects is carried out sequentially regardless of their locations. This results in the relays consuming more power especially in cases where the tasks are sent to remote serving objects. A similar observation can be made about the difference between the power consumed by relays (due to traffic) in both hybrid and VMs only scenarios. In the heuristic, the relays consume higher traffic induced power than in the MILP. This is similar to the objects only scenario. It is also caused by sending the requests far apart in order to be served by the candidate serving VMs.

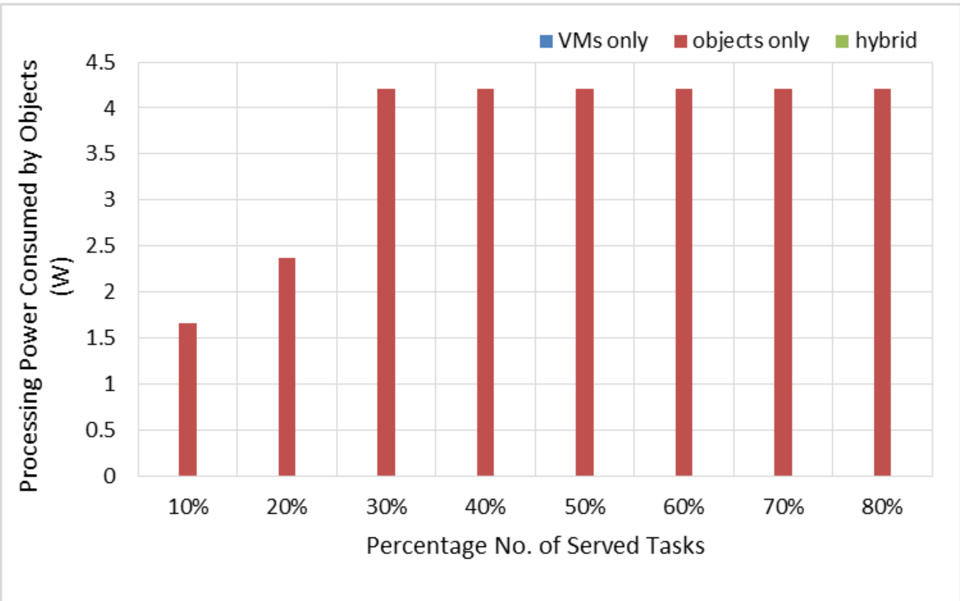

Figure 5.11 Processing induced power consumption of objects

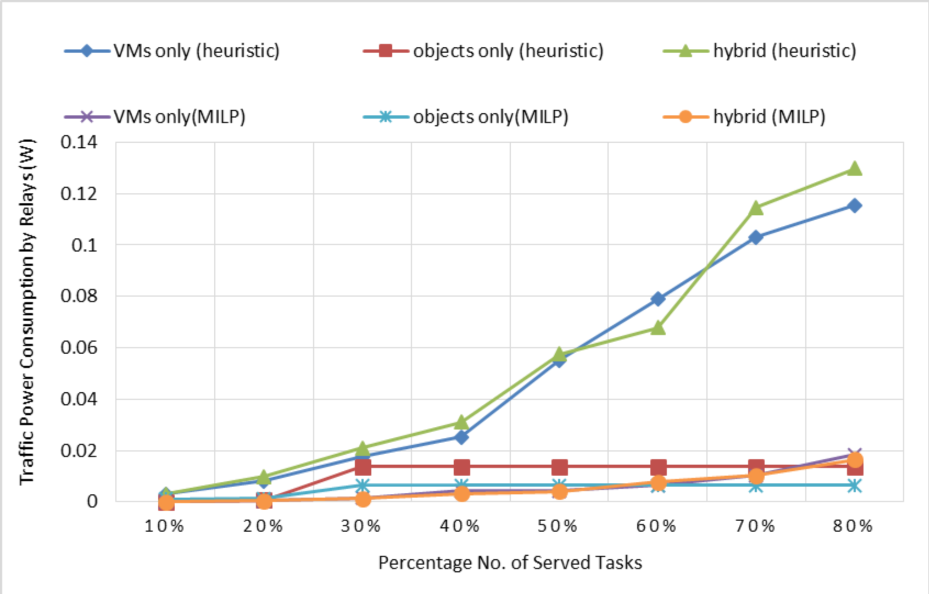

Figure 5.12 Traffic induced power consumption of relays

5.5 Summary

In this chapter, we have investigated the energy efficiency of an IoT virtualisation framework with P2P network and edge computing. This investigation has been carried out by considering three different VM placement scenarios. A MILP was developed to maximise the number of processing tasks served by VMs and minimise the total power consumption of the network.

