# Energy Minimised Placement of IoT Based Machine Learning Services over Cloud-Fog Networks

**Mohammed M Alenazi**

Submitted in accordance with the requirements for the degree of

*Doctor of Philosophy*

The University of Leeds

School of Electronic and Electrical Engineering

October 2022

## The Intellectual Property Statement

Candidate's and other authors' contributions to this work have been noted explicitly below. Where references to the work of others are mentioned in the thesis, the applicant affirms that proper credit has been given.

**The work in Chapter 4 is based on publications as follows:**

Yosuf, B.A., Mohamed, S.H., Alenazi, M.M., El-Gorashi, T.E. and Elmirghani, J.M.H., 2021, June. Energy-Efficient AI over a Virtualized Cloud Fog Network. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems* (pp. 328-334).

The candidate participated the introduction and model architecture to the paper.

Yosuf, B. A, and T. Elgorashi developed and reviewed and validated findings throughout the work.

Sanaa H. Mohamed supported in early design and development of the model.

Professor Elmirghani reviewed the paper, helped with the content flow, and suggested the concept of the paper.

Alenazi, M. M., Yosuf, B. A., Mohamed, S. H., El-Gorashi, T. E., & Elmirghani, J. M. (2021) "Energy-Efficient Distributed Machine Learning in Cloud Fog Networks" 7th IEEE World Forum on Internet of Things, WF-IoT 2021, New Orleans, LA, USA, June 14 - July 31, 2021.

The candidate developed and considered the optimisation of ML service placement to improve energy efficiency of distributed processing for resource intensive applications.

B. A. Yousef, S. H. Mohamed and T. Elgorashi reviewed and validated findings throughout the work.

Professor Elmirghani reviewed the paper, helped with the content flow, and suggested the concepts in the paper.

**The work in Chapter 5 is based on publications as follows:**

Alenazi, M. M., Yosuf, B. A., Mohamed, S. H., El-Gorashi, T. E., & Elmirghani, J.M.H. “Energy Efficient Placement of ML-Based Services in IoT Networks”, *accepted, IEEE International Mediterranean Conference on Communications and Networking (MeditCom), 5-8 September 2022, Athens, Greece.*

The candidate has developed and extended previous work in distributing ML services taking advantage of virtualisation in the framework to provide abstract services for deep neural networks (DNNs) by restricting virtual machines placement in the IoT layer and studying their impact on cloud fog.

B. A. Yousef Sanaa H. Mohamed and T. Elgorashi reviewed and validated findings throughout thework.

Professor Elmirghani reviewed the paper, helped with the content flow, and suggested the concept of the paper.

## Acknowledgements

First and foremost, Peace, mercy, and blessings of Allah, and I want to thank Allah for all the benefits and gifts he has bestowed upon me. Professor Jaafar Elmirghani, my supervisor, deserves special recognition for his leadership, direction, and patience during my PhD journey and his unwavering support and encouragement are greatly appreciated. I would also like to express my gratitude to Dr Taisir, my co-supervisor. I thank her for her help, patience, and guidance.

Also, through his unwavering daily support, help and advice, Dr Barzan has supported me to reach my objectives, so I want to express my gratitude and thanks to him and Dr Sanaa.

On the other hand, I want to express my appreciation for my wife Nawal for her unwavering love and dedication during my PhD's journey. Thank you for assisting and encouraging me in achieving my objectives. I want to express my gratitude to my parents, "my mother and father." I ask Allah to give them the best of health, happiness, and well-being.

I want to thank all my colleagues in the department. I had a great time with them, appreciate their partnership and constructive discussions and wish them all the best in their future lives.

## Abstract

Massive amounts of data are expected to be generated by the billions of objects that form the Internet of Things (IoT). A variety of automated IoT services will largely depend on the use of different Machine Learning (ML) algorithms. Traditionally, ML models are processed by centralised cloud data centres. In the context of IoT, sensory data are offloaded to the cloud via multiple networking hops via the access, metro, and core network layers. This approach will inevitably lead to excessive networking power consumption as well as Quality-of-Service (QoS) degradation such as increased latency. In this thesis, we propose a cloud fog network (CFN) architecture where processing takes place in IoT nodes and fog servers in addition to the cloud. We use virtualisation to abstract deep neural network algorithms into Virtual Service Requests (VSRs) to represent the multiple interconnected layers of a Neural Network (DNN). Using Mixed Integer Linear Programming (MILP), we design an optimisation model that embeds the DNN VSRs into the CFN in an energy efficient way. We examine the performance of the proposed solutions and draw comparisons to embedding the DNN VSRs in the cloud. Several scenarios have been investigated in this thesis. These include distributed versus centralised source nodes, imposing constraints to limit Virtual Machine (VM) , that serve Inference DNN models, allocation at local IoT devices and studying the performance of the proposed approach when requests are embedded in a non-pre-emptive settings. The results show that significant energy savings can be obtained by optimising the deep neural network layers embedding in the network. The power savings are up to a maximum of 91% (68% on average) and vary with the scenario considered.

# Table of Contents

## List of Abbreviations

| | |
|---|---|
| AP | Access Point |
| CPU | Central Processing Unit |
| DC | Data Centre |
| DSL | Digital Subscriber Line |
| GB | Gigabyte |
| Gbps | Gigabit Per Second |
| GPU | Graphical Processing Unit |
| IaaS | Infrastructure as a Service |
| IoT | Internet of Things |
| IP/WDM | Internet Protocol over Wavelength Division Multiplexing |
| LAN | Local Area Network |
| LTE | Long Term Evolution |
| Mbps | Megabits Per Second |
| MILP | Mixed Integer Linear Programming |
| NaaS | Network as a Service |
| OLT | Optical Line Terminal |
| ONU | Optical Network Unit |
| PaaS | Platform as a Service |
| PON | Passive Optical Network |

| | |
|---|---|
| PUE | Power Usage Effectiveness |
| QoS | Quality of Service |
| RFID | Radio Frequency Identification |
| SaaS | Software as a Service |
| W | Watt |
| FLOPS | Floating point operations per second |
| WDM | Wavelength Division Multiplexing |
| Wi-Fi | Wireless Fidelity |
| NN | Neural Network |
| ML | Machine Learning |
| DNN | Deep Neural Network |
| VM | Virtual Machine |
| VSR | Virtual Service Request |
| AI | Artificial Intelligence |

# List of Tables

# List of Figures

# Chapter 1: Introduction

## 1.1 Introduction

Machine learning (ML) is increasingly used in many fields such as medical applications, smart cities, and autonomous cars where the goal is to efficiently and accurately predict the output or take decisions in response to real-time input data [1].

Nowadays, massive amounts of data can be produced by distributed Internet-of-Things (IoT) devices [2]. Using the abundant IoT data, intelligent services can be provided at edge networks such as distributed detection, monitoring, and classification [3] using ML tools such as Deep Neural Networks (DNNs). Traditionally, due to computational complexity, ML algorithms used to be executed in centralised cloud data centres (CDCs). While it is evident that the usage of centralised data centres for ML has provided accuracy and high performance, nevertheless this is achieved at the cost of high energy consumption [4], [5]. Transferring input data to CDCs imposes networking overheads in terms of power consumption and delay and also raises privacy concerns as the data could be accessed for unauthorised purposes [6].

As ML algorithms increase in their computational complexity, their associated energy consumption becomes challenging. In the case of edge / fog computing, such challenges heighten because the edge devices are resource constrained as they operate on a limited energy budget [7].

In the literature, the energy efficiency of ML algorithms, specifically deep learning models was tackled on a number of levels [5], (i) improving the algorithms so that the number of multiplication-and-accumulations (MACs) is

minimised in the code, (ii) performing specialised optimisation at the hardware level e.g., using high-end Graphical Processing Units (GPUs), and Application-Specific Integrated Circuits (ASICs) (iii) distributing the hidden layers across heterogeneous processing resources offered by Cloud/Fog networks. The last approach, also known as DNN inference partitioning, have been proposed and its feasibility have been studied in [8], and [9]. Inference involves using pre-determined weights in the computation of outputs and the DNN can hence be successfully partitioned across several layers with a cost of networking requirement between its different layers. Despite these efforts, attention has not been given to the end-to-end power consumption minimisation when ML algorithms are placed anywhere in the processing and networking continuum between the edge device and the central data centre.

In this thesis we take the approach in (iii) by designing a cross-layer optimisation framework that efficiently allocates virtualised DNNs across heterogeneous layers of processing offered by a cloud fog network (CFN) architecture. We use virtualisation to abstract the DNN algorithms into Virtual Service Requests (VSRs) where each layer of the DNN algorithm is represented by a VM and interconnections between layers are represented as virtual links. It is worth to note that in this work we only consider the inference phase (i.e., using a trained DNN with determined weights) and not the training phase as training is typically more complex and might require powerful central processing. We focus on optimizing the assignment of the elements of the partitioned DNN inference model between the edge (i.e., the IoT layer) and the central cloud layer with the objective of minimizing the power consumption.

## 1.2 Research Objectives

The work presented in this thesis has the following objectives:

1. To abstract DNN algorithms as VSRs composed of VMs representing the layers of the DNN algorithm and virtual links representing the connections between the layers.
2. To optimise the embedding of the DNN VSRs in a distributed processing environment across IoT, fog and cloud layers with the objective of minimising the processing and networking power consumption.
3. To compare optimum embedding of DNN VSRs over a distributed processing architecture to embedding in the cloud.
4. To study the energy efficiency degradation due the limited ability of IoT nodes to host VMs, imposed by hardware/software limitations.
5. To study the energy efficiency implications of suboptimum utilisation of resources in a non-pre-emptive embedding setting (i.e., sequential embedding setting) in which newly arriving VSRs are embedded without interrupting existing VSR to maintain QoS.

## 1.3 Original Contributions

The work in this thesis has resulted in the following original contributions:

1. Proposed a cloud –fog network (CFN) architecture that spans the edge, access, metro, and core networks providing processing nodes to embed DNN VSRs.

2. Developed a MILP model that minimises the network and processing power consumption of the CFN architecture by optimising the embedding of the DNN VSRs.
3. Evaluated the energy savings obtained by optimising the embedding of DNN VSRs over the CFN architecture by comparing them to the power consumption resulting from embedding DNN VSRs in the cloud.
4. Evaluated the increase in power consumption resulting from limiting the embedding of DNN VSRs to the IoT and fog layers.
5. Developed a MILP model to study the impact of constraining the number of VMs an IoT node can host on the energy efficiency of the CFN architecture.
6. Developed a MILP model to study the embedding of DNN VSRs in a non-pre-emptive setting in which requests are embedded sequentially and compared non-pre-emptive embedding in terms of energy efficiency and VSRs acceptance ratio to a pre-emptive setting where existing VSRs are re-embedded as a new VSR arrives to ensure optimum use of resources.

## 1.4 List of Publications

The following publications resulted from the work presented in Chapter 3 - Chapter 5:

1. Yosuf, B.A., Mohamed, S.H., Alenazi, M.M., El-Gorashi, T.E. and Elmirghani, J.M.H., 2021, June. Energy-Efficient AI over a Virtualized Cloud Fog Network. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems* (pp. 328-334).

2. Alenazi, M. M., Yosuf, B. A., Mohamed, S. H., El-Gorashi, T. E., & Elmirghani, J.M.H. (2021) “Energy-Efficient Distributed Machine Learning in Cloud Fog Networks” 7th IEEE World Forum on Internet of Things, WF-IoT 2021, New Orleans, LA, USA, June 14 - July 31, 2021.
3. Alenazi, M. M., Yosuf, B. A., Mohamed, S. H., El-Gorashi, T. E., & Elmirghani, J.M.H. “Energy Efficient Placement of ML-Based Services in IoT Networks”, *IEEE International Mediterranean Conference on Communications and Networking (MeditCom*), 5-8 September 2022, Athens, Greece.

## 1.5 Thesis Organisation

Following Chapter 1, this thesis is organised as follows:

Chapter 2 provides an overview of the key technologies related to this thesis such as IoT, cloud computing, fog computing, deep neural networks (DNN).

Chapter 3 presents the CFN architecture and the MILP model that optimises the embedding of DNN VSRs. It evaluates the energy efficiency of the CFN architecture under two scenarios: 1) DNN VSRs with a single source node, and 2) DNN VSRs with multiple source nodes. Chapter 3 also compares embedding DNN VSRs over the CDN architecture to embedding over a fog architecture.

Chapter 4 examines the impact of limiting the number of VMs that can be processed by an IoT node at any given time compared to multiple VMs allocation in IoT nodes.

Chapter 5 presents a MILP model to sequentially embed DNN VSRs in a non-pre-emptive setting maintaining QoS of existing VSRs and compares different objective functions to achieve server centric, network centric and total power energy efficiency.

Chapter 6 summarises the contributions of this thesis and provides directions for future work.

# Chapter 2: Background

## 2.1 Introduction

This chapter reviews the topics related to this thesis including IoT, cloud computing, fog computing and deep neural networks. It provides the reader with the background required to understand the chapters presenting original work.

## 2.2 Internet of Things (IoT) Key Elements

The Internet of Things (IoT) is defined as a distributed network of physical objects with sensing, processing or actuation capabilities able to communicate with each other and with users [10]. In the past, most of the IoT applications aimed at passive data collection and monitoring. Through the coupling of sensors, actuators and machine learning (ML), IoT systems are capable of interacting with the physical world and performing sophisticated tasks in an automated and dynamic manner [11].

The uptake of the IoT is increasing at unprecedented levels across a wide variety of domains in our daily lives, primarily due to advances in technology and manufacturing with respect to reduction in cost, size, and power consumption of next-generation low-power radio transceivers and microcontrollers [12]. Cisco predicted in 2011 that the number of linked IoT items would reach 50 billion by 2020 [13], [14] outnumbering the 7.7 billion people on the planet. This prediction whilst not totally fulfilled (10 billion connected devices in 2020), the trend is continuing on the rise [15], [16].

The Internet of Things (IoT) will change the modern world and its industries. Smart meters, for example, will be used to improve utility control. Sensors and actuators will allow factory floor mechanisation. City surveillance cameras will be used to aid law enforcement agencies in preventing crimes before they occur using ML algorithms [17]. It is claimed that IoT-based application systems earned $4.8 trillion in revenue alone in 2012 [18]. Also, according to the McKinsey Research Institute, the Internet of Things will have an economic impact in 2025 of approximately 11.1 trillion dollars [19]. IoT is still in its early stages. Energy consumption is a concern with the increasing number of IoT devices [20].

The sheer number of IoT devices leads to the generation of massive amounts of data which is usually transported over multiple domains of the network towards the centralised cloud data center for processing to extract knowledge from the data [21]. The costly overhead of the transport network (energy usage, latency and monetary cost) created a need for processing the collected data closer to the IoT end-devices. This led to the introduction of fog computing concepts which can fill this void by complementing the cloud and extending its services to the edge of the network and even further into the IoT devices [22].

Technically, the Internet of Things is built on a number of key technology elements that are needed to allow IoT to fully function and provide benefits to the users [23]. These elements include the identification mechanism for the IoT and other devices, the sensing and actuators that collect data, the communication system that connects the IoT and other devices, the computational devices that process the collected data, the services offered

based on the data collected by the IoT devices and the semantics and useful knowledge extracted via these services. These elements are summarised in Figure 2.1 and outlined below:

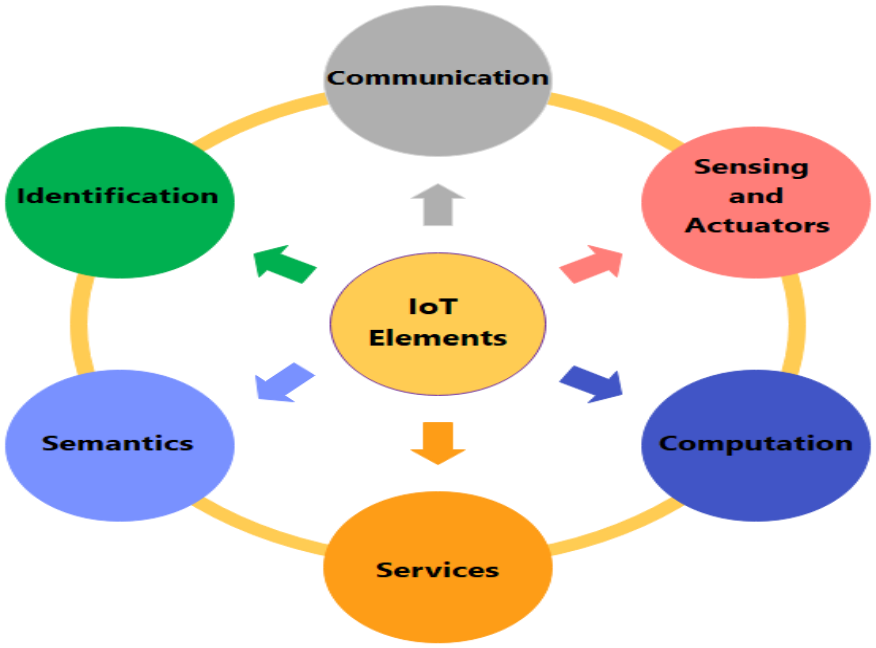



**Figure 2. 1:Key of Elements of IoT**

**Identification:**

Identification provides explicit identity for each object within the network [23]. There are two types of operations provided by identity: naming and addressing. When identifying IoT objects, it is necessary to distinguish between the object ID and the address. The object ID is the item's name, such as a temperature sensor, and the object's address is its location inside a communications network [24]. Usually there are multiple objects with the same name, but each object has its own unique address.

**Sensing and Actuators:**

Sensors and actuators are the primary physical components of IoT nodes. Sensors are physical detectors that read and gather data about the environment. The sensors are usually low-cost, low-power, and have limited processing capability and interfaces to communicate over defined communication channels [25]. There are many different types of sensors for perceiving the physical environment and measuring things like temperature, acceleration, vibration, light, electromagnetic characteristics, humidity, and the locations of physical objects [26]. The sensors typically monitor conditions and send signals when a change occurs (or can send continuous measurements), while actuators receive a signal and perform an action [27].

**Communication:**

Many communications technologies support IoT devices and smart services. Wi-Fi, Bluetooth [26], IEEE 802.15.4, and Long-Term cellular Evolution (LTE), 5G Ultra Reliable Low Latency (URLL) are examples of IoT communication protocols [28]. These communication technologies are different in terms of the data rates they can handle, the distances they can cover, and the amount of power they need [29]. In addition to the communication technologies, IoT devices require storage and battery power. Memory and storage technology, without a doubt, will continue to improve in the future [30]. In terms of battery power consumption, a lithium battery of 3v/225mAh, for example, can provide lifetime up to 3 years when performing task-based on RFID transmitters [31].

**Computation:**

Computation is required to process data obtained to make decisions [32], [33]. In IoT networks computation can be centralised or distributed:

In centralised computation and in the context of the work in this thesis, the IoT nodes are assumed to send the request for demands followed by the captured data to a specific cloud data centre location. As all IoT nodes rely on a single computing location, the computations are considered centralised.

In distributed computation, the IoT nodes can either process their own data or send a request followed by the data to one of multiple computing locations (e.g., in the cloud or fog layer) [33]. Thus, in the context of the work in this thesis, this form of workload allocation is considered a distributed computing architecture.

**Services:**

In general, IoT services have been implemented to support different functions in our daily lives. Building services, power and cooling services, safety services, industrial automation services, and so on have all benefited from the widespread usage of wireless communication [34].

**Semantics:**

Semantics in IoT is defined as the capacity to generate knowledge by deploying smart services utilising various methodologies. Identifying and using resources and modelling and analysing data are part of the knowledge

production process [35]. Based on these, the logical choice to deploy smart services is made.