Our results show that the hybrid scenario serves up to 77% (57% on average) processed task requests, but with higher energy consumption compared with other scenarios. The VMs only scenario can serve 74% (57% on average) of the processing task requests and 28% (22% on average) of task requests can be successfully handled by applying the objects only scenario. The results also revealed the low percentage of addressed task requests in the objects only scenario resulted from the capacity limit of the IoT objects' processors. In addition, the small difference between the serving percentage of hybrid scenario and VMs only scenario resulted from the allowed internal processing of objects in the hybrid scenario.

For real time implementation, we have developed EEVIPN heuristic based on the MILP model concepts. The heuristic achieved a comparable power efficiency and comparable number of executed tasks to the MILP model. The hybrid Scenario in the heuristic executes up to 74% of the total tasks (MILP 77%), up to 74% of tasks by the VMs only scenario (MILP 74%) while the objects only scenario executes up to 21% of the tasks (MILP 28%).

Chapter 6 : Conclusion and Future Work

This chapter summarises the work presented in this thesis. Furthermore, it suggests potential directions for future research on IoT systems.

6.1 Summary of Contributions

This thesis investigated the energy efficiency challenges in IoT and possible solutions, including distributed cloudlets for data processing, PON access networks and P2P systems. These solutions were evaluated through the development of MILP models that were behaviourally mimicked by heuristics for real time evaluation.

In Chapter 3, an energy efficient edge computing platform for IoT where the processing of IoT traffic was achieved by VMs hosted by cloudlets distributed over all the IoT network was investigated and the power savings of the proposed architecture were determined. First, a MILP model was developed with the objective of reducing the total power consumption induced by traffic and processing. The total number and location of cloudlets and VMs placement were optimised by investigating two scenarios: OPS and GPS. OPS optimally placed the VMs within the network elements of the proposed network (relays, coordinator and gateway), while GPS restricted the placement of VMs in the gateway only. The study in Chapter 3 considered a range of traffic processing reduction percentages caused by VMs processing of IoT data. The power consumption of OPS and GPS models were evaluated and compared. Four different types of VMs in terms of processing demands were evaluated for both scenarios. The results showed that one copy of each type of VMs can handle all the IoT object service-requests, as the VMs'

CPU utilisation does not depend on the number of served IoT objects. The results revealed that GPS consumed higher total power than OPS, because all traffic travels through more hops than in OPS to reach the destined cloudlet in the gateway. The results also show that OPS achieves 38% power saving compared to GPS.

Second, the impact of a limited number of served IoT objects by each VM copy was studied with respect to power consumption. While one copy of each VM type could serve all its objects, considering a capacitated VM with a lower number of served objects would result in generating more copies. The results showed that the highest power was consumed by the lowest number of served IoT objects per VM. This is because more VM copies were generated in this case.

Third, the dependency of VMs' CPU utilisation on the number of served IoT objects was considered and its influence on the power consumption was investigated. Accordingly, a MILP model considering blocking requests of IoT objects was developed to maximise the number of served IoT objects and minimise the total power consumption of all elements in the network. In this case, the results illustrated that GPS can save up to 47% of the power compared to OPS. However, GPS blocked 50% of the total IoT object requests and this led to a large reduction in power consumption while all IoT objects in OPS were satisfied. The results of this case were compared to the results of the first case, which considered the independency of VM CPU utilisation from the number of served IoT objects. This comparison proved that the first case consumed much less processing induced power than the second case.

Fourth, two heuristics were built to mimic the MILP model behaviour in real time considering GPs and OPS. Energy savings comparable to those of the MILP were achieved.

In Chapter 4, an energy efficient edge computing platform for IoT accompanied by a passive optical access network was introduced. A MILP model was developed to minimise the total power consumption in the proposed network. The optimisation of placement and number of cloudlets and VMs in addition to utilising energy efficient routes was studied under three scenarios of CPU demands and energy efficiency levels over a range of traffic reduction percentages. The results showed that power savings were achieved through packing most of the VMs in OLT at a low traffic reduction percentage and placing them in relays at high traffic reduction rate. In addition, the results indicated that utilising energy efficient PONs and serving heterogeneous VMs can save up to 19% of the total power. Finally, based on insights from the MILP model behaviour, a heuristic was developed for real time implementation, achieving comparable power savings of up to 17%.

In Chapter 5, an energy efficient IoT virtualisation framework with a P2P network and edge computing was proposed. A MILP model was developed with the objective of minimising the total power consumption and maximising the number of served tasks by the system. In this MILP model, the impact of VMs placement, fairness constraint, uplink and downlink capacity, limitations of task number and processing capability were investigated. Three scenarios were considered: The first scenario was the VMs only scenario, where the task requests were processed using hosted VMs in relays only. The second scenario was the objects only scenario, where the task requests were processed using the IoT objects only. The last scenario was the hybrid scenario, where the task requests were processed using both IoT objects and VMs. The results showed that the hybrid scenario served up to 77% (57% on average) of total task requests, which resulted in consuming the highest amount of power compared to other scenarios. The VMs-only scenario yielded up to 74% (57% on average) of task requests, while the objects-only scenario served up to 28% (22% on average) of task requests. The results also revealed that the low capacity of IoT objects' processors was the main reason for the low percentage of addressed task requests in the objects-only scenario. In addition, the allowed internal processing of objects in the hybrid scenario caused the slight difference between it and the VMs-only scenario, in terms of serving percentage. Finally, a heuristic was developed to mimic the MILP model behaviour in a real time environment with comparable results in terms of energy efficiency and task processing. Heuristic results indicated that hybrid, VMs-only and objects-only scenarios served up to 74%, 74% and 21% of total required tasks, respectively.

Finally, by comparing the proposed architectures in our work, the architectures proposed in Chapters 3 & 5 are suitable for local IoT network implementation such as a companies while the architecture proposed in Chapter 4 is more beneficial for larger network implementations such as cities. The layered and centeralised architectures in Chapters 3 & 4 are less reliable than the architecture proposed in Chapter 4 (P2P based network) in terms of node failure which leads P2P IoT networks to be more preferable in industrial sectors and in factories.

6.2 Future Work

In this section, future research directions are proposed for the topic of energy-efficiency in IoT.

6.2.1 Energy Efficient IoT in 5G Era

The first possible extension of this work is considering 5G mobile networks as a communication platform for IoT. The road towards 5G-IoT systems are paved by the future requirements of IoT applications and the progressive development of 5G technology. One of the basic requirements of 5G-IoT systems is providing an energy efficient communication platform for IoT devices to cope with the frequent data transmission. An integrated architecture combining both the IoT and 5G can be implemented, while considering the virtualisation of the processing and network functions in the edge, access and core networks.

6.2.2 Energy Efficient Caching in IoT

In all the models presented in this thesis, only processing was considered. These models can be extended by considering caching techniques as well. Caching the popular reusable data close to the requester object or application will reduce latency and network traffic. In addition, since caching reduces the need for persistent connectivity between the data generator object and the requester object, caching is one of the proposed solutions for energy efficiency issue of IoT, because it permits more objects to enter sleep mode [2]. An energy efficient IoT architecture optimising the caching host placement can be considered. In IoT edge networks, caching near the object, the gateway in the core cloud could also be explored. In addition, shared caching between IoT clusters and considering content popularity distribution could also be investigated. Analysing power savings considering the optimum cache replacement strategies (refreshing cache content) is also worth investigating.

6.2.3 Energy Harvesting in IoT

Mobile phones and personal computers are not the only IoT objects; IoT objects also refer to the billions of devices that interconnected wirelessly through the Internet. Billions of batteries with limited energy resource exist as a result of operating these devices. The proposed solution to prolonging the life of these batteries is energy harvesting by powering them with ambient energy resources such as heat or solar power. The impact of the virtualised processing and the distributed caching on energy harvesting of IoT objects could be explored.

6.2.4 IP/WDM core network for IoT

Meeting the requirements generated from diverse IoT applications as well as the need for storage of big data generated from the billions of IoT objects results in a great deal of attention focused on data centre networks form both academic and industrial sectors. The highest layer comprised of the network architecture investigated in this work is the access network represented by PON. With IP/WDM network deployment, an amalgam of IoT network architectures can be connected together over a wide area to cover many cities and towns which is another dimension that needs to be evaluated.