## 2.3 Generic IoT Architecture

A general high-level reference design for IoT systems is proposed in the literature [36] - [41]. The reference design consists of various layers, as shown in Figure 2.2. These levels are briefly explained below:

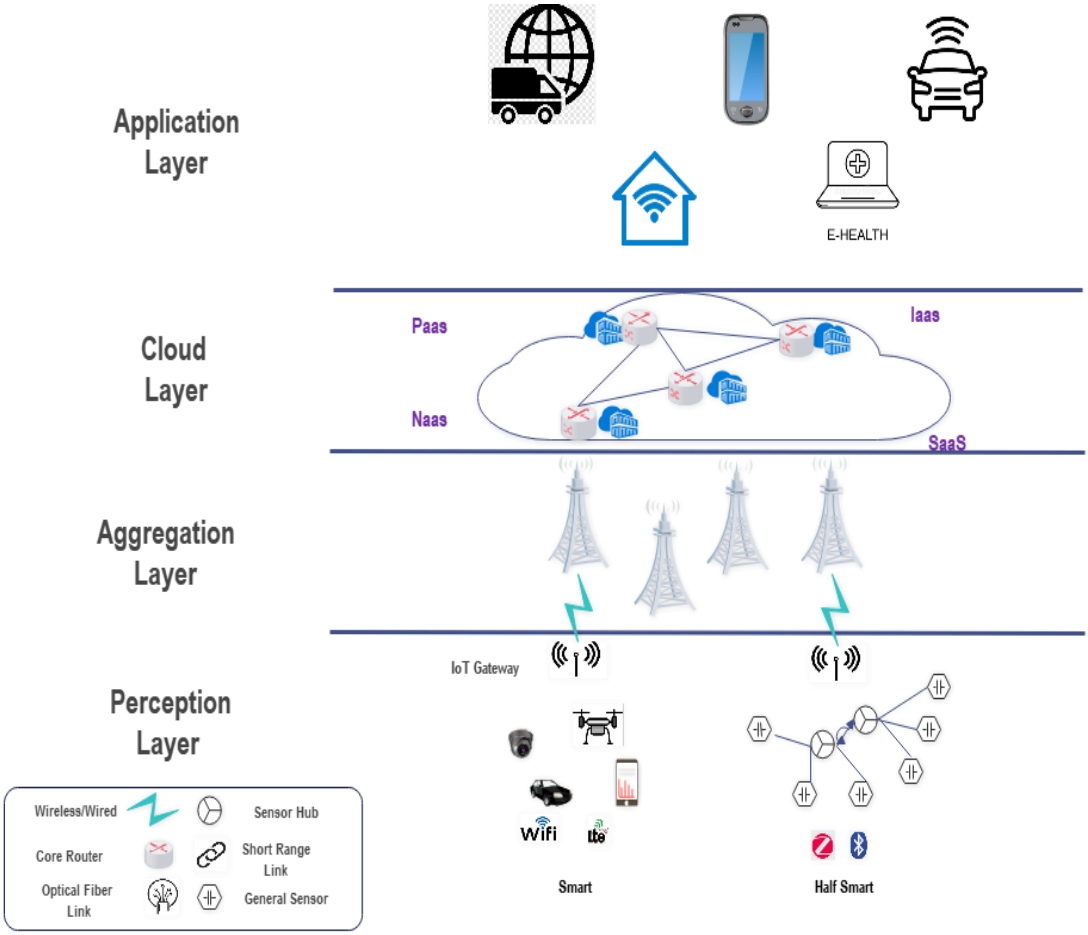



**Figure 2. 2: Generic IoT Architecture**

**Perception Layer**

The perception layer is the lowest in the IoT architecture. Its primary goal is to interpret raw data from objects in the environment. This layer is where all of the data collection and sensing takes place. Temperature sensors, cell phones, automobiles, drones, cameras, and other IoT items are examples of IoT objects [42], [43].

**Aggregation Layer**

The network layer offers the networking infrastructure for securely transmitting aggregated data from objects to the cloud for processing. Transmission can take place across wired and wireless networks. Communication technologies such as Wi-Fi, Bluetooth, ZigBee, LTE, and others are commonly employed [44], [45]. In most cases, an IoT gateway device is used to collect raw data from the perception layer resource-restricted devices (especially the less intelligent ones) [45].

**Cloud layer**

The cloud layer, also known as the middleware layer, gets massive amounts of data from the network layer [46]. This layer primary function is service administration and data storage. It has an analytical centre that processes aggregated data and makes automated decisions based on the analysis of the results, then feeds the output into the application layer. This layer enables data access and storage via cloud-based services such as infrastructure-as-

a-service (IaaS), platform-as-a-service (PaaS), software-as-a-service (SaaS), and network-as-a-service (NaaS) [39].

**Application layer**

The Application layer is at the top of the design and is responsible for presenting the final data to the end-user [47]. Hence, a key function of this layer is to provide a visual representation for the collected data from IoTs and a summary for the outcome or decision taken after processing the data. It gets the decision data from the cloud and, in exchange, provides management services for the applications that display the decision data or take action based on the decision data [46].

## 2.4 IoT Challenges

The abundant economic and societal advantages of IoT are confronted with several challenges that must be addressed. Some of these challenges are given in Figure 2.3 and briefly outlined below. These include Availability, Reliability, mobility, Trust, security and privacy and Energy efficiency.

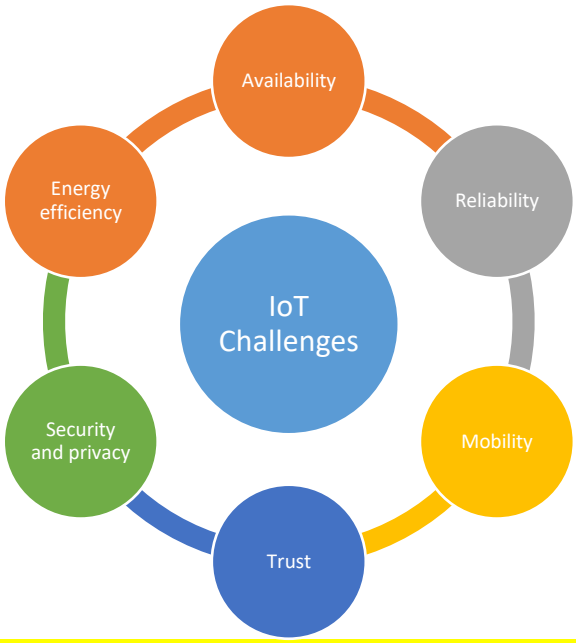



**Figure 2. 3: Challenges faced by IoT.**

## A. Availability

To realise anywhere and anytime services, IoT software and hardware must have high availability. Availability is defined as the percentage of time that the service or component is operating. It quantifies the loss in time resulting from service or component failure. IoT services must be reliable at the software and hardware level. For software, IoT applications should deliver services to users simultaneously anytime and anywhere. For hardware, a reliable IoT systems requires devices that are compatible with the IoT functionalities and protocols. Reliability in IoT system can be provided by redundancy for hardware (i.e., multiple available servers) and software (i.e., availability of multiple software packages within each server to ensure compatibility) [48]. The availability of IoT systems can be improved by evaluating the

availability requirements at the design stage when developing IoT systems to select suitable hardware and software [48], [49].

## B. Reliability

Reliability and availability are closely related. While availability measures the loss in time resulting from failure, reliability measure the frequency and impact of failure by quantifying the probability of the system meeting certain performance standards during a specified period of time. Meeting stringent reliability requirements is essential in designing IoT systems delivering critical services such as emergency services and healthcare services [50]. In addition to the reliability of the hardware and software delivering the IoT services, a resilient communication network is required to reliably connect users to the service. Reliability is required in all the layers of the IoT system as a failure in perception, data collection, computations, or networking can result in delays and data losses which can lead to the service failure [51]. A reliable transmission scheme to reduce packet loss in IoT systems is proposed in [50]. In [52], [53], a probabilistic model to evaluate the reliability and cost of IoT systems is developed.

## C. Mobility

IoT services will be mostly delivered to mobile users. Providing continuous service for these users is a challenge as services can be interrupted as users move from one gateway to the next. In [53], service

interruption was addressed by supporting caching to provide data in the case of temporary interruptions. Mobile Internet of Vehicles (IoV) formed as ad-hoc networks is specifically challenging. In [54], mechanisms for supporting mobility for vehicle-to-vehicle networking are discussed. The authors in [55], proposed a group mobility mechanism for IoV inspired by flying bird flocks.

### D. Trust, Security and Privacy

The Internet of Things faces significant challenges in security, privacy and trust. In addressing these challenges, the scalability of IoT systems of billions of devices and the heterogeneity of resources need to be considered [56]. As IoT-based systems and applications spread across various administrative domains, multiple stakeholders and ownerships are unavoidable. A trustworthy robust architecture is required to give users of the IoT system complete assurance that the services being offered is dependable [57]. Specific procedures and mechanisms are required to guarantee that IoT services are not interrupted or compromised by cyber-attacks [56]. IoT employs many types of identifying technology, such as RFID tags that may be attached to things and from which people's whereabouts may be deduced. It is critical to have proper systems to avoid the inference of personal information and allow IoT users who prefer to remain anonymous to do so. One strategy to secure personal information is to retain data as close to home as feasible by utilising decentralised processing and key management [58]. The authors in [59] optimized cooperative task

execution for resource-constraint IoT by utilizing a double-auction mechanism that finds the optimal policy for task execution with minimal need for private information about the nodes.

## E. Energy Efficiency

Energy efficiency is crucial for IoT systems. The energy efficiency of devices and communication interfaces determine the lifetime of the IoT systems. In many IoT systems, the devices are typically either battery powered or powered by renewable energy or energy harvesting. Hence, the scarce energy sources need to be efficiently managed to prolong the lifetime of the service supported by the IoT nodes and avoid service disruption. [60], [61], [62]. Relay nodes are introduced to IoT networks to prolong the IoT system lifetime [63]. Also, discontinuous reception/transmission is used in [64] in IoT sensors to turn off their communication interfaces and go into sleep mode.

Processing the massive amount of data created by IoT devices is challenging given the limited processing capacity and power sources of IoT nodes. Therefore, cloud intervention becomes necessary. However, sending the massive amount of data created by IoT devices to the cloud data centre is a huge burden on communication networks. Fog computing is an emerging computing paradigm that complements cloud computing by offering computational resources closer to the user at the access and metro networks. Fog computing offers energy efficient solutions to process IoT data.

## 2.5 Access Network Architecture for IoT systems

Traditionally, fixed-line access technologies such as copper-based xDSL, and cellular technologies such as 3G/LTE provide the final drop to end-devices. Given the massive predicted expansion in the number of connected IoT nodes and the requirement to reach the distant cloud for data processing, the technologies mentioned above can face challenges in meeting the IoT demand due to bandwidth constraints and energy inefficiencies [65]. Several significant developments have been made to address the concerns mentioned above, including integrating many heterogeneous access networks into a single platform as in 5G to allow seamless data interchange with the cloud. The combination of a wireless front end to provide connectivity for mobile/fixed nodes and fibre links to provide backhauling support ubiquitous services which are not achievable with wireless infrastructure alone [66]. Because of its high bandwidth, low cost, and point-to-multipoint design, Passive Optical Networks (PONs) have been the most appealing alternative for the backhaul in access networks providing high bandwidth in both the uplink and downlink [67].

## 2.6 Cloud Computing for IoT

The IoT layer can use the cloud for computing, data storage, and additional services based on network scale and application demand [68]. To address the issues connected with the massive data storage necessary and the computation and processing required, academics and stakeholders have boosted their efforts geared towards integrating cloud computing with IoT [60]. IoT devices are frequently supported by a local or global cloud infrastructure that increases their capabilities and provides extra services because of their

low cost, compact size, and low power consumption, as well as their restricted data processing, storage, and traffic handling capabilities [69]. Cloud plays critical roles in data storage, resource management, service creation, service management, service discovery, and power management, among other things [70]. When IoT and cloud are combined, a new paradigm emerges that can lead to IoT success in service provisioning, high-performance, dependability, ubiquity, and scalability. For fast and scalable service delivery, it can give cloud features with high elasticity and flexibility, [71], [72].

## 2.7 Fog Computing for IoT

Fog computing is a distributed processing paradigm that extends the abilities of the centralised cloud by bringing processing computational resources closer to the users as opposed to the centralised processing at the cloud [73]. Fog computing could be only a step away from the end devices. However, unlike cloud data centres that comprise thousands of powerful servers, in fog computing processing nodes are of limited resources [74].

As shown in Figure 2.4, fog computing can be represented in a hierarchy, mostly in three layers. The bottom layer comprises all IoT end devices, of limited computing, storage, and networking capabilities while the upper layers may comprise more powerful devices. Any device with communication, computational, and storage resources can be a fog node [75]. There are numerous potential fog nodes at the network edge. These nodes can release massive amounts of computing power as they are distributed across millions of devices, including routers, switches, gateways, smartphones, surveillance cameras, etc [39].

The IoT is an important source of big data since it relies on the connection of numerous intelligent devices to the internet that continuously reports the status of the physical environments. The focus in Internet of Things is not actually on the things in this thesis, but on the large amounts of data generated and hence the energy consumed used by devices. In this context, machine learning (ML) is a reliable technique for processing generated data into information and knowledge, predicting trends, gaining valuable insights, and driving automated decision-making processes. However, the integration of ML techniques in IoT faces several challenges, mostly in terms of the computational requirements they impose. Based on the application quality of service (QoS) parameters (such as response time) and the processing complexity needed, optimum processing of ML can occur at the IoT nodes, fog or cloud [76].

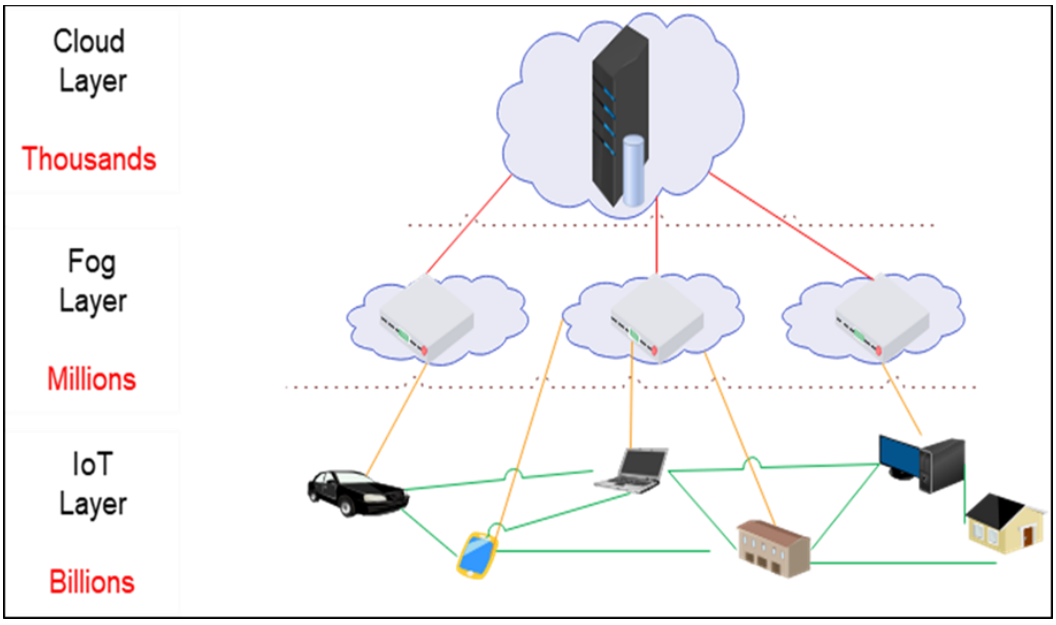



**Figure 2. 4: Fog Computing Architecture**

## 2.8 Deep Neural Networks (DNN)

Artificial Intelligence and neural network (NN) algorithms can enable many applications and services, especially in the IoT area where massive amounts of sensor data must be processed, identified, classified and acted upon [77]. DNN algorithms have surpassed human accuracy in numerous applications, however, this comes at the cost of high computational complexity. In the traditional approach, cloud datacentres alone were the main platform for processing NNs due to their abundant processing resources, which can be attractive, however, due to latency drawbacks and energy efficiency, it becomes necessary to evaluate other processing architectures [78].

A NN comprises of processing units referred to as neurons performing processing to revealing underlying patterns or connections within a dataset, much like the human brain making decisions. A Neuron is an activation function with many inputs and outputs as seen in Figure 2.5 [79]. The topology of a neural network is based on three layers: 1) input, 2) hidden, and 3) output. The first layer of the neural network processes the raw input data and relays that information to the second layer. The input layer nodes are all connected to each hidden layer nodes via links. The links are needed to establish communication and synchronisation between the layers. The data generated by the hidden nodes are fed into the output node(s) for decision based on the weight of edges and bias values of hidden nodes. The data processing of the hidden layer is mostly offloaded to a centralised cloud data center in which the input nodes' data is routed through the local gateway to data centers. Once knowledge is extracted from the processed data, the required output signal is

returned to back to the IoT local gateway and then this signal is used by the actuator devices.

The depth of the neural network is what distinguishes a single neural network from a deep neural network [78]. A neural network that contains more than three layers, including input and output, is considered a deep neural network [78]. Neural networks work with small sample size sets to do small tasks while deep neural networks not usually work with small samples size to generate significant insights [80]. To train neural networks (i.e., perform machine learning), three options are available which are supervised learning, unsupervised learning, and reinforcement learning [81].

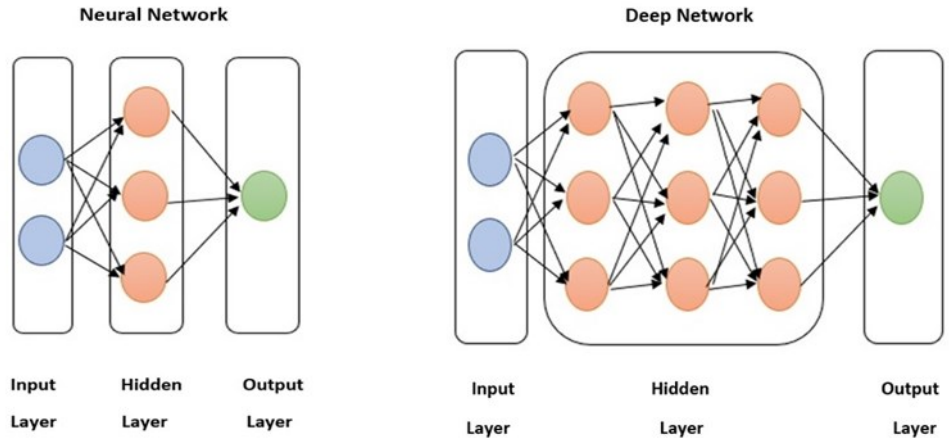



**Figure 2. 5: Neural Network VS Deep Neural Network**

## 2.9 Energy-Efficiency in communication networks and data processing

The performance of cloud and fog computing rely heavily on the transport or communication networks that link several network domains [82]. Those network domains include the core networks, the metro networks, and the access networks. Core networks contain mostly an IP over WDM nodes with

an optical networking layer and interfaces to IP equipment. Core networks typically connect cities and provide interfacing to cloud computing [83].

Metro networks typically have ring topologies and are the intermediate layer between the core and the access network. Finally, access networks, where most of fog computing resources reside, provide the connectivity to end users and IoT devices through various wired (e.g., PONs) and wireless interfaces [72], [84].

Optimising the design, protocols, and various workload assignment in different layers of the communication networks that provide connectivity for cloud and fog computing have been addressed by several studies and proposals in the last decade. The following subsections provide a summarised categorisation of these studies:

### 2.9.1 Optimising the design of IP over WDM core networks and optical networks

The authors in [85] studied the optimisation of the physical topology of IP over WDM networks under nodal degree constraints and under symmetric full-mesh and asymmetric traffic demands. An investigation on a full-mesh topology and a star topology showed that these topologies achieved 95% and 92% energy efficiency improvement compared to the NSFNET topology. In [86], the physical topology of IP over WDM networks was optimised while also considering the embodied energy (i.e., the energy required to manufacture and transport the components and equipment) and the operational energy. The results show that when considering the embodied energy and the

operational energy, embodied energy savings by up to 59% were achieved compared to the case of considering the operational energy only.

The work in [87] addressed the optimisation of the location of a data centre or multiple data centres in IP over WDM networks to reduce the power consumption of the networking required to support uplink and downlink requests by end-users. The study considered the network topology, number of data centres, traffic profiles, rate of uplinks and downlinks, and the impact of the power consumption minimisation on the delay. This work also considered the use of optical bypass, where the IP routers in the core nodes are used only in the source and destination nodes.

The results of using the optical bypass were compared to the case of using an optical non-bypass technique, where the transformation of optical signals into electronic signals is done at each core node for processing the packets. Moreover, the replication of content in these data centres was optimised to minimise the power consumption while considering content popularity and power consumption of storage. An energy-delay optimal routing algorithm was proposed to maintain Quality of Service (QoS) and minimise the power consumption.

The authors in [88] presented a summary of energy-efficiency key studies for core networks carried out as part of the GreenTouch consortium over five years. The key studies included the use of energy-efficient and improved components in core networks, putting devices in idle or sleep mode when not in use, optical bypass for intermediate core nodes, the use of optimised mixed line rates, putting protection lines into idle mode, optimising the network

topology and optimising the content distribution and the networking resources virtualisation.

The study provides a comprehensive MILP model that considered the aforementioned techniques with the objective of minimising the power consumption of future and current core networks. For the future networks, two scenarios were considered which are the business as usual (BAU) and the GreenTouch (GT) with BAU (i.e., BAU + GT). The former was based on a projection study for the power consumption expected for devices in 2020, while the later considers the projected power consumption values for 2020 devices, in addition to the techniques proposed as part of GT.