6.2.5 Extensions based on considering more metrics

One of the possible future directions is adding more metrics to the current objective function, such as mobility, cost and latency. Since the work presented in this thesis considered fixed IoT devices, investigating mobile IoT devices such as smart cars and its impact on the power consumption of the network elements can be considered. The high deployment of IoT networks all over the world leads to make the goal of low cost implementation one of main research trends. The design of energy efficient IoT architectures considering network implementation cost can be evaluated. Some latency-sensitive applications such some health care applications have very low latency requirements. Investigating such requirements in the edge computing architectures proposed in this thesis with the energy constraint can be considered.

List of References

[1] N. Kord, H. Haghighi, "An energy-efficient approach for virtual machine placement in cloud based data centers", *Information and Knowledge Technology (IKT) 2013 5th Conference on*, pp. 44-49, 2013.

[2] A. Beloglazov, R. Buyya, "Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers", *Proc. 8th Int'l Workshop MGC*, pp. 1-6, 2010, [online] Available: http://portal.acm.org/citation.cfm?doid=1890799.1890803 [Last accessed 26 Jun. 2018].

[3] A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems", *Adv. Comput.*, vol. 82, pp. 47-111, 2011.

[4] Zhou, Wei, Yan Jia, Anni Peng, Yuqing Zhang, and Peng Liu, "The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," *IEEE Internet of Things Journal (early access)*, 2018.

[5] J. Pan , and J. McElhannon, "Future Edge Cloud and Edge Computing for Internet of Things Applications", in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439-449, Feb. 2018.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015.

[7] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IOT) technologies, applications and challenges," 2016 IEEE Smart Energy Grid Engineering (SEGE), Oshawa, ON, 2016, pp. 381-385.

[8] R. Khan, S. U. Khan, R. Zaheer and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," 10th International Conference on Frontiers of Information Tchnology, pp. 257-260, 2012.

[9] A.Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Distributed energy efficient clouds over core networks," *IEEE/OSA* j. Lightwave technology, Vol. 32, No. 7, pp. 1261-1281, 2014.

[10] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani,"Energy efficient virtual network embedding for cloud networks," *IEEE/OSA* j. Lightwave technology, Vol. 33, No. 9, pp. 1828-1849, 2015.

[11] A.Q. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "BitTorrent content distribution in optical networks," *IEEE/OSA* j. Lightwave technology, Vol. 32, No. 21, pp. 4209-4225, 2014.

[12] X. Dong, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Green IP over WDM networks with data centers," *IEEE/OSA* j. Lightwave technology, Vol. 29, No. 12, pp. 1861-1880, 2011.

[13] X. Dong, T. E. H. El-Gorashi and J. M. H. Elmirghani, "On the energy efficiency of physical topology design for IP over WDM networks," *IEEE/OSA* j. Lightwave technology, Vol. 30, No. 12, pp. 1931-1942, 2012.

[14] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," in *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854-864, Dec. 2016.

[15] W. Shi, S. Dustdar, "The Promise of Edge Computing", *Computer*, vol. 49, no. 5, pp. 78-81, 2016.

[16] K. Dolui, S. K. Datta, "Comparison of edge computing implementations: Fog computing cloudlet and mobile edge computing", *Proc. Global Internet Things Summit (GIoTS)*, pp. 1-6, Jun. 2017.

[17] Pattar, S, et al., "Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review and Future Directions", *IEEE Communications Surveys & Tutorials (early access)*, 2018.

[18] U. Urosevic and Z. Veljovic, "Distributed MIMO solutions for peer-to-peer communications in future wireless systems," *2016 24th Telecommunications Forum (TELFOR)*, Belgrade, 2016, pp. 1-4.

[19] Y. Zhang, "Technology Framework of the Internet of Things and its Application", *Electrical and Control Engineering (ICECE) 2011 International Conference*, IEEE, pp. 4109-4112, 2011.

[20] A. R. Biswas, R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges", *Internet of Things (WF-IoT) 2014 IEEE World Forum on*, pp. 375-376, 2014.

[21] Dmitry Budko, "the most promising Internet of things trends for 2018", *Artificial Intelligence, Nov*, 2017. [Blog] . Available at: https://dashbouquet.com/blog/artificial-

intelligence/the-most-promising-internet-of-things-trends-for-2018 [Last accessed 26 Jun. 2018].

[22] Á. Asensio, Á. Marco, R. Blasco, R. Casas, "Protocol and architecture to bring things into Internet of Things", *Int. J. Distrib. Sensor Netw.*, vol. 2014, pp. 1-18, 2014.

[23] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, H. Y. Du, "Research on the architecture of Internet of Things," *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 5, pp. V5-484-V5-487, 20-22 Aug. 2010.

[24] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, pp. 1645-1660, 2013.

[25] V. Bhuvaneswari, R. Porkodi, "The Internet of Things (IoT) Applications and Communication Enabling Tehnology Standards: An Overview", *Proceedings of the International Conference on Intelligent Computing Applications*, pp. 324-329, 2014.

[26] C. Sarkar et al., "A Scalable Distributed Architecture Towards Unifying loT Applications", *Proc. IEEE World Forum on Internet of Things (WF-loT)*, pp. 508-13, Mar. 2014.

[27] B. P. Rao, P. Saluia, N. Sharma, A. Mittal, S. V. Sharma, "Cloud computing for internet of things & sensing based applications", *Proc. 2012 6th Int. Conf. Sens. Technol. (ICST)*, pp. 374-380.

[28]F. Jalali, K. Hinton, R. Ayre, T. Alpcan and R. S. Tucker, "Fog computing may help to save energy in cloud computing," in *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728-1739, May 2016.

[29]F. Jalali, A. Vishwanath, J. de Hoog and F. Suits, "Interconnecting Fog computing and microgrids for greening IoT," *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia),* Melbourne, VIC, 2016, pp. 693-698.

[30] N. H. Kumar, P. Karthikeyan, B. Deeksha, T. Mohandas, "Enhanced Routing over Sleeping Nodes in 6LoWPAN Network", *Future Internet of Things and Cloud (FiCloud) 2014 International Conference on*, pp. 272-279, 2014.

[31] J. Huang, Y. Meng, X. Gong, Y. Liu, and Q. Duan, "A novel deployment scheme for green Internet of Things," *IEEE Internet Things J,* vol. 1, pp. 196-205, 2014.

[32] D. Puthal, B. P. S. Sahoo, S. Mishra, S. Swain, "Cloud computing features issues and challenges: A big picture", *Proc. Int. Conf. Comput. Intell. Netw.*, pp. 116-123, Jan. 2015.

[33] A. Gajbhiye and K. M. P. Shrivastva, “Cloud computing: Need, enabling technology, architecture, advantages and challenges,” in Confluence The Next Generation Information Technology Summit, pp. 1–7, Sept 2014.

[34] Y. Jadeja, K. Modi, "Cloud computing - concepts architecture and challenges", *Computing Electronics and Electrical Technologies (ICCEET) 2012 International Conference*, pp. 877-880, 2012.

[35] K. Parsi and M. Laharika, "A Comparative Study of Different Deployment Models in a Cloud," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, pp. 512-515, 2013.

[36] A. Bahga and V. Madisetti, *Cloud Computing: A Hands-On Approach*: CreateSpace, 2013.

[37] B. P. Rimal, E. Choi, I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems", *Fifth International Joint Conference on INC IMS and IDC*, pp. 44-51, 2009.

[38] R. K. Lomotey, R. Deters, "Architectural designs from mobile cloud computing to ubiquitous cloud computing—Survey", *Proc. IEEE World Congr. Services (SERVICES)*, pp. 418-425, 2014.

[39] P. You, Y. Peng, H. Gao, "Providing information services for wireless sensor neworks through cloud computing", *IEEE Conference Publications Asia-Pacific Services Computing Conference*, pp. 362-364, 2012.

[40] J. Baliga, R. Ayre, K. Hinton, R. S. Tucker, "Energy consumption in wired and wireless access networks", *IEEE Commun. Mag.*, vol. 49, pp. 70-77, Jun. 2011.

[41] R. Q. Shaddad, A. B. Mohammad, S. A. Al-Gailani, A. M. Al-hetar, M. A. Elmagzoub, "A survey on access technologies for broadband optical and wireless networks", *J. Netw. Comput. Appl.*, vol. 41, pp. 459-472, May 2014.