The results show that for the 2020 BAU scenario, the energy efficiency is improved by a factor of 4.23× compared to 2010 networks. For the 2020 BAU + GT scenario, the improvement in the energy efficiency compared to 2010 networks was found to be 315×. In [89], a validation study for the energy efficiency improvements was provided as part of the GreenMeter study carried by The GreenTouch consortium. The study set bounds on power consumption which can help in predicting the energy efficiency improvements in complex network structures.

The studies in [90], and [91] addressed the resilience of IP over WDM networks by considering and optimising the use of network coding. The protection path in a 1+1 survivable IP over WDM networks was encoded. Results based on MILP and on heuristics showed that encoding the protection path resulted in power consumption savings of 37% and 23% for ring and typical network topologies compared to the case of not using network coding. Moreover, the impact of varying traffic demands was considered. In addition,

an analytical study was provided to obtain bounds for the energy efficiency in the 1+1 survivable IP over WDM networks. This study validated the MILP models, heuristics and provided accurate estimation for large networks that are too complex for the MILP models and heuristics.

Optimising elastic optical networks was addressed in [92] where Orthogonal frequency-division multiplexing (OFDM) with adaptive rate and modulation format was proposed. A MILP model was proposed to minimise the power consumption of the rate and modulation adaptive OFDM based optical network. Under symmetric traffic, the results show that the OFDM-based network reduced the power consumption by 31% compared to conventional IP over WDM networks based on intensity modulation.

The work in [93], considered transparent optical networks and studied the energy consumption of these networks based on a cluster-based architecture where disjoint cluster are assigned different sleep cycles. The results showed that using anycast routing in these networks reduced the power consumption.

The energy consumption of optical burst switching-based network was addressed in [94]. The authors proposed a distributed algorithm that uses anycast routing while using sleep cycles. The results showed significant reduction in the power consumption without degradation in the QoS.

### 2.9.2 Optimising the use of renewable energy in optical networks.

In addition to the optical network design considerations discussed in the previous section, the use of renewable energy has been proposed and optimised to reduce the Greenhouse gas emissions (GHG) and $CO_2$ emission

associated with using non-renewable power resources, and hence provide positive environmental impact.

In [87], the authors utilised the MILP model for the IP over WDM networks with data centres to investigate if it is better to place data centres near renewable energy sources or to transmit renewable energy to the location of the data centres. Using the information about several wind farms and the expected transmission losses, the study optimised the locations of the data centres to minimise the overall power consumption. The results showed that considering the renewable energy in addition to optimising the data centre locations, using optical bypass, and replicating content, power savings of up to 73% were achieved compared to IP over WDM networks that do not use the considered combination of techniques.

In [95], the authors proposed solar and wind renewable energy use to reduce $CO_2$ emission of IP over WDM networks while also minimising their power consumption. Heuristic-based results showed that using optical bypass and renewable energy, $CO_2$ emissions were minimised by up to 52% with minimal impact on QoS. The placement of renewable energy in the nodes of the IP over WDM network was also addressed. The results showed that more reduction in $CO_2$ emissions is achieved when the renewable energy is placed near central nodes.

To maximise the use of solar renewable energy, the work in [72] proposed the use of Energy Storage Devices (ESDs) along with solar-powered fog data centres that cache content in the access network. A MILP model was developed to optimise the content caching from fog or cloud data centres while considering optical bypass and mixed-line rates in the IP over WDM networks,

the availability of the solar energy, and optimising the charge and discharge of the ESDs. The results showed that savings of up to 43% in the brown power consumption are achieved when using solar-powered fog data centres with 250 $m^2$ solar cells and 100 kWh ESDs.

### 2.9.3 Optimising workload assignment, virtual machine placement and content distribution for Internet applications

The authors in [83] addressed the design of energy-efficient IP over WDM networks for cloud computing services such as content delivery, Storage-as-a-Service (StaaS), and Virtual Machine (VM) placement. The authors compared centralised and distributed cloud computing and considered the impact of content popularity and access frequency on the placement of content in cloud data centres. Furthermore, the study considered factors such as the number of servers in each cloud data centre, the number of routers and switches for the data centre network, and the capacity of storage in each cloud data centre.

A MILP model was developed in [83] to optimise content delivery and the results indicated that replicating content on multiple cloud data centres resulted in 43% savings in the power consumption compared to centralised content delivery. A heuristic, DEER-CD, was developed to optimise content delivery and the results based on the heuristic were comparable to the MILP model. For StaaS applications the results showed that migrating the content based on its access frequency resulted in up to 48% networking power savings compared to centralised storage services.

Finally, a MILP model was developed to optimise the placement of VMs for processing-extensive applications. The results showed that slicing the VMs and placing them closer to their users achieved 25% savings in the power consumption compared to VM allocation in a single cloud location. For VM placement, a heuristic, DEER-VM, was proposed and the heuristic-based results were comparable to the MILP model.

The authors in [82], extended the work of VM placement in [83] and considered the placement in fog computing nodes in addition to cloud computing data centres. A comprehensive MILP model and heuristics were developed to provide energy- efficient placement of VMs in fog and cloud computing data centres. This study considered the impact of inter-VM traffic, the VMs workload, and the distance between the fog nodes and the users on the VM allocation results. It was found that under optimal placement for VMs with high traffic demands in the fog-cloud architecture, the total power consumption is reduced by 56% compared to distributed cloud only placement and by 64% compared to placement in existing cloud data centre locations in the AT&T network.

In [96], the authors focused on optimising IPTV content placement over IP over WDM networks. The study considered the TV viewing behaviour over 24 hours, and programs popularity in the UK to optimise the content caching in the IP over WDM core nodes. By placing the most popular content close to the end-users, significant savings in the networking power consumption can be achieved. Moreover, it was found that replacing the content in the caches according to the dynamics of programs viewing throughout the day improved the energy savings further.

A MILP model was developed to optimise the content replacement according to the TV program popularity, optimise the sizes of the caches and optimise the sleep-mode intervals so that the total power consumption of content caching over the IP over WDM network is minimised. The results showed that content replacement with variable cache sizes increased the hit ratio of the caches and reduced the power consumption by up to 86% compared to IPTV content delivery with no caching.

The work in [97] focused on the energy efficiency of peer-to-peer applications such as BitTorrent when used over IP over WDM networks. The energy efficiency of the original BitTorrent protocol was compared to a proposed energy efficient BitTorrent protocol over several IP over WDM networks with different node numbers and hop counts. It was found that smaller networks offer higher energy savings due to the ease of files localisation. Furthermore, a MILP model was proposed to optimise the location and upload rates of operator-controlled speeders (OCS) used to reduce the impact of leechers leaving after the download is complete. In addition to developing MILP and heuristic, experimental results were used to validate the results and the efficiency of the proposed energy efficient BitTorrent protocol.

The impact of network neutrality (i.e., treating Internet traffic equally without priorities) and its repeal was tackled in [98]. A techno-economical MILP model was developed to maximise the profit for Internet service providers, where services with varied qualities and prices were considered. The results revealed that repealing net neutrality can increase the Internet service providers profit by a factor of 8 if the pricing scheme discriminates against

data intensive content. This also results in a reduction in core network power consumption by 55% compared to the network neutrality service delivery.

The energy-efficiency of transporting big data in core networks was addressed in [99] and [100]. The authors proposed a progressive processing method in intermediate core nodes that results in reduction in the size of big data to be transmitted from data sources to cloud data centres. Results based on MILP models and heuristics showed savings by up to 52% when using the proposed method compared to classical big data processing where raw data is sent directly to the cloud data centre. The study considered the impact of the efficiency of the intermediate processing nodes compared to the cloud and the impact of the big data volume, variety and velocity on the power consumption savings.

In [101], the authors focused on the energy efficiency of placing workloads within data centre servers. The concept of data centre disaggregation was examined where the performance and the energy consumption of using disaggregated servers and regular servers were compared. Disaggregation fragments the memory, CPU, and network resources into pools instead of the single box solution provided by regular servers. The study also proposed using optical networking to link the resource pools. Results based on VM allocation indicated that data centre disaggregation can achieve up to 42% savings in the total power consumption compared to the use of conventional data centres.

In [102] and [103], Big data analytics and machine learning algorithms were used to provide priority-based e-healthcare services in 5G networks. Three machine learning algorithms were used to analyse historical medical records

and extract the likelihood of stroke in different patients. The authors proposed a MILP model to optimise the assignment of physical resource blocks (PRBs) to users according to the likelihood of strokes. The results showed an increase in the signal to noise and Interference (SINR) by 57% for high-risk patients.

The energy-efficiency of cloud-based real-time health monitoring applications was addressed in [104]. Fog computing was proposed to reduce the latency and high-power consumption of transporting health data to cloud data centres. The fog computing architecture utilised an energy efficient GPON access network. The authors proposed a MILP model to optimise the number and location of the fog devices and servers to serve the health monitoring applications. Power consumption savings by up to 52% were achieved for high data rate applications when using the fog computing system compared to using the cloud data centre.

### 2.9.4 Optimising virtual network embedding and services embedding in IP over WDM networks.

Virtual network embedding (VNE) is an important feature for future networks that can provide scalability and on-demand allocation of networking resources. End-to-end resources including network nodes and links are reserved for the application that require certain QoS guarantee over the required time window and then, these resources can be freed for further use by other requests or applications.

In [105], the authors provided a comprehensive study of the energy efficiency of virtual network embedding. An energy efficient VNE (i.e., EEVNE) approach was proposed which is based on a MILP model that optimises VNE

assignment in an IP over WDM network with data centres. The study compared EEVNE to two embedding approaches: a bandwidth cost VNE approach (CostVNE) that optimises the use of bandwidth resources and an energy-aware approach (VNE-EA) that only considers the energy-efficiency of VNE by switching off devices without taking into account the efficiencies of the devices.

The results showed that EEVNE achieved up to 60% savings in the power consumption compared to CostVNE when using energy inefficient data centres. A real-time heuristic named real-time energy optimised VNE (REOViNE) was proposed and compared to the MILP model and further evaluations under different data centre efficiencies were provided. Furthermore, the study addressed the impact of propagation delay and virtual node co-location constraints. Also, the trade-off between the profit and the energy efficiency was considered and results showed that both can be maximised concurrently.

In [106] and [107] the authors focused on the energy efficiency of service embedding for IoT applications. In this work, service embedding is regarded as a business process (BP) workflow. The BP is modelled as a virtual network with virtual nodes (i.e., requests for processing) and virtual links. A MILP model was proposed to optimise embedding the BPs while considering the details of IoT implementation in the access network. The objectives of the model were to minimise the power consumption, to minimise the latency, and to minimise both the power consumption and the latency with certain weights. The results show that when considering the reduction of the power consumption, savings by 42% were achieved compared to a system where all

requests are met without considering the power consumption. When considering the reduction of the latency, the results showed a reduction by 47% compared to a system that meets the demands without considering the latency reduction. In [107], the resilience of IoT service embedding was addressed by proposing traffic splitting.

## 2.10 Review of Mixed Integer Linear Programming (MILP) modelling

Linear programming is one of the most powerful mathematical optimisation methods that can describe a linear system and optimise certain aspects of this system (i.e., maximise or minimise a linear function describing revenue for example) while satisfying a number of constraints, in the form of linear equations, that link some parameters to some decision variables by a larger than or equal or less than or equal operation [108].

The solution to a LP problem is the outcome of the objective function in addition to the values that the decision variables take. Typically, all decision variables should be defined to be non-negative.

The general form of an LP can be as the following:

Let $x_1, x_2, \dots x_n$ be the decision variables and let $c_j$ where $j = 1, 2, \dots n$ be the objective parameters. Also, let $a_{ij}$ where $i = 1, 2, \dots n \; and \; j = 1,2, \dots n$ and $b_i$ where $i = 1, 2, \dots n$ be the constraints parameters.

Then the LP problem can be written as:

$$\min or \max c_1 x_1 + \cdots c_n x_n \tag{2.1}$$

Subject to:

$$a_{11}x_1 + \cdots + a_{1n}x_n (\leq or \ \geq) b_1 \quad (2.2)$$

$$a_{21}x_1 + \cdots + a_{2n}x_n (\leq or \ \geq) b_2 \quad (2.3)$$

$$\ldots\ldots$$

$$a_{n1}x_1 + \cdots + a_{nn}x_n (\leq or \ \geq) b_n \quad (2.4)$$

$$x_j \geq 0, \forall\, j = 1,2, \ldots n \quad (2.5)$$

In some optimisation problems, the solution under a set of nonlinear equations might be required such as a multiplication of two variable. To deal with these non-linear equations, some linearisation techniques can be adopted where the non-linear equation is replaced by a representative set of linear equations.

In LP problems as described above, the variables contain only real values. In most of optimisation problems, the use of some integer variables is needed for example to obtain a number of activated devices which should be positive integer. When real and integer variables are used in the problem, it can no longer be described as an LP problem and alternatively, the optimisation problem is described as a Mixed Integer Linear Programming (MILP) problem. Due to the need of obtaining some variables as positive integer, the solvers for MILP typically perform further steps such as branch-and-bound to ensure those variables are only integer.

MILP is widely used for a range of optimisation in transport network design, economic and operations research. MILP can be used to solve several networking optimisation problems.

For example, in a network design problem, the distribution of traffic from sources to destination can be optimised according to desired objectives such as minimising the number of hops. Such problem can be formulated using a

link-path formulation or a node-link formulation [109]. The node-link formulation is a more compact representation where the network can be described as a graph ($N$) where the nodes are connected via several links according to the topology.

To describe the traffic distribution in such a problem, a variable with indices for the source node ($s\ where\ s \in N$), destination node ($d\ where\ d \in N$), current intermediate node and next intermediate node can be used (i.e., $u$ and $v$ in the link $(u, v) where\ u \in N\ and\ v\ in\ N_u$ (the neighbouring nodes of node $u$)).

In the node-link formulation, a key constraint to be used is the flow conservation which states that the total traffic entering a node must leave this node with the same total value if it is an intermediate node between a source and a destination. If the node is source, only exiting traffic exist in the node and if the node is a destination node, only entering traffic exists.

If the objective in such a problem is to minimise the routing cost (i.e., number of hops), this problem can be categorised a multicommodity network flow problem where multiple demands are to be served at the same time while competing for available resources [109].

The general format of a MILP problem to describe multicommodity network flow problem consists of:

1- Defining the sets used (for example the nodes).
2- Defining the parameters and variables.
3- Defining the objective function.
4- Defining the flow conservation and the links capacity constraints.

A complete MILP model for this multicommodity network flow problem can be written as the following:

Sets and parameters:

$N$ Set of the nodes in the network.

$N_u$ Set of the neighbours of node $u$ where $u \in N$.

$R_{sd}$ The traffic demand between the source node and destination node $s, d \in N$.

$C_{uv}$ The capacity of link $(u, v), u, v \in N$.

Variables:

$X_{sd}^{uv}$ The traffic in link $(u, v), u, v \in N$ that is associated with the traffic demand between $s, d \in N$.

The objective is to minimise the routing cost:

$$\min \sum_{s,d,u,v \in N, s \neq d} X_{sd}^{uv} \tag{2.6}$$

Under the following link capacity and flow conservation constraints:

1- Flow conservation:

$$\sum_{v \in N_u} X_{sd}^{uv} - \sum_{v \in N_u} X_{sd}^{vu} = \begin{cases} R_{sd} & u = s \\ -R_{sd} & u = d \\ 0 & otherwise \end{cases}, \forall\, s, d \in N\,, u \in N. \tag{2.7}$$

2- Link capacity constraint:

$$\sum_{s,d \in N} X_{sd}^{uv} \leq C_{uv}, \forall\, u \in N\,, v \in N_u. \tag{2.8}$$

This MILP formulation can be a base for a variety of network and resource assignment optimisation problems. For example, the allocation of resources within a network can be optimised under further capacity constraints in addition to the link capacity constraints and the flow conservation routing constraints.

A set of additional devices such as routers, servers, IoT devices, various gateways and switches can be described and additional constraints related to different devices in different layers of the considered system can be included to customise the optimisation problem and address the needs and constraints of underlying study.

Solving MILP requires a solver and a tool to describe the parameters, variables, objective function and constraints. The Mathematical Programming Language (AMPL) [110] provides such tools and also provides a number of powerful solvers such as the CPLEX. Several options and settings for the CPLEX solver can be chosen to control reaching the optimal solutions. In the models developed for this work, CPLEX was sufficient to reach convergence and hence provide optimal solutions. In more complex MILP models, heuristics and ML tools such as reinforcement learning can be used to approximate the solutions [111].

Typically, the complexity of the MILP model is proportional to the number of variables and constraints used. Low-complexity MILP models can run in typical laptops and PCs, however, high-complexity MILP models or large models require supercomputing infrastructure.

## 2.11 Summary

This chapter provided a high-level review of IoT, cloud computing, fog computing, DNN, energy-efficiency in communication networks and MILP to give the reader a background to facilitate reading the subsequent chapters. Inference models for DNNs (i.e, trained neural networks with multiple hidden layers) explained in this chapter, are abstracted in the next chapter using Virtual Service Requests (VSRs) to represent the different layers of the DNN (i.e., input layer, hidden layers and output layer.

# Chapter 3: Energy Efficient AI over a Virtualised Cloud Fog Network

## 3.1 Introduction

In this chapter we will study energy efficient embedding of DNNs into a cloud-fog network (CFN) architecture. We abstract the DNN algorithms as virtual service requests (VSRs) composed of multiple VMs connected by virtual links. Using Mixed Integer Linear Programming (MILP), we formulate the embedding of the DNN into the CFN architecture as an optimisation problem that minimises the total power consumption through trade-offs between processing and networking power consumption. We study the energy efficiency of the Cloud-Fog architecture by comparing it to a baseline approach in which the DNN VSRs are embedded in a cloud data centre (CDC). We study the embedding of DNN VSRs under two scenarios: In Scenario 1 a single IoT device is generating data for the input layer of the DNN algorithm while in Scenario 2 multiple IoT devices generate data for the input layer.

## 3.2 The Proposed Cloud Fog Architecture

Figure 3.1 shows the considered CFN architecture. It comprises of four networking layers which are edge network, access network, metro network, and core network. In the edge network, we consider distributed IoT devices in different zones. Some of these IoT devices are source nodes collecting data. For the access layer, we consider a Passive Optical Network (PON) in each IoT zone. The PON contains several Optical Network Units (ONUs) that connect with the IoT devices through Wi-Fi. Each zone is covered by an ONU.

The ONU aggregates the traffic from IoT devices into an Optical Line Terminal (OLT) via fibre links and a splitter/combiner. An Access Fog (AF) node containing several servers is connected to each OLT. The OLT connects to the metro network through a metro switch which connects to the core network via a metro edge router. A Metro Fog (MF) node composed of a set of servers is placed in the metro node. The core network we consider is an IP over WDM network [112] which has two layers, an IP layer and an optical layer.

The IP layer in each core node connects to the metro network and aggregates/disaggregates its uplink/downlink traffic and the optical layer performs electrical to optical / optical to electrical conversion and physically connects to other core nodes via optical fibre links. We consider a CDC connected to a core node that is one hop from the core node aggregating traffic from the metro and access networks that link the considered IoT devices. We assume the processing nodes in the CFN architecture depicted in Figure 3.1 are virtualised, i.e., VMs, where a number of networked VMs compose a VSR and a VSR compose a layer of the DNN, can be embedded in the architecture layers regardless of the hardware heterogeneity.

We consider multiple DNN topologies to be embedded into the processing nodes of the CFN architecture. A DNN topology is represented by VSRs, each of which is composed of multiple VMs. Each layer of the DNN is represented by a VSR and the VMs of a VSR are connected via virtual links as illustrated in Figure 3.2 thus forming a VSR topology. The input VM of the DNN VSR must be embedded in IoT source nodes where data is collected while the hidden layers can be embedded into any of the CFN layers. The output layer will be connected to the actuator implementing the decisions taken by the

DNN. However, the embedding of the hidden layers can be influenced by the geographical distribution of the IoT nodes. The networking power consumption incurred due to data transfer should be accounted for. Therefore, one may choose to place the hidden layers as close as possible to the source nodes where the input layer is embedded. Proximity to source nodes is also desirable to limit latency. In Figure 3.1, we exemplify how a VSR is mapped onto the physical resources in the CFN architecture.

Note that, the DNN topology considered, and its associated training algorithm represents an inference model not a training model as the latter will be executed in the cloud due to its high processing requirements (ie training in the cloud to determine the DNN weights, but real time operation at the edge of the network (our focus). Inference DNN models are pre-trained, hence they are not as computationally intensive as training models because the weights and biases have already been determined [113].