[42] F. Effenberger, "An Introduction to PON Technologies", *IEEE Commun. Mag.*, vol. 45, no. 3, pp. S17-S25, Mar. 2007.

[43] Selvarani, D. Roselin, and T. N. Ravi. "Comparative analysis of Wi-Fi and WiMAX", *2014 International Conference on Information Communication and Embedded Systems (ICICES)*, 2014.

[44] Mohamed, M. A., M. Zubair, and H. M. Ibrahim. "QoS Based IPTV Over Mobile WiMAX Networks", *Wireless Personal Communications* Vol. 96, no. 2, pp.1711-1740, Sept. 2017.

[45] J. Maes, C. J. Nuzman, "The past present and future of copper access", *Bell Labs Tech. J.*, vol. 20, pp. 1-10, Mar. 2015.

[46] S Gorshe, A Raghavan, T Starr, S Galli, "The Family of DSL Technologies", Wiley, 2014.

[47] A Kaur, B Kaur, K Singh, Kaur, "Next Generation Optical Access Networks: A Review", *International Conference on Advances in Emerging Technology (ICAET 2016).*

[48] N. Ansari, J. Zhang, "Media Access Control and Resource Allocation for Next Generation Passive Optical Networks", Springer, 2013.

[49] Varghese, Samuel. "Fabrication and characterization of all-fiber components for optical access networks", PhD Thesis, Cochin University of Science and Technology, 2008.

[50] Weinstein Stephen, Yuanqiu Luo, Ting Wang, "The Comsoc Guide to Passive Optical Network: Enhancing the Last Mile Access, " Wiley, 2012.

[51] F. Effenberger, T. El-Bawab, "Passive optical networks (PONs): past present and future", *Journal of Optical Switching and Networking*, no. 6, pp. 143-50, 2009.

[52] G. Kramer, "Ethernet Passive Optical Networks", New York: McGraw-Hill, 2005.

[53] H.S. Abbas, M.A. Gregory, "The next generation of passive optical networks: A review", *Journal of Network and Computer Applications*, vol. 67, pp. 53-74, 2016.

[54] R. van der Linden, "Adaptive modulation techniques for passive optical networks," PhD Thesis, Technische Universiteit Eindhoven, 2018.

[55] Z. Buyankhishig, "Subscriber Reconfigurable TDM PON Architectures," *2014 7th International Conference on Ubi-Media Computing and Workshops (UMEDIA)*, pp. 139-143, 2014.

[56] R. Hu Li, H. B. Li, M. Luo, Q. Yang, S. Yu, "Digital OFDM-PON Based on Delta-Sigma Modulation Employing Binary IM-DD Channels", *IEEE Photonics Journal*, vol. 9, no. 2, pp. 1-7, April 2017.

[57] Wu, B. and Zhang, Z., "Applications of Optical Orthogonal Frequency Division Multiplexing in Optical Access Network", *Procedia Environmental Sciences*, vol. 10, pp.1199-1204, Jan 2011.

[58] C.-H. Lee, W. V. Sorin, B. Y. Kim, "Fiber to the home using a PON infrastructure", *J. Lightw. Technol.*, vol. 24, no. 12, pp. 4568-4583, Dec. 2006.

[59] P. Vetter et al., "Study and Demonstration of Extensions to the Standard FSAN BPON", *Proc. Int. Symp. Services Local Access*, pp. 119-128, 2002.

[60] R. Yadav, "Passive-optical-network-(PON-) based converged access network [Invited]", *IEEE/OSA J. Opt. Commun. Netw.*, vol. 4, no. 11, pp. B124-B130, Nov. 2012.

[61] D. P. Shea, J. E. Mitchell, "Long-Reach Optical Access Technologies", *IEEE Network*, vol. 21, no. 5, pp. 5-11, September 2007.

[62] I. Cale, A. Salihovic, M. Ivekovic, "Gigabit Passive Optical Network - GPON", *International Conference on Information Technology Interfaces*, pp. 649-684, June 2007.

[63] Anis, Muhammad Irfan, Muhammad Shahbaz Qureshi, and Saad Zafar, "Demonstration of TWDM-PON Backward Compatibility with Conventional GPON", *Wireless Personal Communications*, vol. 95, no. 2, pp. 581-592, July 2017.