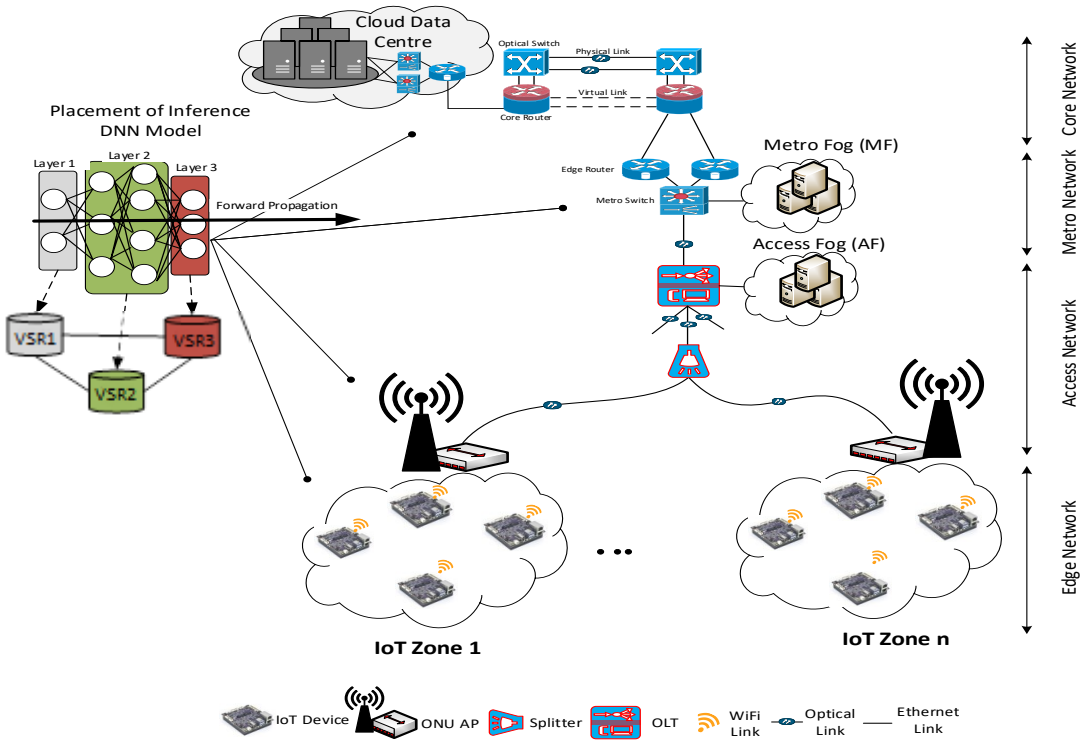



**Figure 3. 1: Architecture of Internet of Things (IoT)**

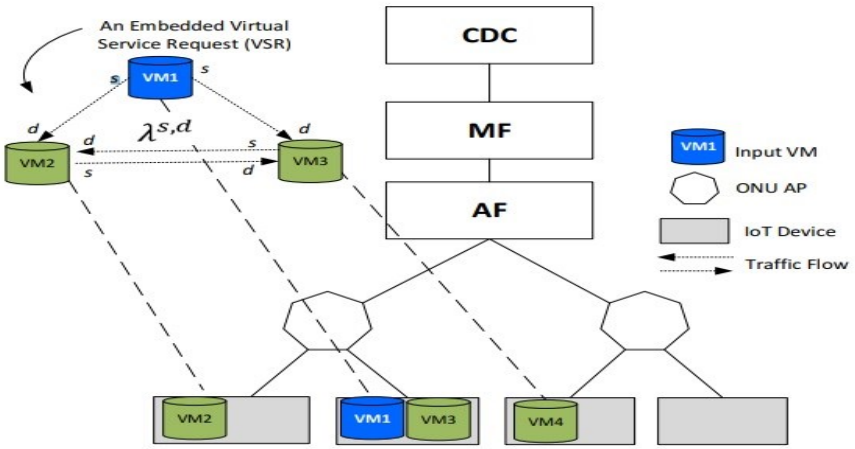



**Figure 3. 2: An illustration of VSR embedding over the Cloud Fog Network (CFN) Architecture**

## 3.3 The MILP Model

We develop a model to optimise the embedding of DNN VSRs into the CFN architecture depicted in Figure 3.1. As explained above, a VSR comprises of multiple VMs, each VM represents a layer of a DNN algorithms that has a demand for processing (in FLOPS) and the VMs are connected by virtual links with data rate demands (in Mbps). A VSR is embedded optimally on the CFN architecture while respecting capacity constraints of processing and networking devices. The CFN architecture shown in Figure 3.1 is modelled as an undirected graph $G = (N, L)$, where $N$ represents the set of all nodes and $L$ the set of links connecting those nodes in the topology. The VSR $s$ is represented by the directed graph $G^r = (R^r, L^r)$, where $R^r$ is the set of VMs, each representing a DNN layer and $L^r$ is the set of virtual links connecting those VMs. Before introducing the optimisation model, we define the sets, parameters and variables used:

**Sets:**

| | |
|---|---|
| $\mathbb{DC}$ | Set of CDCs. |
| $\mathbb{MF}$ | Set of MF nodes. |
| $\mathbb{AF}$ | Set of AF nodes. |
| $\mathbb{I}$ | Set of IoT devices. |
| $\mathbb{P}$ | Set of processing nodes that can process a VSR, where $\mathbb{P} = \mathbb{DC} \cup \mathbb{MF} \cup \mathbb{AF} \cup \mathbb{I}$. |

$\mathbb{IP}$ Set of source node IoT devices, $\mathbb{IP} \subset \mathbb{I}$

$\mathbb{R}$ Set of VSRs.

$\mathbb{VM}_r$ Set of VMs in VSR $r \in \mathbb{R}$.

$\mathbb{N}$ Set of networking nodes in the CFN architecture (IoT devices, ONUs, OLTs, metro nodes, core nodes).

$\mathbb{N}_m$ Set of neighbour nodes of node $\mathrm{m} \in \mathbb{N}$ in the CFN.

**Parameters:**

$s\ and\ d$ Index the source and destination nodes of a virtual link in a VSR topology, $s, d \in \mathbb{VM}_r,\ r \in \mathbb{R}$.

$b\ and\ e$ Index source and destination processing nodes of an end to end traffic demand aggregated from embedding VSR, $b, e \in P, b \neq e$.

$m\ and\ n$ Index the end nodes of physical links in the CFN topology.

$A_n^p$ $A_n^p = 1$ if processing node $p \in P$ and networking node $n \in N$ are co-located, otherwise $A_n^p = 0$.

$F^{r,s}$ Processing requested by node $s$ in VSR $r \in \mathbb{R}$.

$H^{r,s,d}$ Data rate of virtual link $(s, d)$ in VSR $r \in \mathbb{R}$.

$P_s^r$ $\mathrm{P_s^r} = 1$ if VM $s \in VM_r$ in VSR $r \in \mathbb{R}$ is the input layer, otherwise $P_s^r = 0$.

$\Pi_n^{(net)}$ Maximum power consumption of network node $n \in \mathbb{N}$ accounting for all equipment in the node.

$\pi_n^{(net)}$ Idle power consumption of network node $n \in \mathbb{N}$ accounting for all equipment in the node.

$C_n^{(net)}$ Capacity of network node $n \in \mathbb{N}$.

$\epsilon_n$ Energy per bit of network node $n \in \mathbb{N}$,

$\epsilon_n = \frac{\Pi_n^{(net)} - \pi_n^{(net)}}{C_n^{(net)}}$.

$\Pi_p^{(LAN)}$ Maximum power consumption of LAN network inside processing node p$\in \mathbb{P}$ accounting for all equipment in the LAN.

$\pi_p^{(LAN)}$ Idle power consumption of LAN network inside processing node p$\in \mathbb{P}$ accounting for all equipment in the LAN.

$C_p^{(LAN)}$ Capacity of LAN network inside processing node $p \in \mathbb{P}$.

$El_p$ Energy per bit of LAN network inside processing node $p \in \mathbb{P}, El_p = \frac{\Pi_p^{(LAN)} - \pi_p^{(LAN)}}{C_p^{(LAN)}}$.

$\Pi_p^{(pr)}$ Maximum power consumption of a single server at processing node $n \in \mathbb{P}$.

$\pi_p^{(pr)}$ Idle power consumption of a single server at processor node $p \in \mathbb{P}$.

$C_p^{(cpu)}$ Processing capacity of a serve at processing node $p \in \mathbb{P}$.

$E_p$ Energy per FLOPS of processing node $p \in \mathbb{P}, E_p = \frac{\Pi_p^{(pr)} - \pi_p^{(pr)}}{C_p^{(cpu)}}$

$NS_p$ Maximum number of servers that can be deployed at processing node $p \in \mathbb{P}$.

$\delta_n$ Proportion of idle power consumed in high-capacity networking equipment $n \in N$.

$PUE_n^{(net)}$ Power Usage Effectiveness (PUE) factor of node $n \in N$ for networking.

$PUE_p^{(net)}$ Power Usage Effectiveness (PUE) factor of node $p \in P$ for processing.

**Variables:**

$\lambda^{b,e}$ Traffic demand between processing node pair $(b, e)$ aggregated after all VSRs are embedded, $b, e \in \mathbb{P}$.

$\lambda_{m,n}^{b,e}$ Traffic demand between processing node pair $(b, e) \in \mathbb{P}$ aggregated after all VSRs are embedded, traversing physical link $(m, n)$, $m \in \mathbb{N}$ and $n \in \mathbb{N}_m$.

$\lambda_n$ Amount of traffic originating/passing by/destined to network node $n \in \mathbb{N}$,

where $\lambda_n = \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m} \lambda_{m,n}^{b,e} + \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{b,e}$ .

$\beta_n$ Amount of traffic destined to network node $n \in \mathbb{N}$,

where

$$\beta_n = \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m:n=e} \lambda_{m,n}^{b,e}$$

$\theta_p$ Amount of traffic destined to a processing node $p \in \mathbb{P}$,

where $\theta_p \ \forall n \in \mathbb{N}: A_n^p = 1$.

$\alpha_n$ $\alpha_n = 1$ if networking node $n \in \mathbb{N}$ is activated, otherwise $\alpha_n = 0$.

$\Omega_p$ Amount of workload in FLOPS, allocated to processing node $p \in \mathbb{P}$.

$N_p$ Number of activated servers at processing node $p \in \mathbb{P}$.

$\Phi_p$ $\Phi_p = 1$ if processing node $p \in \mathbb{P}$ is activated, otherwise $\Phi_p = 0$.

$\delta_b^{r,s}$ $\delta_b^{r,s} = 1$ if VM $s \in VM_r$ in VSR $r \in \mathbb{R}$ is embedded into processing node b$\in P$, otherwise $\delta_b^{r,s} = 0$.

$w_{b,e}^{r,s,d}$ $w_{b,e}^{r,s,d}$ is the XOR of $\delta_b^{r,s}$ and $\delta_e^{r,d}$, i.e. $w_{b,e}^{r,s,d} = \delta_b^{r,s} \oplus \delta_e^{r,d}$.

$\rho_{b,e}^{r,s,d}$ $\rho_{b,e}^{r,s,d} = 1$ if the virtual nodes $s, d \in \mathbb{VM}_r$ in VSR $r \in \mathbb{R}$ are successfully embedded in processing nodes $b, e \in \mathbb{P}$ respectively and a link between processing nodes $b, e$ is established if a virtual link exists between virtual nodes $s, d$, otherwise $\rho_{b,e}^{r,s,d} = 0$.

The total power consumption comprises of two parts: 1) network power consumption, 2) processing power consumption.

The adopted power profile consists of a proportional part and an idle part. The proportional part increases with the volume of workload, whilst the idle part is consumed as soon as the device is activated. We assume that any unused equipment is switched off completely.

The network power consumption is given by:

$$\sum_{n \in \mathbb{N}} PUE_n^{(net)} \left( \epsilon_n \lambda_n + \alpha_n \pi_n^{(net)} \delta_n \right). \quad (3.1)$$

The power consumption of the networking equipment comprises of the power consumption of all the networking nodes in the CFN topology depicted in Figure 3.1 multiplied by the PUE of each networking node. The first term of the above expression is the proportional power consumption of the networking equipment whilst the second term calculates the idle power consumption of these equipment.

The processing power consumption includes the power consumed by the servers as well as the switches and routers within these nodes to provide the LAN. The processing power consumption is given by:

$$\sum_{p\in\mathbb{P}} PUE_p^{(pr)}\left(E_p\Omega_p + N_p\pi_p^{(pr)} + EL_p\theta_p + \Phi_p\pi_p^{(LAN)}\delta_n\right) \tag{3.2}$$

The first term of the above expression is the proportional power consumption of the servers whilst the second term calculates the idle power consumption of these servers. The third and fourth terms are the idle and proportional power consumed by switches and routers of the internal LAN of the processing nodes.

The objective of the MILP is to minimise the total power consumption given as follows:

**Minimise:**

$$\sum_{n\in\mathbb{N}} PUE_n^{(net)}\left(\epsilon_n\,\lambda_n + \alpha_n\pi_n^{(net)}\delta_n\right) \tag{3.3}$$
$$+\sum_{p\in\mathbb{P}} PUE_p^{(pr)}\left(E_p\Omega_p + N_p\pi_p^{(pr)} + EL_p\theta_p\right.$$
$$\left.+\,\Phi_p\pi_p^{(LAN)}\delta_n\right).$$

**Subject to:**

$$\sum_{b\in\mathbb{P}} \delta_b^{r,s} = 1 \qquad \forall r \in \mathbb{R}, s \in \mathbb{VM}_r\colon \mathrm{P}_s^{\mathrm{r}} \neq 1 \tag{3.4}$$

Constraint (3.4) ensures that VMs of a VSR, except for input VMs, are embedded into any of the processing nodes.

$$\sum_{b\in\mathbb{IP}} \delta_b^{r,s} = 1 \qquad \forall r \in \mathbb{R}, s \in \mathbb{VM}_r\colon \mathrm{P}_s^{\mathrm{r}} = 1 \tag{3.5}$$

Constraint (3.5) ensures that input VMs of a VSR are embedded into source data IoT devices only.

$$\sum_{n\in\mathbb{N}_m} \lambda_{m,n}^{b,e} - \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{b,e} = \begin{cases} \lambda^{b,e} & m = s \\ -\lambda^{b,e} & m = d \\ 0 & otherwise \end{cases} \tag{3.6}$$

$$\forall b, e \in \mathbb{P}, d \in \mathbb{P}, m \in \mathbb{N}: b \neq e.$$

Constraint (3.6) preserves the flow of traffic in the network. This linear Equation ensures that, excluding the source and destination nodes, the total traffic demand passing through an intermediate node in the virtual connection from the source node to the network is identical to the total outgoing traffic demands leaving that intermediate node.

$$\sum_{b\in\mathbb{P}} \sum_{s\in\mathbb{VM}_r} \delta_b^{r,s} F^{r,s} = \sum_{s\in\mathbb{VM}_r} F^{r,s} \qquad \forall r \in \mathbb{R} \tag{3.7}$$

Constraint (3.7) ensures that the processing demand of request $r \in \mathbb{R}$ is fulfilled.

$$\delta_b^{r,s} + \delta_e^{r,d} = w_{b,e}^{r,s,d} + 2\rho_{b,e}^{r,s,d} \tag{3.8}$$

$$\forall r \in \mathbb{R}, (s, d) \in \mathbb{VM}_r, (b, e) \in \mathbb{P}: b \neq e, s \neq d$$

Constraint (3.8)(4.16) ensures that virtual nodes connected in the VSR topology are also connected on the physical network. This is done by introducing a binary variable $w_{b,e}^{r,s,d}$ that is only equal to 1 if $\delta_b^{r,s}$and $\delta_e^{r,d}$ e are exclusively equal to 1, otherwise $w_{b,e}^{r,s,d} = 0$.

$$\sum_{r\in\mathbb{R}} \sum_{s\in\mathbb{VM}_r} \sum_{\substack{d\in\mathbb{VM}_r: \\ s\neq d}} H^{r,s,d} \rho_{b,e}^{r,s,d} = \lambda^{b,e} \qquad \forall (b, e) \in \mathbb{P}: b \neq e \tag{3.9}$$

Constraint (3.9) ensures that the data rate requirement of virtual links is fulfilled. $H^{r,s,d}$ represents the bitrate requested by VSR on the virtual link $s, d$ and $\rho_{b,e}^{r,s,d}$ is variable to be =1 if connected virtual nodes are also connected

on the physical network; otherwise =0. $\lambda^{b,e}$ represents a variable of traffic demand between processing node pairs $b, e \in \mathbb{P}$ aggregated after all VSRs are embedded.

$$N_p \geq \frac{\Omega_p}{C_p^{(cpu)}} \quad \forall p \in \mathbb{P} \tag{3.10}$$

$$N_p \leq NS_p \quad \forall p \in \mathbb{P} \tag{3.11}$$

Constraints (3.10) and (3.11) determine the number of activated processing servers and ensures it is not larger than the number of servers the processing node host, respectively.

$$\lambda_n \geq \alpha_n \quad \forall n \in \mathbb{N} \tag{3.12}$$

$$\lambda_n \leq M\alpha_n \quad \forall n \in \mathbb{N} \tag{3.13}$$

Constraints (3.12) and (3.13) elate the binary variable $\alpha_n$ to the continuous variable $\lambda_n$, i.e. determine if a network node is activated or not based on the traffic traversing/generated/destined to the node.

$$\sum_{r \in \mathbb{R}} \sum_{s \in \mathbb{VM}_r} \delta_b^{r,s} \geq \Phi_p \quad \forall p \in \mathbb{P} \tag{3.14}$$

$$\sum_{r \in \mathbb{R}} \sum_{s \in \mathbb{VM}_r} \delta_b^{r,s} \leq M\Phi_p \quad \forall p \in \mathbb{P} \tag{3.15}$$

Constraints (3.14) and (3.15) determine the binary value of $\Phi_p$, i.e. determine if a processing node is activated based on the amount of processing performed by the node.

## 3.4 Results and Discussions

In this section, we study the energy efficiency of the CFN architecture by comparing optimised embedding of the DNN VSRs into the CFN to a baseline approach in which the DNN VSRs are processed by a CDC. We consider two scenarios: In Scenario 1 a single IoT device generates data for the input layer of the DNN VSRs while in Scenario 2 multiple IoT devices generates data for the input layer.

### 3.4.1 Scenario 1

In this scenario, we assume that there is only a single IoT device generating input data for a DNN VSR. We consider the parameters in Table 3.1 and Table 3.2 for the networking and processing nodes, respectively. It is important to note that, where possible, device parameters have been obtained using equipment datasheets, however, we have also made simple but realistic assumptions. For example, high-capacity networking equipment located in the aggregation point of the access, metro and core networks are used by many applications and services. Hence, we have assumed that, only a portion of the idle power consumption is associated with our applications. We assume this to be 3% of the equipment idle power consumption for access node (OLT), metro node and core node [114]. Notably, 3% of traffic globally is due to surveillance-type applications; hence, the proportional percentage chosen for the Idle power to serve those applications is 3% [114]. For the IoT devices and ONUs, we assume the device is use only by the DDN applications, i.e., all the idle power is attributed to the application. We have also assumed that the centralised data centre is a single hop from the aggregation core router and based on the topology of the NSFNET, the average distance between the core

nodes is 509 km [112]. We assume that in total, there are 20 IoT devices. The IoT devices are distributed among four zones such that zone one and zone two comprise of 6 IoT devices each whilst zone 3 and zone 4 comprise of 4 IoT devices each.

We examine the embedding of multiple DNN based VSRs up to 20. We assume that the VSRs arrive one at a time and each time a new VRS is embedded, all the existing VSRs are re-embedded to ensure the most optimum utilisation of resources. The number of VMs per VSR is randomly distributed between 2–4 VMs. The processing workload of input VMs and hidden layer VM is randomly distributed between 0.1 – 1 GFLOPS and between 0.6 – 10 GFLOPS, respectively. The virtual links data rate is randomly distributed between 0.1-2 Mbps.

We also assume that at each AF node and MF node, 6 and 10 servers are hosted, respectively while we assume the centralised data centre nodes have unlimited number of servers. It is important to note that the relatively high processing efficiency of the IoT layer is due to the use of low-power microprocessors in these nodes. However, they are very limited in terms of computational capacity compared to the fog and cloud layers [115]. We consider a PUE of 1.25 in AF node, 1.35 in the MF node, 1.12 in the CDC node, 1.5 for core nodes and 1 in ONUs and IoT devices as these do not require cooling [114]. The MILP model is solved using IBM’s commercial solver CPLEX over the University of Leeds high performance computing facilities (ARC3) using 24 cores with 126 GB of RAM [116].