[64] Giacoumidis, Elias, "Adaptive optical OFDM for local and access networks", PhD thesis, Bangor University (Electronic Engineering), July 2011.

[65] G. Kramer, G. Pesavento, "Ethernet Passive Optical Network (EPON): Building a Next-Generation Optical Access Network", *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 66-73, Feb. 2002.

[66] M. L. Bonilla, F. R. Barbosa, and E. Moschim, "Techno-Economical Comparison Between GPON and EPON Networks", *Proceeding of ITU-T Kaleidoscope Academic Conference*, 2009.

[67] A. Dhaini, "Dynamic Wavelength and Bandwidth Allocation in Hybrid TDM/WDM EPON Networks", *IEEE/OSA J. Lightwave Tech.*, vol. 25, no. 1, pp. 277-286, Jan. 2007.

[68] Ahsan, M.S., Lee, M.S., Newaz, S.S. and Asif, S.M., "Migration to the next generation passive optical network", In *Computers and Information Technology, 2009. ICCIT'09. 12th International Conference on*, pp. 79-84, December 2009.

[69] N. Ghazisaidi, M. Scheutzow, M. Maier, "Survivability Analysis of Next-Generation Passive Optical Networks and Fiber-Wireless Access Networks", *IEEE Trans. Reliability*, vol. 60, no. 2, pp. 479-92, June 2011.

[70] J. Kani, F. Bourgart, A. Cui, A. Rafel, M. Campbell, R. Davey, S. Rodrigues, "Next-Generation PON—Part I: Technology Roadmap and General Requirements", *IEEE Commun. Mag.*, vol. 47, no. 11, 2009.

[71] B. Batagelj, V. Erzen, J. Iratnik, L. Naglic, V. Bagan, Y. Ignatov, and M. Antonenko, "Optical Access Network Migration from G-PON to 102 XG-PON", *The Third International Conference on Access Networks*, 2013

[72] B Pourebrahimi, K.L.M. Bertels, S. Vassiliadis, "A Survey of Peer-to- Peer Networks," *Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing, ProRisc 2005*, November 2005.

[73] S. Androutsellis-Theotokis, D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies", *ACM Computing Surveys*, vol. 36, no. 4, pp. 335-371, 2004.

[74] Y. Liu, Y. Guo, C. Liang, "A survey on peer-to-peer video streaming systems", *J. Peer-to-Peer Netw. Applicat.*, vol. 1, no. 1, pp. 18-28, 2008.

[75] Q.H. Vu, M. Lupu, B.C. Ooi, Peer-To-Peer Computing: Principles and Applications, Nov. 2009.

[76] Chaudhary, Ms Neelkamal, "A survey on peer to peer system applications," *International Journal of Innovative Computer Science & Engineering*, vol. 2, No. 1, April 2015.

[77] R. Hayek, G. Raschia, P. Valduriez and N. Mouaddib, "Data Sharing in P2P Systems", Handbook of Peer-to-Peer Networking, Springer, pp. 531-569, 2010.

[78] Atip Asvanund, Clay Karen, Krishnan Ramayya, Smith Michael, "An Empirical Analysis of Network Externalities in Peer-To-Peer Music Sharing Networks", *Proceedings of the 23rd International Conference on Information Systems (ICIS)*, 2002.

[79] K. Wehrle, R. Steinmetz, "Peer-to-Peer systems and applications", *Distributed Hash Tables Notes in Computer Science*, vol. 3485, 2005.

[80] S. M. Thampi, C. K. Sekaran, "Survey of search and replication schemes in unstructured p2p networks," *Network Protocols and Algorithms*, Vol 2, no. 1, pp. 93-131, 2010.

[81] G. Sierksma, Linear and integer programming: theory and practice: CRC Press, 2001.

[82] D. M. Gay and B. W. Kernighan, "Ampl: A modeling language for mathematical programming," *Duxbury Press/Brooks/Cole*, vol. 2, 2002.

[83] T. Nowatzki, M. Ferris, K. Sankaralingam, C. Estan, N. Vaish, and D. Wood, "Optimization and Mathematical Modeling in Computer Architecture," *Synthesis Lectures on Computer Architecture*, vol. 8, pp. 1-144, 2013.