**Table 3. 1: Processing Device parameters for scenario 1 [117]**

| Devices | Max(W) | Idle(W) | GFLOPS | Efficiency (W/GFLOPS) |
|---|---|---|---|---|
| IoT (Rpi 4 B 4GB) | 7.3 | 2.56 | 13.5 | 0.35 |
| AF Server (Intel i5-3427U) | 37.2 | 13.8 | 34.5 | 0.67 |
| MF Server (Intel i5-3427U) | 37.2 | 13.8 | 34.5 | 0.67 |
| CDC (Intel Xeon E5-2640) | 298 | 58.7 | 428 | 0.55 |

**Table 3. 2: Networking Devices for Scenario 1**

| Devices | Max (W) | Idle (W) | Bitrate (Gbps) | Efficiency (J/Gb) |
|---|---|---|---|---|
| IoT Wi-Fi interface [117] | 0.56 | 0.34 | 0.1 | 2.2 |
| ONU (including Wi-Fi interface) [114] | 15 | 9 | 10 | 0.6 |
| OLT [114] | 1940 | 60 | 8600 | 0.22 |
| Metro Router Port [114] | 30 | 27 | 40 | 0.08 |
| Metro Switch [114] | 470 | 423 | 600 | 0.08 |
| IP/WDM Node [114] | 878 | 790 | *40/$\lambda$ | 0.14 |

In Figure 3.3, four different embedding scenarios are evaluated: 1) VSRs being embedded at the CDC, 2) VSRs being embedded at the AF, 3) VSRs being embedded at the MF, and 4) optimised embedding of VSRs across the CFN (CFN-MILP). We observed significant power consumption savings with the optimised embedding into the CFN architecture compared to processing at the CDC due to local computation in the IoT layer as seen in Figure 3.4. The savings are up to 91% (68% on average). These savings can be attributed

to the processing efficiency of the local IoT devices as well as the access network used to connect them, hence avoiding various costly overheads such as network power consumption and PUE values associated with the higher processing layers. With the optimised embedding into the CFN, due to the abundance of processing resources, VSRs are embedded by the IoT layer. In Figure 3.3(a), there is a spike in processing and networking power consumption. This is because during very high workloads (20 VSRs), the MILP model chooses to split the workload among the IoT and CDC servers as seen in Figure 3.4 due to capacity violation in the IoT layer. Hence, the CDC node is used to process the excessive workloads only and most of the total workload is kept at the IoT layer. Note that the Access Fog (AF) and Metro Fog (MF) nodes are never utilised despite their proximity to the input node at the IoT layer and the negligible network power consumption as per Figure 3.3 (b) and (c). This is due to the processing inefficiency of these nodes coupled with the high PUE values. If the CDC node was to be further away from the aggregating metro node (more than one hop) then the processing efficiency of the cloud data centre (CDC) may not compensate for the power consumed in the core network to access the CDC, hence the AF and MF may have a role to play.

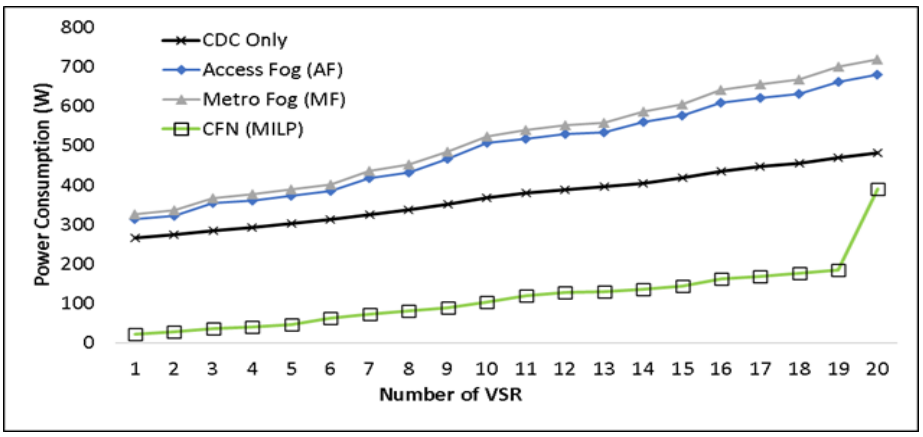



**Figure 3. 3: Total power consumption vs. no. of Virtual Service Requests (VSRs) under different placement solutions**

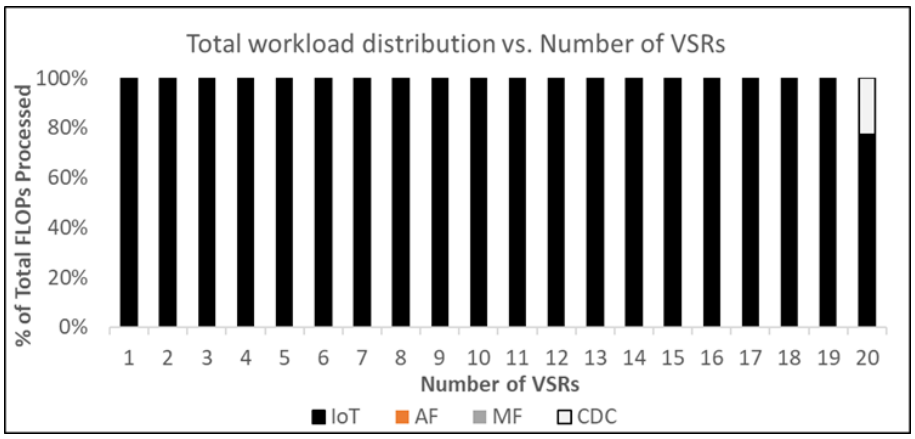
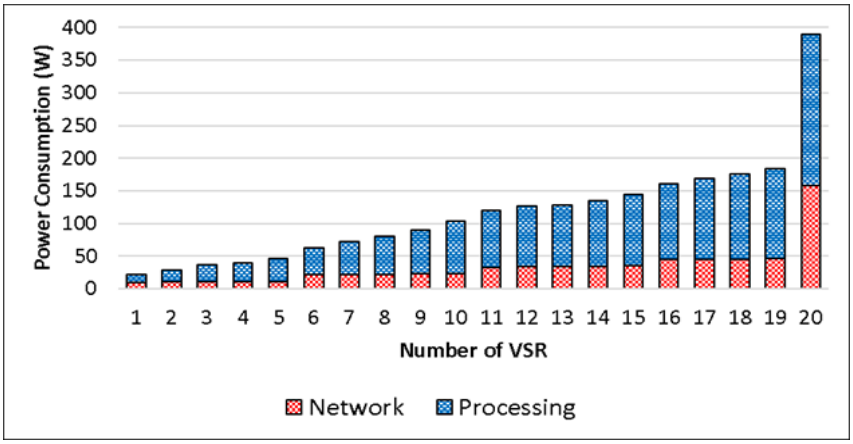
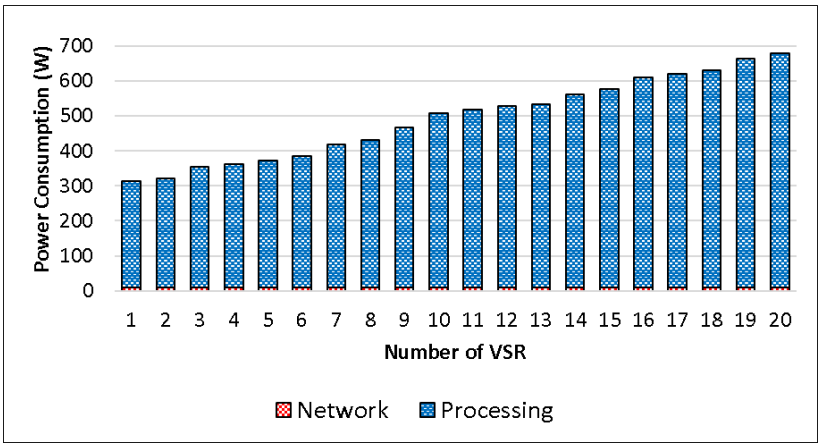


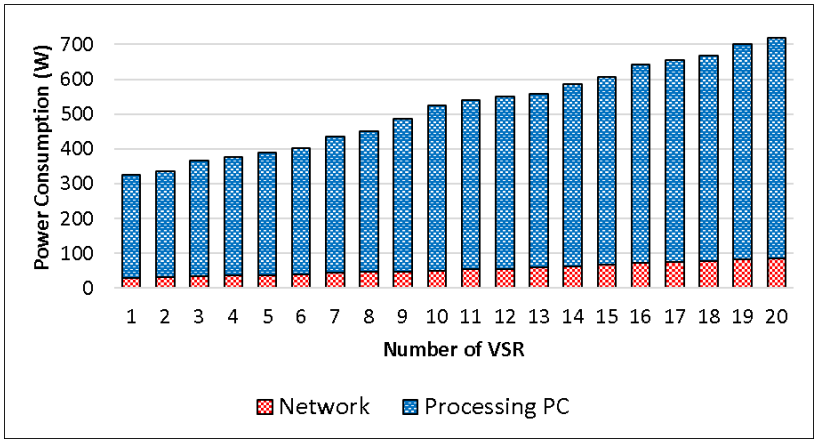

**Figure 3. 4: Workload distribution of Scenario 1**



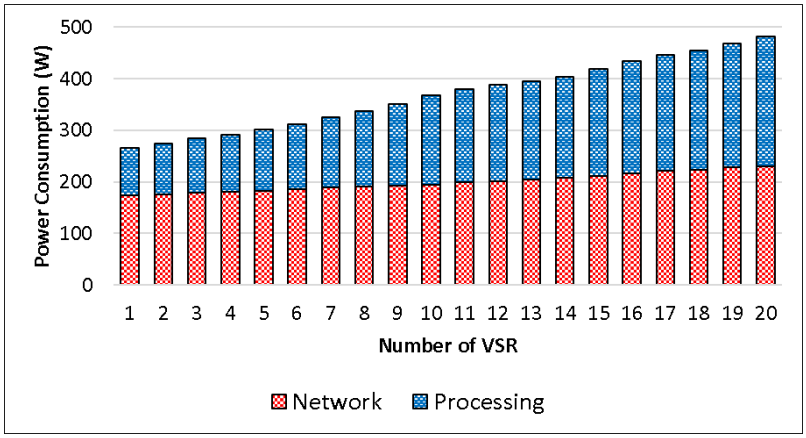

(a)




(b)




(c)




(d)

**Figure 3. 5: Network vs. processing power consumption of; (a) CFN (MILP), (b) AF and (c) MF (d) CDC.**

### 3.4.2 Scenario 2

This scenario considers multiple IoT nodes generating data for the input layer of the DNN VSRs. The impact of having single versus multiple inputs on the performance of the CFN is studied and the total power savings compared to the baseline solution (processing in CDC) are quantified. We have made some changes to the architecture. As can be seen in Figure 3.6, we have increased the scale of the network by adding more OLTs and AFs in addition to the multiple IoT nodes feeding data into the input layer.

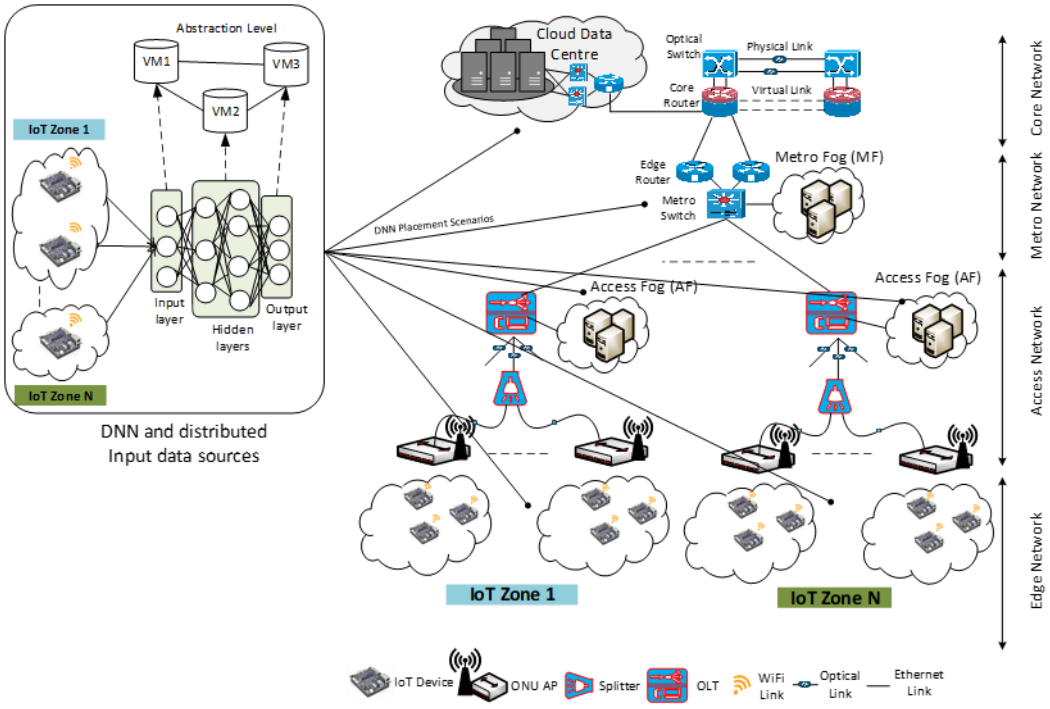



**Figure 3. 6: Multiple IoT nodes generating data for DNN input layer over the proposed CFN architecture.**

**Table 3. 3: Processing Device Parameters for Scenario 2 [117]**

| Devices | Max(W) | Idle(W) | GFLOPS | Efficiency (W/GFLOPS) |
|---|---|---|---|---|
| IoT (Rpi 4 B 4GB) | 7.3 | 2.56 | 13.5 | 0.35 |
| AF Server (Intel i5-3427U) | 32.6 | 10 [1][2] | 47.7 | 0.47 |
| MF Server (Intel i5-3427U) | 134 | 29 [1] [2] | 181 | 0.58 |
| CDC (Intel Xeon E5-2640) | 298 | 58.7 | 428 | 0.55 |

**Table 3. 4: Network Devices Parameter for Scenario 2**

| Devices | Max (W) | Idle (W) | Bitrate (Gbps) | Efficiency (W/Gbps) |
|---|---|---|---|---|
| IoT Wi-Fi interface [117] | 0.56 | 0.34 | 0.1 | 2.2 |
| ONU (including Wi-Fi interface) [114] | 15 | 9** | 10 | 0.6 |
| OLT [114] | 1940 | 1746*** | 8600 | 0.22 |
| Metro Router Port [114] | 30 | 27*** | 40 | 0.08 |
| Metro Switch [114] | 470 | 423*** | 600 | 0.08 |
| IP/WDM Node [114] | 878 | 790*** | 40/wavelength | 0.14 |

*40Gbps / wavelength ** is 60% of max power ***is 90% of max power.

In addition to the aforementioned changes, we have also made some changes to network and processing parameters as seen in Table 3.3 and Table 3.4. In the previous scenario the OLT device was much more efficient in terms of the idle power consumption however in this scenario we have considered a more practical OLT device that has a higher idle power consumption. This was done to see if distributing DNN layers would be impacted since the network is less efficient now. As for the server parameters, unlike the previous scenario, the AF and MF nodes have been assigned heterogeneous processing capability.

We assume that in total, there are 30 IoT devices, equally distributed among 10 IoT zones: IoT Zone 1 – IoT Zone 10. Ten of the IoT devices act as data sources. In total, we consider 3 OLT devices and each one aggregates traffic from a cluster of 3 or 4 ONUs.

We consider DNN VSRs similar to those considered in Scenario 1 but with input VMs to be embedded in 10 source data IoT devices. The input VMs workload is randomly distributed between 0.1–1 GFLOPS. Also, a higher workload is considered for hidden VMs randomly distributed between 2 – 13.5 GFLOPS.

We consider the nodes to have more efficient PUE factors. The PUE factors are 1.1, 1.25, 1.1 in AF, MF and CDC nodes, respectively. Similar to Scenario 1, we also assume that at each AF and MF node hosts 6 and 10 servers, respectively while we assume the centralised data centre nodes have unlimited number of servers. Similar to the Scenario 1, the MILP model is solved using IBM's commercial solver CPLEX over the University of Leeds high performance computing facilities (ARC3) using 24 cores with 126 GB of RAM [116].

In the following subsections, we compare Scenario 2 to Scenario 1 considering the CFN architecture in Figure 3.6 and a fog architecture where processing is only available in IoT nodes, AF nodes and MF nodes. The fog architecture has similar parameters to those of the CFN architecture in Figure 3.6. This comparison evaluates the increase in power consumption resulting from limiting the processing of DNN VSRs to the IoT devices and fog nodes.

### 3.5.2.1 Performance of the CFN architecture

Figure 3.7 shows the total power consumption which included processing and networking power consumption versus the number of VSRs for Scenario 1 and Scenario 2 considering the optimised CFN architecture and the baseline approach where all processing takes place in the CDC. Figure 3.8 and Figure 3.9 break the total power consumption to networking power consumption and processing power consumption. We reproduced Scenario 1 results considering the input parameters of this section for fair comparison. For Scenario 1, compared to the baseline, the CFN solution achieves up to 68% (average 12%) power consumption reduction for VSRs (1 VSR – 9 VSRs). This is because, as explained above, VSRs can be processed on local low-power IoT devices as seen in Figure 3.10 (a). For 10 VSRs, some of the VMs are embedded in the cloud. From 11 VSRs to 27 VSRs, all the hidden layers are embedded in the cloud. Interestingly, IoT utilisation increases again (from 28 VSRs – 30 VSRs). This happens to avoid activating an additional server at the CDC due to its high idle power consumption.