[84] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A mathematical programming language*: AT&T Bell Laboratories Murray Hill, NJ 07974, 1987.

[85] = Pióro, Michal, and Deep Medhi, “Routing, flow, and capacity design in communication and computer networks”. Elsevier, 2004.

[86] *Network Optimization*. Available: http://mat.gsia.cmu.edu/classes/QUANT/NOTES/chap11.pdf [Last accessed 25 June 2018]

[87] J. Huang, Y. Meng, X. Gong, Y. Liu and Q. Duan, "A novel deployment scheme for green Internet of Things," in *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 196-205, April 2014.

[88] Y. Sung, S. Lee, and M. Lee, "A Multi-Hop Clustering Mechanism for Scalable IoT Networks, " *Sensors*, Vol. 18, No. 4, p. 961, Mar. 2018.

[89] W. Feng and J. M. H. Elmirghani, "Lifetime evaluation in energy-efficient rectangular ad hoc wireless networks," International J. Communication Systems. Vol. 23. No. 12, pp. 1500-1520. 2010.

[90] Frank Comeau, Nauman Aslam, "Analysis of LEACH Energy Parameters", Procedia Computer Science, Volume 5, 2011, pp. 933-938.

[91] Cisco 910 Industrial Router [online]. Available: https://www.cisco.com/c/en/us/support/routers/910-industrial-router/model.html

[92] Intel Atom Z510 specifications [online]. Available: http://www.cpu-world.com/CPUs/Atom/Intel-Atom%20Z510%20AC80566UC005DE.html [Last accessed: 11 Jun 2018]

[93] SENSORO Alpha Product Suite- Alpha Node-4AA [online]. Available: https://www.sensoro.com/static/node4aa_en.pdf [Last accessed: 11 Jun 2018]

[94] Ren, Zujie, et al., "Workload Characterization on a Cloud Platform: An Early Experience." *International Journal of Grid and Distributed Computing*, vol. 9. No. 6, 2016, pp. 259-268.

[95] P. Szczytowski, A. Khelil, and N. Suri, “DKM: Distributed k-connectivity maintenance in wireless sensor networks,” in Proc. IEEE 9th Annu. Conf. Wireless On-Demand Netw. Syst. Services (WONS), Courmayeur, Italy, Jan. 2012, pp. 83–90.

[96] S. Electric, "FTE7502 EPON Optical Network Unit (10G ONU) datasheet". [Online]. Available https://www.sumitomoelectric.com/cms/wp-content/uploads/2016/01/FTE7502-UA_12_17_15_r6.1.pdf. [Last accessed: 11 Jun 2018]

[97] L. Sumitomo Electric Industries, "FSU7100 Series OLT datasheet". [Online]. Available: https://www.sumitomoelectric.com/cms/wpcontent/uploads/2016/01/FSU7101_121515_r5.0.pdf [Last accessed: 11 Jun 2018]

[98] THOMAS LEXTRAIT, Arduino: Power Consumption Compared [online]. Available: https://tlextrait.svbtle.com/arduino-power-consumption-compared

[99] Arduino and Genuino products [online]. Available: https://www.arduino.cc/en/Main/ArduinoBoard101. [Last accessed: 11 Jun 2018]

[100] Raspberry Pi3 Model B Technical Specification [online]. http://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf [Last accessed: 11 Jun 2018]

[101] N.Javaid, S.Faisal, Z.A.Khan, D.Nayab, M.Zahid, "Measuring Fatigue of Soldiers in Wireless Body Area Sensor Networks", *8th IEEE International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA’13),* Compiegne, France,2013,pp. 227-231.

[102] W. Xu, W. Liang, X. Jia, Z. Xu, "Maximizing sensor lifetime in a rechargeable sensor network via partial energy charging on sensors", *Proc. 13th IEEE Int. Conf. Sensing Commun. Netw.*, 2016.

[103] C. Tung et al., "A mobility enabled inpatient monitoring system using a zigbee medical sensor network", *Sensors*, vol. 14, no. 2, pp. 2397-2416, 2014.

[104] A. Q. Lawey, T. El-Gorashi and J. M. H. Elmirghani, "Impact of peers behavior on the energy efficiency of BitTorrent over optical networks," *2012 14th International Conference on Transparent Optical Networks (ICTON)*, Coventry, 2012, pp. 1-8.