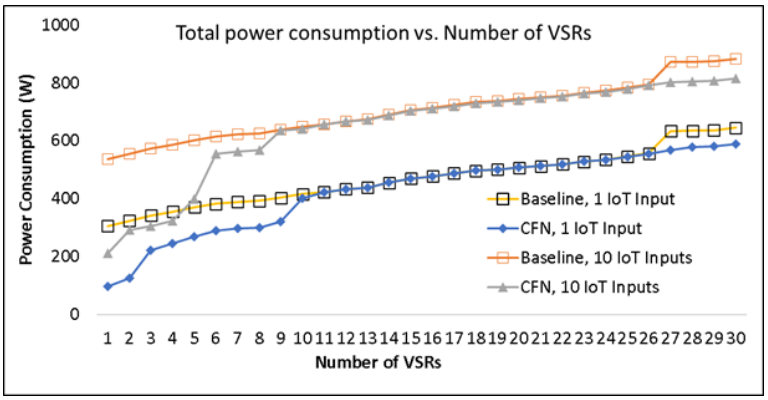



**Figure 3. 7: Total power consumption of CFN with/out cloud data centre.**

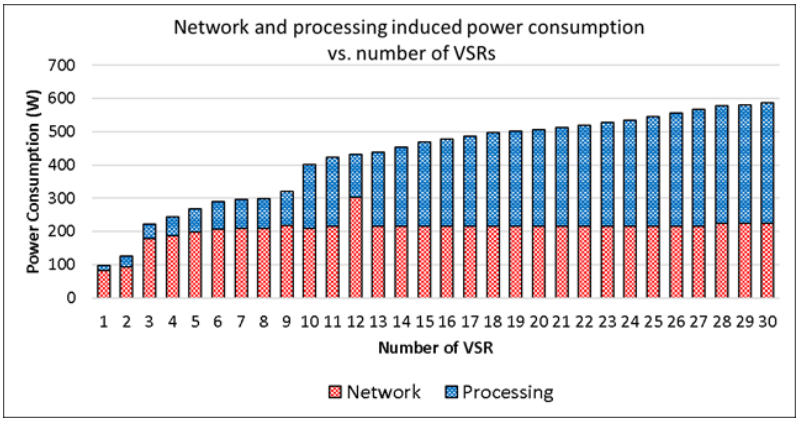



**Figure 3. 8: Breakdown of CFN's network and processing power consumption (scenario 1).**

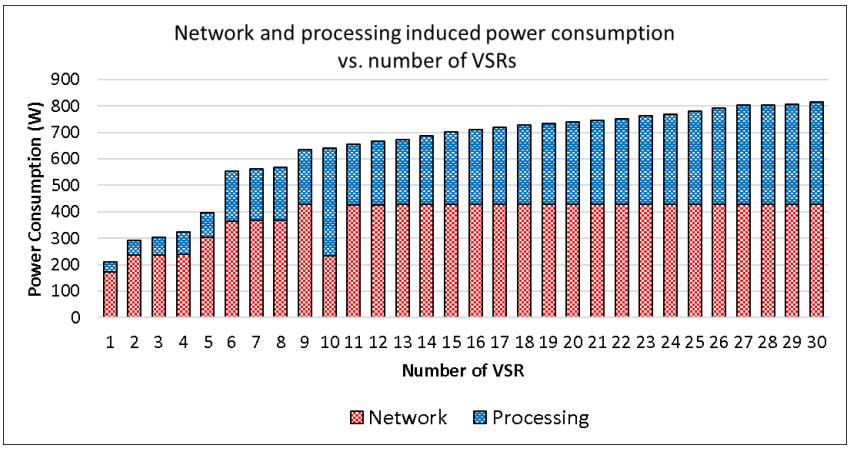
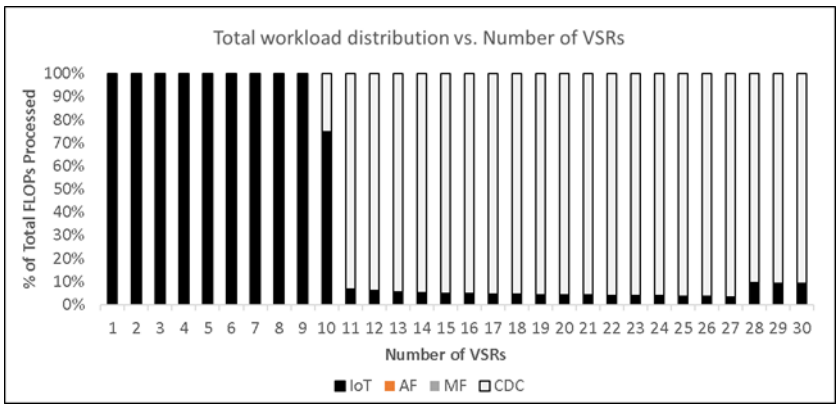
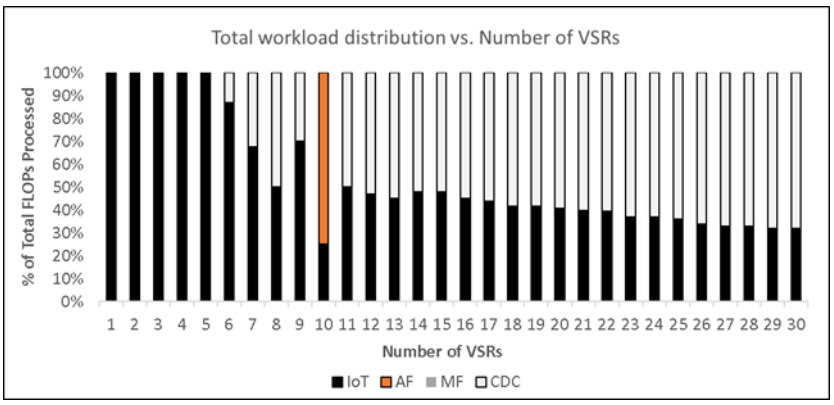



**Figure 3. 9: Breakdown of CFN's network and processing power consumption (scenario 2).**




(a)




(b)

**Figure 3. 9: Workload distribution of: (a) Scenario 1, and (b) Scenario 2.**

In Scenario 2, the IoT and CDC layers are predominantly the optimal choice with utilisation of the AF layer at 10 VSRs only as seen in Figure 3.10 (b). Note that all of the OLT devices that connect the different zones will be activated as we have 1 input per zone. This makes the power consumed to access the higher processing nodes (AF, MF and cloud) nodes lower and therefore the cloud and the AF are used to embed VMs for number of VSRs lower than that of Scenario 1. From Figure 3.7, compared to the baseline, Scenario 2 achieves up to 60% (average 10%) power consumption reduction for 1 VSR – 9 VSRs. In future, as the processing efficiency of fog servers is improves and PUE factors are minimised, fog nodes may provide energy efficient solutions for embedding DNN VSRs.

### 3.5.2.2 Performance of the fog architecture

In this subsection, we aim to evaluate the increase in power consumption resulting from limiting the processing of DNN VSRs to the IoT devices and fog nodes for Scenario 1 and Scenario 2. Figure 3.11 shows that in Scenario 1, the fog architecture yields an increase in power consumption up to 44% (19% on average) compared to the CFN architecture. In Scenario 2 the increase in power consumption of the fog architecture compared to the FCN architecture is reduced to 20% maximum (10% average). This limited increase in power consumption as a result of embedding VMs in the fog architecture compared to the CFN architecture indicates that improvement in the energy efficiency of fog node by similar magnitude, which is anticipated in the near future, would allow efficient use of fog nodes in embedding DNN VSRs.

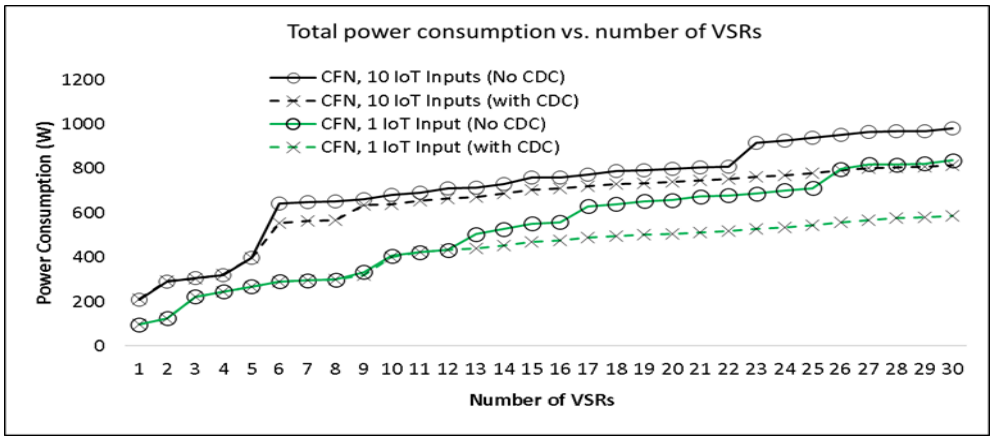



**Figure 3. 10: The total power consumption of the CFN approach with/without CDC collaboration under single and multiple input IoT nodes.**

## 3.5 Summary

This chapter investigated the power consumption associated with the embedding of DNN VSRs over a CFN architecture. The layers of the DNN were abstracted as VMs that are interconnected by virtual links. Each VM has

processing demand in FLOPs and each virtual link has bandwidth demand in bps. The studies in this chapter were divided into two scenarios: Scenario 1 and Scenario 2. Under Scenario 1, we looked at a case where DNN VSRs have a single IoT device generating data for the input layer, whilst in Scenario 2, we studied multiple IoT devices generating data for the input layer of the DNN VSRs. We developed a MILP model to optimise the embedding of DNN VMs under the two scenarios. For Scenario 1, the optimisation results showed the VMs are optimally processed on local IoT devices in collaboration with the cloud. This resulted in up to 68% power savings (12% on average) compared to processing all the VMs in the cloud considering a scalable architecture with multiple OLTs.

For Scenario 2, the results also showed that despite of the number of source nodes and their geographical deployment, the distribution of the VMs among the IoT devices in other parts of the network is still more favourable than consolidating them in higher capacity fog servers that were associated with a higher PUE value. The optimum embedding over the CFN architecture produced up to 60% power savings (10% on average) considering a scalable architecture with multiple OLTs.

Furthermore, we evaluated the energy efficiency of a fog architecture in embedding DNN VSRs. The results showed that optimising the embedding over the fog architecture resulted in up to 44% (19% on average) and up to 20% (10% on average) increase in total power consumption compared to the optimum embedding over the CFN architecture considering scenario 1 and Scenario 2, respectively. This increase in power consumption is due to the high PUE factor and less energy efficiency of servers in fog nodes. This

limited increase in power consumption indicates that improvement in the energy efficiency of fog node by similar magnitude, which is anticipated in the near future, would allow efficient use of fog nodes in embedding DNN VSRs.

# Chapter 4: Energy Efficient Placement of DNN Services over a Cloud-Fog Network: the impact of VM allocation constraint

## 4.1 Introduction

In Chapter 4, we evaluate the embedding of each layer of a DNN as by a VSR, whose is modelled as a number of connected VMs. under the assumption that the IoT layer could process all types of VMs. However, in a practical scenario, IoT devices can be limited in terms of the type and the number of VMs they can host due to hardware / software limitations. Therefore, we introduce a constraint that only permits a limited number of VMs to be processed by any IoT device at a given time. We refer to this case as IoT limited VM allocation constraint. To draw comparisons and quantify the power savings, we evaluate the performance under different limits on the number of VMs an IoT node can host.

Furthermore, we extend the studies of the previous chapter by studying how the performance of the CFN architecture is affected by the idle power proportion ratio ($\delta$) attributed to the DNN applications in highly shared networking equipment in the access, metro and core networks. A range of values of $\delta$ (3%, 6% and 10%) are considered. The increase in $\delta$ represents the potential growth of the DNN applications in the future.

## 4.2 MILP Model

We extend the MILP model in Chapter 4 to reflect the restriction on the number of VMs that can be embedded into an IoT device. We present the complete model in this section for improved readability.

The sets, parameters and variables are defined:

**Sets:**

| | |
|---|---|
| $\mathbb{DC}$ | Set of CDCs. |
| $\mathbb{MF}$ | Set of MF nodes. |
| $\mathbb{AF}$ | Set of AF nodes. |
| $\mathbb{I}$ | Set of IoT devices. |
| $\mathbb{P}$ | Set of processing nodes that can process a VSR, where $\mathbb{P} = \mathbb{DC} \cup \mathbb{MF} \cup \mathbb{AF} \cup \mathbb{I}$. |
| $\mathbb{IP}$ | Set of source node IoT devices, $\mathbb{IP} \subset \mathbb{I}$ |
| $\mathbb{R}$ | Set of VSRs. |
| $\mathbb{VM}_r$ | Set of VMs in VSR $r \in \mathbb{R}$. |
| $\mathbb{N}$ | Set of networking nodes in the CFN architecture (IoT devices, ONUs, OLTs, metro nodes, core nodes). |
| $\mathbb{N}_m$ | Set of neighbour nodes of node $\mathrm{m} \in \mathbb{N}$ in the CFN. |

**Parameters:**

| | |
|---|---|
| $s\ and\ d$ | Index the source and destination nodes of a virtual link in a VSR topology, $s, d \in \mathbb{VM}_r,\ r \in \mathbb{R}$. |
| $b\ and\ e$ | Index source and destination processing nodes of an end to end traffic demand aggregated from embedding VSR, $b, e \in P, b \neq e$. |

$m\ and\ n$ Index the end nodes of physical links in the CFN topology.

$A_n^p$ $A_n^p = 1$ if processing node $p \in P$ and networking node $n \in N$ are co-located, otherwise $A_n^p = 0$.

$F^{r,s}$ Processing requested by node $s$ in VSR $r \in \mathbb{R}$.

$H^{r,s,d}$ Data rate of virtual link $(s, d)$ in VSR $r \in \mathbb{R}$.

$P_s^r$ $\mathrm{P_s^r} = 1$ if VM $s \in VM_r$ in VSR $r \in \mathbb{R}$ is the input layer, otherwise $P_s^r = 0$.

$\Pi_n^{(net)}$ Maximum power consumption of network node $n \in \mathbb{N}$ accounting for all equipment in the node.

$\pi_n^{(net)}$ Idle power consumption of network node $n \in \mathbb{N}$ accounting for all equipment in the node.

$C_n^{(net)}$ Capacity of network node $n \in \mathbb{N}$.

$\epsilon_n$ Energy per bit of network node $n \in \mathbb{N}$,

$\epsilon_n = \frac{\Pi_n^{(net)} - \pi_n^{(net)}}{C_n^{(net)}}$.

$\Pi_p^{(LAN)}$ Maximum power consumption of LAN network inside processing node p$\in \mathbb{P}$ accounting for all equipment in the LAN.

$\pi_p^{(LAN)}$ Idle power consumption of LAN network inside processing node p∈ $\mathbb{P}$ accounting for all equipment in the LAN.

$C_p^{(LAN)}$ Capacity of LAN network inside processing node $p \in \mathbb{P}$.

$El_p$ Energy per bit of LAN network inside processing node $p \in \mathbb{P}, El_p = \frac{\Pi_p^{(LAN)} - \pi_p^{(LAN)}}{C_p^{(LAN)}}$.

$\Pi_p^{(pr)}$ Maximum power consumption of a single server at processing node $n \in \mathbb{P}$.

$\pi_p^{(pr)}$ Idle power consumption of a single server at processor node $p \in \mathbb{P}$.

$C_p^{(cpu)}$ Processing capacity of a serve at processing node $p \in \mathbb{P}$.

$E_p$ Energy per FLOPS of processing node $p \in \mathbb{P}, E_p = \frac{\Pi_p^{(pr)} - \pi_p^{(pr)}}{C_p^{(cpu)}}$

$NS_p$ Maximum number of servers that can be deployed at processing node $p \in \mathbb{P}$.

$\delta_n$ Proportion of idle power consumed on high-capacity networking equipment $n \in N$.

$PUE_n^{(net)}$ Power Usage Effectiveness (PUE) factor of node $n \in N$ for networking.

$PUE_p^{(net)}$ Power Usage Effectiveness (PUE) factor of node $p \in P$ for processing.

**Variables:**

$\lambda^{b,e}$ Traffic demand between processing node pair $(b,e)$ aggregated after all VSRs are embedded, $b,e \in \mathbb{P}$.

$\lambda_{m,n}^{b,e}$ Traffic demand between processing node pair $(b,e) \in \mathbb{P}$ aggregated after all VSRs are embedded, traversing physical link $(m,n)$, $m \in \mathbb{N}$ and $n \in \mathbb{N}_m$.

$\lambda_n$ Amount of traffic originating/passing by/destined to network node $n \in \mathbb{N}$,

where $\lambda_n = \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m} \lambda_{m,n}^{b,e} + \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{b,e}$ .

$\beta_n$ Amount of traffic destined to network node $n \in \mathbb{N}$,

where

$$\beta_n = \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m:n=e} \lambda_{m,n}^{b,e}$$

$\theta_p$ Amount of traffic destined to a processing node $p \in \mathbb{P}$,

where $\theta_p \ \forall n \in \mathbb{N}: A_n^p = 1$.

$\alpha_n$ $\alpha_n = 1$ if networking node $n \in \mathbb{N}$ is activated, otherwise $\alpha_n = 0$.

$\Omega_p$ Amount of workload in FLOPS, allocated to processing node $p \in \mathbb{P}$.

$N_p$ Number of activated servers at processing node $p \in \mathbb{P}$.

$\Phi_p$ $\Phi_p = 1$ if processing node $p \in \mathbb{P}$ is activated, otherwise $\Phi_p = 0$.

$\delta_b^{r,s}$ $\delta_b^{r,s} = 1$ if VM $s \in VM_r$ in VSR $r \in \mathbb{R}$ is embedded into processing node b$\in P$, otherwise $\delta_b^{r,s} = 0$.

$w_{b,e}^{r,s,d}$ $w_{b,e}^{r,s,d}$ is the XOR of $\delta_b^{r,s}$ and $\delta_e^{r,d}$, i.e. $w_{b,e}^{r,s,d} = \delta_b^{r,s} \oplus \delta_e^{r,d}$.

$\rho_{b,e}^{r,s,d}$ $\rho_{b,e}^{r,s,d} = 1$ if the virtual nodes $s, d \in \mathbb{VM}_r$ in VSR $r \in \mathbb{R}$ are successfully embedded in processing nodes $b, e \in \mathbb{P}$ respectively and a link between processing nodes $b, e$ is established if a virtual link exists between virtual nodes $s, d$, otherwise $\rho_{b,e}^{r,s,d} = 0$.

The total power consumption comprises of two parts: 1) network power consumption, 2) processing power consumption.

The power profile adopted consists of a proportional part and idle part. The proportional part increases with the volume of workload, whilst the idle part is consumed as soon as the device is activated. We assume that any unused equipment is switched off completely.

The network power consumption is given by:

$$\sum_{n \in \mathbb{N}} PUE_n^{(net)} \left( \epsilon_n \lambda_n + \alpha_n \pi_n^{(net)} \delta_n \right). \quad (4.1)$$

The power consumption of the networking equipment comprises of the power consumption of all the networking nodes in the CFN topology depicted in

Figure 3.1 multiplied by the PUE of each networking node. The first term of the above expression is the proportional power consumption of the networking equipment whilst the second term calculates the idle power consumption of these equipment.

The processing power consumption includes the power consumed by the servers as well as the switches and routers within these nodes to provide the LAN. The processing power consumption is given by:

$$\sum_{p\in\mathbb{P}} PUE_p^{(pr)}\left(E_p\Omega_p + N_p\pi_p^{(pr)} + EL_p\theta_p + \Phi_p\pi_p^{(LAN)}\delta_n\right) \tag{4.2}$$

The first term of the above expression is the proportional power consumption of the servers whilst the second term calculates the idle power consumption of these servers. The third and fourth terms are the idle and proportional power consumed by switches and routers of the internal LAN of the processing nodes.

The objective of the MILP is to minimise the total power consumption given as follows:

**Minimise:**

$$\sum_{n\in\mathbb{N}} PUE_n^{(net)}\left(\epsilon_n\,\lambda_n + \alpha_n\pi_n^{(net)}\delta_n\right) \tag{4.3}$$

$$+\sum_{p\in\mathbb{P}} PUE_p^{(pr)}\left(E_p\Omega_p + N_p\pi_p^{(pr)} + EL_p\theta_p + \Phi_p\pi_p^{(LAN)}\delta_n\right).$$

$\epsilon_n$represent energy per bit of a network node as parameter $\lambda_n$ is variable for amount of traffic passing a network node. $\alpha_n$ is variable to indicate if $n \in \mathbb{N}c$ is activated, $\pi_n^{(net)}$ is parameter of idle power consumption of networking devices. $\delta_n$is parameter of proportion idle power consumption on high-capacity network equipment. $E_p$is a parameter for the energy per FLOPS of processing node $p \in \mathbb{P}$, $\Omega_p$ is a variable for the amount of workload in FLOPS that is located in processing node $p \in \mathbb{P}$. $N_p$is variable of number activated servers at processing node $p \in \mathbb{P}$. $\pi_p^{(pr)}$ is parameter of idle power consumption of single server at processing node $p \in \mathbb{P}$. $EL_p$is parameter of energy per bit of LAN network inside processing node $p \in \mathbb{P}$. $\theta_p$ is variable of the amount traffic destined to a processing node $p \in \mathbb{P}$. $\Phi_p$ is variable to indicate if a processing node $p \in \mathbb{P}$ is activated. $\pi_p^{(LAN)}$ is parameter of idle power consumption of LAN network inside processing node $p \in \mathbb{P}$.

**Subject to:**

$$\sum_{b \in \mathbb{P}} \delta_b^{r,s} = 1 \qquad \forall r \in \mathbb{R}, s \in \mathbb{VM}_r : \mathrm{P}_s^r \neq 1 \tag{4.4}$$

Constraint (4.4) ensures that VMs of a VSR, except for input VMs, are embedded into any of the processing nodes.

$$\sum_{b \in \mathbb{IP}} \delta_b^{r,s} = 1 \qquad \forall r \in \mathbb{R}, s \in \mathbb{VM}_r : \mathrm{P}_s^r = 1 \tag{4.5}$$

Constraint (4.5) ensures that input VMs of a VSR are embedded into source data IoT devices only.

$$\sum_{n\in\mathbb{N}_m} \lambda_{m,n}^{b,e} - \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{b,e} = \begin{cases} \lambda^{b,e} & m = s \\ -\lambda^{b,e} & m = d \\ 0 & otherwise \end{cases} \tag{4.6}$$

$$\forall b, e \in \mathbb{P}, d \in \mathbb{P}, m \in \mathbb{N}: b \neq e.$$

Constraint (4.6) preserves the flow of traffic in the network.

$$\sum_{b\in\mathbb{P}} \sum_{s\in\mathbb{VM}_r} \delta_b^{r,s} F^{r,s} = \sum_{s\in\mathbb{VM}_r} F^{r,s} \qquad \forall r \in \mathbb{R} \tag{4.7}$$

Constraint (4.7) ensures that the processing demand of request $r \in \mathbb{R}$ is fulfilled.

$$\delta_b^{r,s} + \delta_e^{r,d} = w_{b,e}^{r,s,d} + 2\rho_{b,e}^{r,s,d} \tag{4.8}$$

$$\forall r \in \mathbb{R}, (s, d) \in \mathbb{VM}_r, (b, e) \in \mathbb{P}: b \neq e, s \neq d$$

Constraint (4.8) ensures that virtual nodes connected in the VSR topology are also connected on the physical network. This done by introducing a binary variable $w_{b,e}^{r,s,d}$ that is only equal to 1 if $\delta_b^{r,s}$and $\delta_e^{r,d}$ e are exclusively equal to 1, otherwise $w_{b,e}^{r,s,d} = 0$.

$$\sum_{r\in\mathbb{R}} \sum_{s\in\mathbb{VM}_r} \sum_{\substack{d\in\mathbb{VM}_r: \\ s\neq d}} H^{r,s,d}\, \rho_{b,e}^{r,s,d} = \lambda^{b,e} \qquad \forall (b, e) \in \mathbb{P}: b \neq e \tag{4.9}$$

Constraint (4.9) ensures that the data rate requirement of virtual links are fulfilled.

$$N_p \geq \frac{\Omega_p}{C_p^{(cpu)}} \qquad \forall p \in \mathbb{P} \tag{4.10}$$

$$N_p \leq NS_p \qquad \forall p \in \mathbb{P} \tag{4.11}$$

Constraints (4.10) and (4.11) determine the number of activated processing servers and ensures it is not larger than the number of servers the processing node host, respectively.

$$\lambda_n \geq \alpha_n \qquad \forall n \in \mathbb{N} \tag{4.12}$$

$$\lambda_n \leq M\alpha_n \qquad \forall n \in \mathbb{N} \tag{4.13}$$

Constraints (4.12) and (4.13) relates the binary variable $\alpha_n$ to the continuous variable $\lambda_n$, i.e. determines if a network node is activated or not based on the traffic traversing/generated/destined to the node.

$$\sum_{r\in\mathbb{R}}\sum_{s\in\mathbb{VM}_r} \delta_b^{r,s} \geq \Phi_p \quad \forall p \in \mathbb{P} \tag{4.14}$$

$$\sum_{r\in\mathbb{R}}\sum_{s\in\mathbb{VM}_r} \delta_b^{r,s} \leq M\Phi_p \quad \forall p \in \mathbb{P} \tag{4.15}$$

Constraints (4.14) and (4.15) determine the binary value of $\Phi_p$, i.e. determine if a processing node is activated based on the amount of processing performed by the node.

$$\sum_{r\in\mathbb{R}}\sum_{s\in\mathbb{VM}_r} \delta_b^{r,s} \leq k \qquad \forall b \in \mathbb{IoT} \tag{4.16}$$

Constraint (4.16) restricts the sum of VMs allocated to an IoT node to be less than or equal to the parameter $k$.

## 4.3 Results and Discussions

In this chapter we consider the architecture, input parameters and power consumption values of Section 3.4.1 with the scenario where input VMs is embedded in a single source node IoT device. In this chapter we have

executed the model to embed 15 DNN VSRs under single VM allocation and multiple VMs allocation under different values for the idle power proportion ratio ($\delta$).

### 4.3.1 Single VM Allocation at IoT

We evaluate the impact of limiting the number of VMs embedded in an IoT node at a given time to one VM (i.e., k=1). Our aim is to represent a scenario in which, due to hardware / software limitations or low power limitations, IoT nodes are not capable of processing multiple types of VMs. Figure 4.1 shows the total power consumption which is the sum of the networking and processing power consumptions of all nodes against different values of the $\delta$ factor. It can be observed that when $\delta$ is higher than 3%, the total networking and computing power consumption increased by 22% compared the case when $\delta$ equals 3% It also shows that when $\delta$ is low (3%), the networking power consumption is about 30% of the total power consumption while in the cases when $\delta$ is higher than 3%, the networking power consumption is about 10% of the total power consumption. Figure 4.2 shows the distribution of processing among the cloud, MFN, AFN, and the IoT layers for the case of k=1 and for different values of the $\delta$ factor. The results show that when $\delta$ is low (3%), the IoT devices are used to embed part of the VMs and the remaining is embedded in the CDC (60% of the workload is processed by the CDC while 40% is processed by the IoT nodes). The CDC is favoured over the AFN and MFN due to the processing efficiency of the CDC and low network power consumption to access the CDC with low $\delta$. For high $\delta$ (6% and 10%), it can be observed that the CDC is no longer the favourite choice as it loses its merit due to the high-power consumption of the transport

network. In these cases, the IoT nodes are assigned to process 60% of the workload while the AFN is assigned to process 40% of the workload. When $\delta = 3\%$, Figure 4.2 shows that the total power consumption reached 300w. This is attributed to the total power consumption of processing and networking of all considered nodes in the architecture.

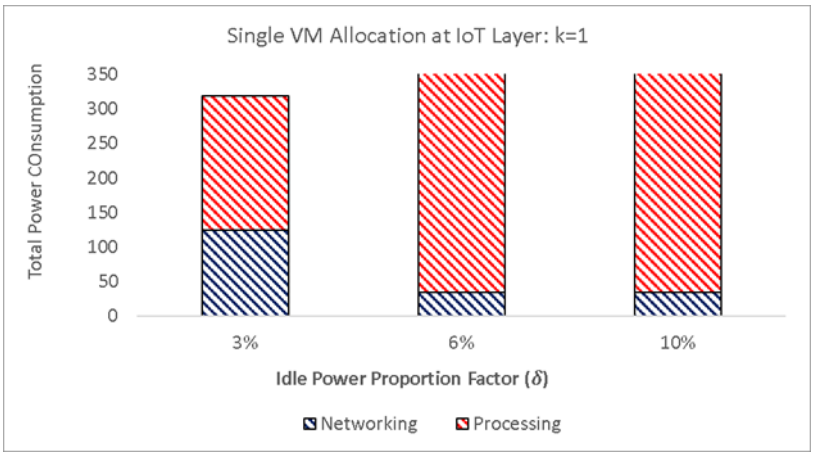



**Figure 4. 1: Total power consumption under different values of δ when k=1.**

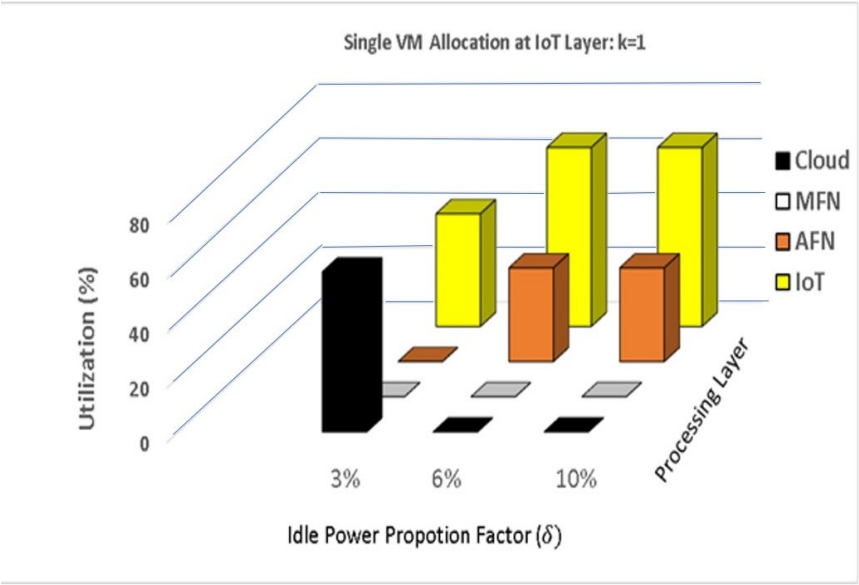



**Figure 4. 2: Workload distribution under different values of δ when k=1 (k the number of VMs that can be allocated to a single IoT node).**

### 4.3.2 Multi VM Allocation at IoT

In this scenario, we increase the value of paremer *k* and thus allow multiple VMs to be processed at an IoT node. Figure 4.3 shows the total power consumption of all nodes against different values of the $\delta$ factor when *k*=2. The results show that the different values of $\delta$ result in the same level of power consumption. Also, compared to the case where k=1, the total power consumption is reduced by more than 50%. Figure 4.4 shows the distribution of processing among the cloud, MFN, AFN, and the IoT layers for the case of k=2 and for different values of the $\delta$ factor. The results show that for all considered values of $\delta$, the IoT nodes are the preferred location to process all the demands. Hence, the higher than 50% reduction in the total power consumption for the case when k=2 compared to the case when k=1 is due to the ability of the IoT nodes to process more and having higher utilisation. Also,allowing two VMs to be processed by a single IoT device gives enough capacity for all the VMs to be processed at the IoT layer and therefore avoiding higher processing layers as seen in Figure 4.4. Figure 4.5 shows that the flexibility in the VM allocation scheme substantially achieve significant power savings up to 63% under all values of $\delta$ compared to the single VM allocation scenario (*k*=1).

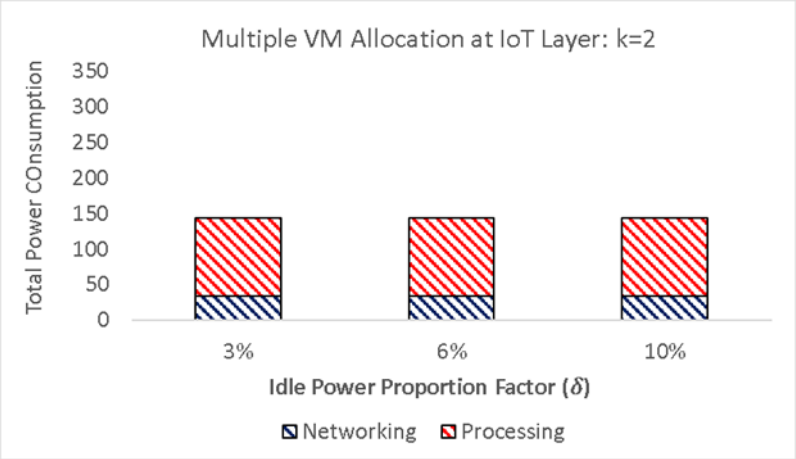



**Figure 4. 3: Total power consumption under different values of δ when k=2(all VMs allocated at IoT nodes when k>1).**

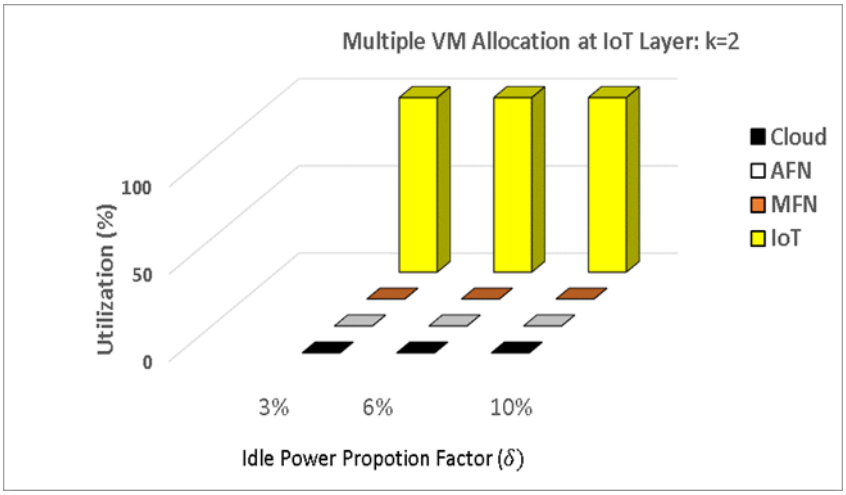



**Figure 4. 4: Workload distribution under different values of δ when k = 2.**

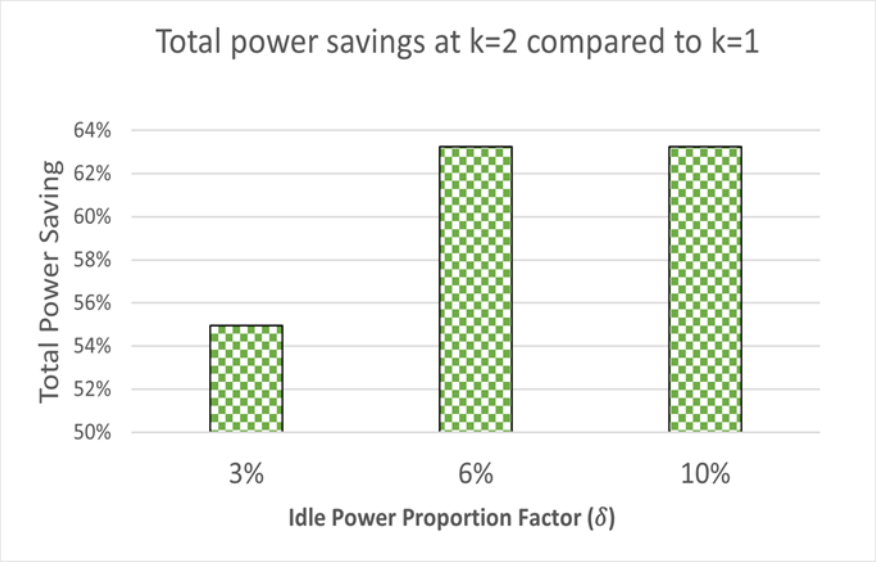



**Figure 4. 5: Total power savings achieved at k=2 compared to k=1.**

## 4.4 Summary

This chapter extended the studies in the previous chapter by evaluating the impact of constraining the VM allocation in IoT devices. It also evaluated the impact of the idle power of the network devices. It further examined the effect of varying the amount of idle power in core and access network devices attributed to ML / DNN applications (the proportion factor). The results showed that relaxing the VMs allocation constraint to allow two VMs to be embedded into a single IoT device allows all VSRs to be hosted in the IoT layer (for our set of network, processing and demand parameters) resulting in substantial power savings up to 63% compared to constraining VMs allocation in IoT devices to a single VM. The results also show that under the single VM allocation constraint, the increase in the idle power proportion factor to 6% and 10% resulted in 22% increase in total power consumption compared to an idle power proportion factor of 3%.

# Chapter 5: Sequential Job Placement in a Non-Pre-emptive Network

## 5.1 Introduction

In the previous chapters, we studied the embedding of DNN VSRs under the assumption that on the arrival of new requests, the network could be re-optimised by re-embedding the existing VSRs. We refer to this approach as a pre-emptive VSRs embedding scheme. In this chapter we study the embedding of VSRs assuming that requests are embedded sequentially, and the new requests are served on the spare capacity, hence existing VSRs will not be changed. We refer to this approach as a non-pre-emptive embedding scheme. This means that existing user's quality of experience (QoE) will not be impacted as a continued service is maintained. We study three objective functions: 1) a network centric approach whereby only the power consumption of the network is minimised, 2) a server centric approach whereby only the power consumption of processing is minimised, and 3) a hybrid approach whereby the network and the server power consumption are jointly minimised. The different objective functions give different approaches to handle scarcity of resources as sequential embedding of VSRs results in suboptimum utilisation of resources. In the previous chapters, all the VSRs were embedded successfully as the re-embedding of existing VSRs allows optimum utilisation of resources. With sequential embedding of VSRs, the suboptimum utilisation of resources might result in rejecting some VSRs. Hence, the constraints of the optimisation model allow rejection of VSRs and the optimisation objective function increases the number of accepted VSRs in addition to minimising the

power consumption. Also, to reduce the complexity of the optimisation model we study optimising the embedding over the IoT layer only.

## 5.2 MILP Model

We modified the MILP model in Chapter 4 to represent sequential embedding of VSRs in an architecture where VMs embedding is limited to the IoT layer. In the following we present the complete modified model for improved readability.

The following sets, parameters and variables are defined:

**Sets:**

| | |
|---|---|
| $\mathbb{I}$ | Set of IoT devices. |
| $\mathbb{P}$ | Set of processing nodes that can process a VSR, where $\mathbb{P} = \mathbb{I}$. |
| $\mathbb{IP}$ | Set of source node IoT devices, $\mathbb{IP} \subset \mathbb{I}$ |
| $\mathbb{R}$ | Set of VSRs. |
| $\mathbb{VM}_r$ | Set of VMs in VSR $r \in \mathbb{R}$. |
| $\mathbb{N}$ | Set of networking nodes (IoT devices, ONUs, OLTs,). |
| $\mathbb{N}_m$ | Set of neighbour nodes of node $\mathrm{m} \in \mathbb{N}$. |

**Parameters:**

$s\ and\ d$ Index the source and destination nodes of a virtual link in a VSR topology, $s, d \in \mathbb{VM}_r, r \in \mathbb{R}$.

$b\ and\ e$ Index source and destination processing nodes of an end to end traffic demand aggregated from embedding VSR, $b, e \in P, b \neq e$.

$m\ and\ n$ Index the end nodes of physical links.

$F^{r,s}$ Processing requested by node $s$ in in VSR $r \in \mathbb{R}$.

$H^{r,s,d}$ Data rate of virtual link $(s, d)$ in VSR $r \in \mathbb{R}$,

$P_s^r$ $\mathrm{P_s^r} = 1$ if VM $s \in VM_r$ in VSR $r \in \mathbb{R}$ is the input layer, otherwise $P_s^r = 0$.

$\Pi_n^{(net)}$ Maximum power consumption of network node $n \in \mathbb{N}$ accounting for all equipment in the node.

$\pi_n^{(net)}$ Idle power consumption of network node $n \in \mathbb{N}$ accounting for all equipment in the node.

$C_n^{(net)}$ Capacity of network node $n \in \mathbb{N}$.

$\epsilon_n$ Energy per bit of network node $n \in \mathbb{N}$, in J/Gb,

$\epsilon_n = \frac{\Pi_n^{(net)} - \pi_n^{(net)}}{C_n^{(net)}}$.

$\Pi_p^{(pr)}$ Maximum power consumption of a single server at processing node $n \in \mathbb{P}$.

$\pi_p^{(pr)}$ Idle power consumption of a single server at processor node $p \in \mathbb{P}$.

$C_p^{(cpu)}$ Processing capacity of a serve at processing node $p \in \mathbb{P}$.

$RC_p^{(cpu)}$ Residual processing capacity of a serve at processing node $p \in \mathbb{P}$.

$E_p$ Energy per FLOPS of processing node $p \in \mathbb{P}, E_p = \frac{\Pi_p^{(pr)} - \pi_p^{(pr)}}{C_p^{(cpu)}}$

$\delta_n$ Proportion of idle power consumed on high-capacity networking equipment $n \in N$.

$PUE_n^{(net)}$ Power Usage Effectiveness (PUE) factor of node $n \in N$ for networking.

$PUE_p^{(net)}$ Power Usage Effectiveness (PUE) factor of node $p \in P$ for processing.

$PU_p$ The amount of processing workload on node $p \in P$ for existing VSRs.

$TR_n$ The amount of traffic served by network node $n \in \mathbb{N}$ for existing VSRs.

$\gamma_r$ $\gamma_r = 1$ if request $r \in \mathbb{R}$ is accepted otherwise $\gamma_r = 0$.

**Variables:**

$\lambda^{b,e}$ Traffic demand between processing node pair $(b,e)$ aggregated after all VSRs are embedded, $b,e \in \mathbb{P}$.

$\lambda_{m,n}^{b,e}$ Traffic demand between processing node pair $(b,e) \in \mathbb{P}$ aggregated after all VSRs are embedded, traversing physical link $(m,n)$, $m \in \mathbb{N}$ and $n \in \mathbb{N}_m$.

$\lambda_n$ Amount of traffic originating/passing by/destined to network node $n \in \mathbb{N}$,

where $\lambda_n = \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m} \lambda_{m,n}^{b,e} + \sum_{b\in\mathbb{P}} \sum_{e\in\mathbb{P}:b\neq e} \sum_{m\in\mathbb{N}} \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{b,e}$ .

$\alpha_n$ $\alpha_n = 1$ if network node $n \in \mathbb{N}$ is activated, otherwise $\alpha_n = 0$.

$\Omega_p$ Amount of workload in FLOPS, allocated to processing node $p \in \mathbb{P}$.

$N_p$ Number of activated servers at processing node $p \in \mathbb{P}$.

$\Phi_p$ $\Phi_p = 1$ if processing node $p \in \mathbb{P}$ is activated, otherwise $\Phi_p = 0$.

$\delta_b^{r,s}$ $\delta_b^{r,s} = 1$ if VM $s \in VM_r$ in VSR $r \in \mathbb{R}$ is embedded into processing node b$\in P$, otherwise $\delta_b^{r,s} = 0$.

$w_{b,e}^{r,s,d}$ $w_{b,e}^{r,s,d}$ is the XOR of $\delta_b^{r,s}$ and $\delta_e^{r,d}$, i.e. $w_{b,e}^{r,s,d} = \delta_b^{r,s} \oplus \delta_e^{r,d}$.

$\rho_{b,e}^{r,s,d}$ $\rho_{b,e}^{r,s,d} = 1$ if the virtual nodes $s, d \in \mathbb{VM}_r$ in VSR $r \in \mathbb{R}$ are successfully embedded in processing nodes $b, e \in \mathbb{P}$ respectively and a link between processing nodes $b, e$ is established if a virtual link exists between virtual nodes $s, d$, otherwise $\rho_{b,e}^{r,s,d} = 0$.

The total power consumption comprises of two parts: 1) network power consumption, 2) processing power consumption.

The adopted power profile consists of a proportional part and idle part. The proportional part increases with the volume of workload, whilst the idle part is consumed as soon as the device is activated. We assume that any unused equipment is switched off completely.

The network power consumption resulting from embedding the newly arriving VSR is given by:

$$\sum_{n \in \mathbb{N}} PUE_n^{(net)} \epsilon_n \lambda_n + \sum_{\substack{n \in \mathbb{N}: \\ TR_n = 0}} PUE_n^{(net)} \alpha_n \pi_n^{(net)} \delta_n . \quad (5.1)$$

The first term of the above expression is the proportional power consumption of the networking equipment whilst the second term calculates the idle power consumption of these equipment. The condition $TR_n = 0$ ensures that we do not duplicate the idle power consumption of a network node that has been activated previously.

The processing power consumption of IoT nodes includes the power consumed by the servers as given below:

$$\sum_{p\in\mathbb{P}} PUE_p^{(pr)} E_p \Omega_p + \sum_{\substack{p\in\mathbb{P}: \\ PU_p=0}} PUE_p^{(pr)} N_p \pi_p^{(pr)} \tag{5.2}$$

The first term of the above expression is the proportional power consumption of the servers whilst the second term calculates the idle power consumption of these servers. The condition $PU_u = 0$ ensures that idle power consumption of the processing nodes activated to embed pervious VSRs is not duplicated.

Three objective functions are defined as given below.

With sequential embedding of VSRs, the suboptimum utilisation of resources might result in rejecting some VSRs. Hence, the optimisation model allows rejection of VSRs and the optimisation objective function increases the number of accepted VSRs in addition to minimising the power consumption.

**The Server Centric Approach**

Minimise the processing power consumption only given as:

$$\sum_{p\in\mathbb{P}} PUE_p^{(pr)} E_p \Omega_p + \sum_{\substack{p\in\mathbb{P}: \\ PU_p=0}} PUE_p^{(pr)} N_p \pi_p^{(pr)} - \gamma \tag{5.3}$$

**The Network Centric Approach**

Minimise the network power consumption only given as:

$$\sum_{n\in\mathbb{N}} PUE_n^{(net)} \epsilon_n \lambda_n + \sum_{\substack{n\in\mathbb{N}: \\ TR_n=0}} PUE_n^{(net)} \alpha_n \pi_n^{(net)} \delta_n - \gamma \tag{5.4}$$

**The hybrid Approach**

Minimise the total power consumption given as:

$$\sum_{p\in\mathbb{P}} PUE_p^{(pr)} E_p \Omega_p + \sum_{\substack{p\in\mathbb{P}: \\ PU_p=0}} PUE_p^{(pr)} N_p \pi_p^{(pr)} + \sum_{p\in\mathbb{P}} PUE_p^{(pr)} E_p \Omega_p \tag{5.5}$$

$$+ \sum_{\substack{p\in\mathbb{P}: \\ PU_p=0}} PUE_p^{(pr)} N_p \pi_p^{(pr)} - \gamma$$

**Subject to:**

$$\sum_{b\in\mathbb{P}} \delta_b^{r,s} = \gamma_r \qquad \forall r \in \mathbb{R}, s \in \mathbb{VM}_r \colon \mathrm{P}_s^{\mathrm{r}} \neq 1 \tag{5.6}$$

Constraint (5.6) ensures that VMs of an accepted VSR, except for input VMs, are embedded into any of the processing nodes.

$$\sum_{b\in\mathbb{IP}} \delta_b^{r,s} = \gamma_r \qquad \forall r \in \mathbb{R}, s \in \mathbb{VM}_r \colon \mathrm{P}_s^{\mathrm{r}} = 1 \tag{5.7}$$

Constraint (5.7) ensures that input VMs of an accepted VSR are embedded into data source IoT devices only.

$$\sum_{n\in\mathbb{N}_m} \lambda_{m,n}^{b,e} - \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{b,e} = \begin{cases} \lambda^{b,e} & m = s \\ -\lambda^{b,e} & m = d \\ 0 & otherwise \end{cases} \tag{5.8}$$

$$\forall b, e \in \mathbb{P}, d \in \mathbb{P}, m \in \mathbb{N} \colon b \neq e.$$

Constraint (5.8) preserves the flow of traffic in the network.

$$\sum_{b\in\mathbb{P}} \sum_{s\in\mathbb{VM}_r} \delta_b^{r,s} F^{r,s} = \sum_{s\in\mathbb{VM}_r} F^{r,s} \gamma_r \qquad \forall r \in \mathbb{R} \tag{5.9}$$

Constraint (5.9) ensures that the processing demand of request $r \in \mathbb{R}$ is fulfilled.

$$\delta_b^{r,s} + \delta_e^{r,d} = w_{b,e}^{r,s,d} + 2\rho_{b,e}^{r,s,d} \tag{5.10}$$

$$\forall r \in \mathbb{R}, (s,d) \in \mathbb{VM}_r, (b,e) \in \mathbb{P}: b \neq e, s \neq d$$

Constraint (5.10) ensures that virtual nodes connected in the VSR topology are also connected on the physical network. This done by introducing a binary variable $w_{b,e}^{r,s,d}$ that is only equal to 1 if $\delta_b^{r,s}$and $\delta_e^{r,d}$ e are exclusively equal to 1, otherwise $w_{b,e}^{r,s,d} = 0$.

$$\sum_{r \in \mathbb{R}} \sum_{s \in \mathbb{VM}_r} \sum_{\substack{d \in \mathbb{VM}_r: \\ s \neq d}} H^{r,s,d} \rho_{b,e}^{r,s,d} = \lambda^{b,e} \quad \forall (b,e) \in \mathbb{P}: b \neq e \tag{5.11}$$

Constraint (5.11) ensures that the data rate requirement of virtual links are fulfilled.

$$\Omega_p \leq RC_p^{(cpu)} \qquad \forall p \in \mathbb{P} \tag{5.12}$$

Constraints (5.12) ensures that processing handled by an IoT node does not exceed the residual processing capacity of the IoT node server (Note that the IoT node has one server).

$$\lambda_n \geq \alpha_n \qquad \forall n \in \mathbb{N} \tag{5.13}$$

$$\lambda_n \leq M\alpha_n \quad \forall n \in \mathbb{N} \tag{5.14}$$

Constraints (5.13) and (5.14) relate the binary variable $\alpha_n$ to the continuous variable $\lambda_n$, i.e. determines if a network node is activated or not based on the traffic traversing/generated by the node.

$$\sum_{r\in\mathbb{R}}\sum_{s\in\mathbb{VM}_r}\delta_b^{r,s}\geq\Phi_p\quad\forall p\in\mathbb{P} \tag{5.15}$$

$$\sum_{r\in\mathbb{R}}\sum_{s\in\mathbb{VM}_r}\delta_b^{r,s}\leq M\Phi_p\quad\forall p\in\mathbb{P} \tag{5.16}$$

Constraints (5.15) and (5.16) determine the binary value of $\Phi_p$, i.e. determine if a processing node is activated based on the amount of processing performed by the node.

## 5.3 Results and Discussions

As explained, in this chapter we consider an architecture where processing takes place at the IoT layer only to reduce the complexity of the MILP model.

We consider a scenario where input VMs are embedded in a single source node IoT device. We consider power consumption values of Section 3.5.1 (Table 3.1 and Table 3.2). We assume that in total, there are 20 IoT devices. We have distributed the IoT devices among four zones such that Zone 1 and Zone 2 comprise of 6 IoT devices each whilst Zone 3 and Zone 4 comprise of 4 IoT devices each. One of the IoT device in Zone 1 acts as a data source to collect data.

We examine the sequential embedding of 20 VSRs arriving one at a time. The number of VMs per VSR is randomly distributed between 2–4 VMs. The processing workload of input VMs is randomly distributed between 0.1 – 1 GFLOPS. The hidden layer VM are randomly distributed between 10 – 20 Mbps. These data rates are representative of high-resolution image/video files that are exchanged between the NN layers [118]. The virtual links data rate is randomly distributed between 0.1-2 Mbps.

Moreover, we have made the assumption that the processing efficiency at the IoT layer is heterogeneous. The IoT nodes that reside at the source node's zone (IoT 1 – IoT 6) have a lesser processing efficiency than the IoT nodes that are located in the other IoT zones. Figure 5.1 shows the processing efficiency of all the IoT devices in Watts / Flops. Having a heterogeneous processing efficiency across the IoT zones is expected to give interesting trade-offs between minimising the network power consumption and minimising the processing power consumption. Table 5.1 and Table 5.2 give the processing and networking devices power consumption.

**Table 5. 1: Processing device parameters**

| Devices | Max(W) | Idle(W) | GFLOPS | Efficiency (W/GFLOPS) |
|---|---|---|---|---|
| IoT (Rpi 4 B 4GB) | 7.3 [117] | 2.56 [117] | 13.5 [117] | 0.35 |

**Table 5. 2: Networking devices parameters**

| Devices | Max (W) | Idle (W) | Bitrate (Gbps) | Efficiency (J/Gb) |
|---|---|---|---|---|
| IoT Wi-Fi interface | 0.56 [117] | 0.34 [117] | 0.1 [117] | 2.2 |
| ONU ( including Wi-Fi interface) | 15 [114] | 9 [114] | 10 [114] | 0.6 |
| OLT | 1940 [114] | 60 [114] | 8600 [114] | 0.22 |

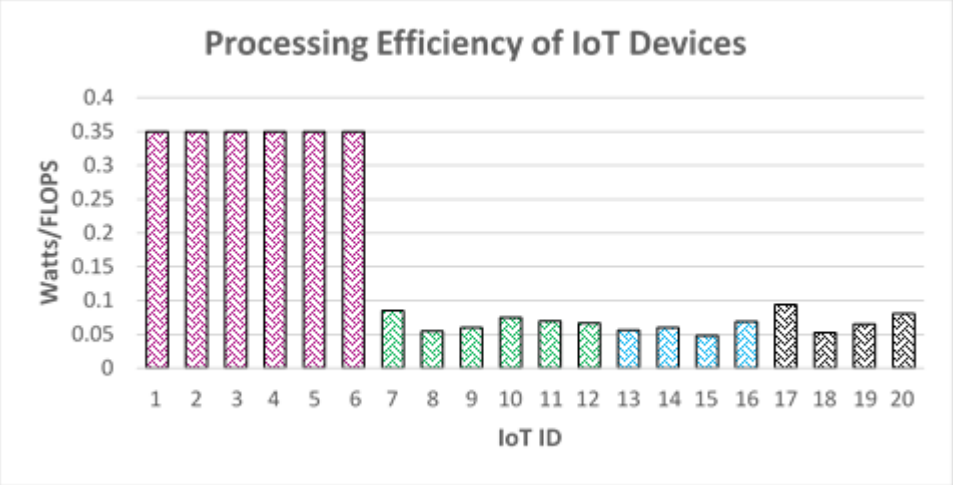



**Figure 5. 1: Processing efficiency at the IoT layer**

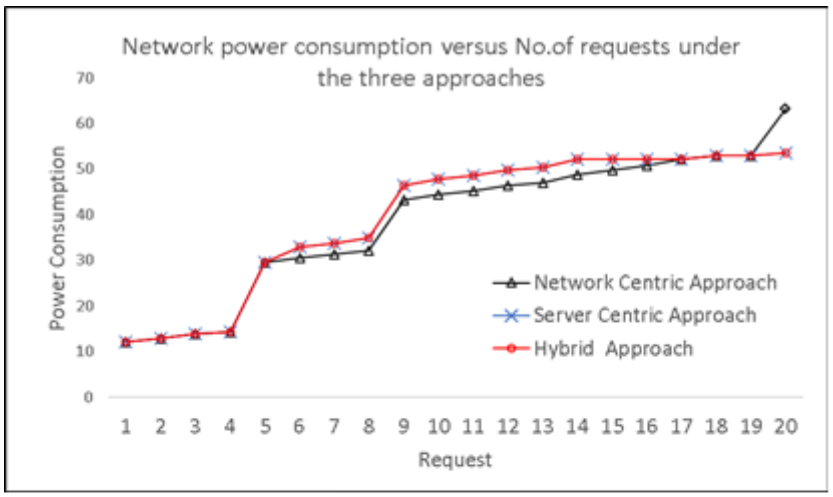
The results in Figure 5.2 show the network power consumption and the processing power consumption versus the number of requests under the network centric, server centric and hybrid approaches. The requests arrive one at a time and the existing requests are not reconfigured when allocating arriving requests. As expected, the network centric approach has resulted in

the minimum network power consumption as the VMs are embedded on the IoT devices that can be energy-efficiently connected regardless of the devices processing efficiency. The server centric and the hybrid approaches performed similarly as far as network power consumption is concerned as the processing power consumption is higher than the network power consumption. Compared to the hybrid approach and server centric approach, the network centric approach has saved up to 8% of the network power consumption.

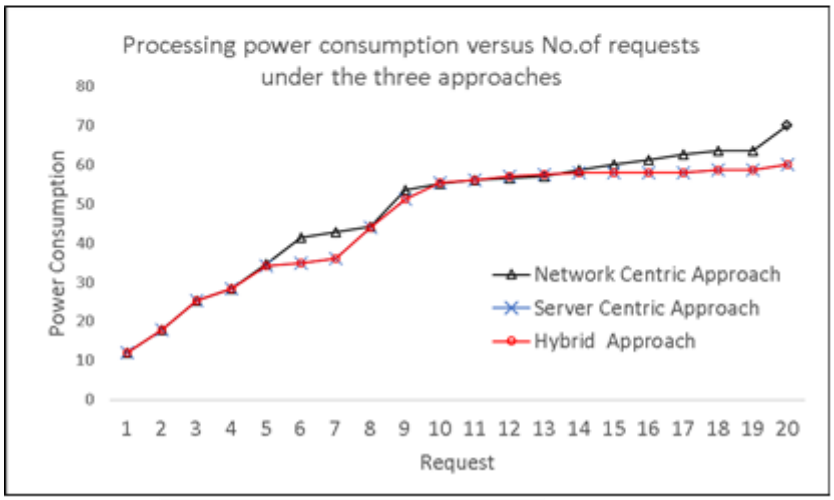
As for the processing power consumption, the network centric approach has resulted in a maximum increase of 16% compared to the server centric approach. As was previously mentioned and shown in Figure 5.1, the most efficient IoT devices were placed at zones other than the source node. Hence, with the server centric approach, the model is able to make better use of the processing IoT devices by packing them more efficiently regardless of the network overhead.

In addition to the power consumption metric, we have also shown the number of accepted requests in Figure5.3 under all the three approaches for the sequential embedding scheme and the pre-emptive embedding scheme. The architecture processing and networking resources were sufficient to host all the 20 requests under the pre-emotive embedding with hybrid approach. The server centric and the hybrid approaches have both rejected 4 requests. The sequential embedding network centric approach has resulted in accepting 19 out of 20 requests. The improved performance of the network centric approach can be attributed to the better utilisation of the networking resources which becomes a bottleneck for the processing centric approach and hybrid

approach as they attempt to place the requests on the most efficient IoT devices therefore depleting the capacity of certain Wi-Fi APs.




(a)




(b)

**Figure 5. 2: (a) Network power consumption under the different approaches, (b) processing power consumption under the different approaches**

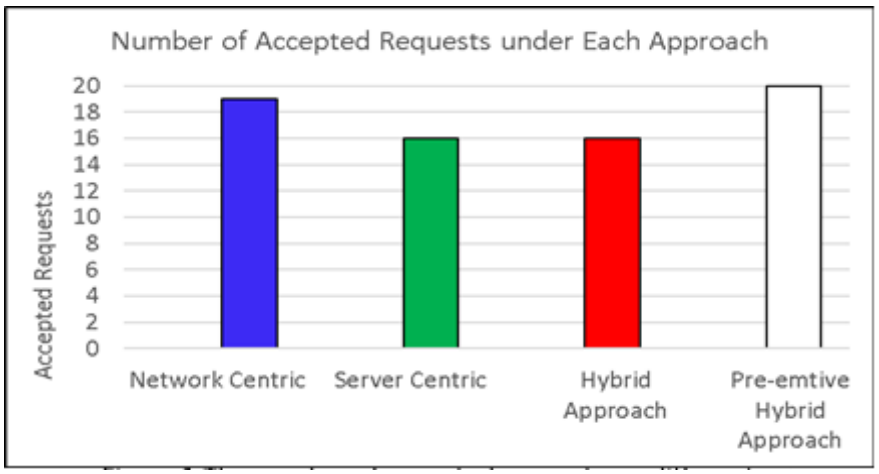



**Figure 5. 3: The number of accepted requests vs. different approaches**

## 5.4 Summary

This chapter studied the embedding of DNNs VSRs in a non-pre-emptive scenario in an architecture of IoT devices only. We considered a scenario where requests are embedded sequentially, and the existing requests are not reconfigured. We considered three energy minimisation approaches to handle scarcity of resources: a network centric approach, a server centric approach and a hybrid approach. The results showed that the suboptimal utilisation of resources with sequential embedding resulted in blocking some requests. The network centric approach was shown to be the most efficient in serving requests under sequential embedding with a maximum increase in processing power consumption of 16%.

# Chapter 6: Conclusions and Future Work

In this chapter, the contributions made in this thesis will be summarised. The chapter will also suggest a number of topics for future research.

## 6.1 Conclusions

The work in this thesis has studied the energy efficient embedding of DNN VSRs over a heterogeneous CFN architecture under different scenarios and constraints to reflect DNN algorithm topologies and IoT capabilities.

In Chapter 3, we investigated the power consumption associated with the embedding of DNN VSRs over a CFN architecture. The studies in this chapter were divided into two scenarios. Under Scenario 1, we looked at a scenario where DNN VSRs have a single IoT device generating data for the input layer of the DNN VSRs, whilst Scenario 2 considered multiple IoT devices generating data for the input layer. A MILP model is developed to optimise the embedding of DNN VSRs under the two scenarios. For Scenario 1, the optimisation of embedding over the CFN architecture resulted in up to 68% power savings (12% on average) compared to processing all the VMs in the cloud. For Scenario 2, the optimum embedding over the CFN architecture produced up to 60% power savings (10% on average). For both scenarios the results also showed that despite the number of source nodes and their geographical deployment, the distribution of the VMs among the IoT devices is favoured over consolidating them in higher capacity fog servers that were associated with a higher PUE value. Furthermore, we evaluated the energy efficiency of a fog architecture in embedding DNN VSRs. The results showed

that optimising the embedding over the fog architecture resulted in up to 44% (19% on average) and up to 20% (10% on average) increase in total power consumption compared to the optimum embedding over the CFN architecture considering Scenario 1 and Scenario 2, respectively. This limited increase in power consumption indicates that improvement in the energy efficiency of fog nodes by similar magnitude, which is anticipated in the near future, would allow efficient use of fog nodes in embedding DNN VSRs.

Chapter 4 extended the studies in Chapter 3 by studying the introduction of constrains on the VM allocation in IoT devices and evaluating the impact of the idle power proportion factor attributed to the DNN applications. The results showed that relaxing the VMs allocation constraint to allow two VMs to be embedded into a single IoT device allows all VSRs to be hosted in the IoT layer resulting in power savings up to 63% compared to constraining VMs allocation in IoT devices to a single VM. The results also show that under a single VM allocation constraint, the increase in idle power proportion factor to 6% and 10% has resulted in 22% increase in total power consumption compared to an idle power proportion factor of $3\%$.

Chapter 6 studied the embedding of DNNs VSRs in a non-pre-emptive scenario where newly arriving VSRs are embedded without reconfiguring existing VSRs. An architecture that contains IoT devices only was considered. We considered three energy minimisation approaches to handle scarcity of resources in IoT networks: A network centric approach, a server centric approach and a hybrid approach. The results showed that the network centric approach is the most efficient in serving requests and reducing blocking resulting from the suboptimal utilisation of resources with sequential

embedding. The increase in processing power consumption resulting from adopting the network centric approach is limited to a maximum of 16%

## 6.2 Future Work

This thesis provides a framework that can be the basis of further studies. In the following we suggest some future research directions:

### 6.2.1 Delay sensitive services

The embedding of DNN VSRs that serve delay sensitive applications and ultra-delay sensitive applications such as healthcare applications is an interesting topic to investigate. Serving such VSRs will require giving them priority over non delay sensitive applications and opting for less efficient processing nodes at the fog layer to ensure minimum delay. It would be interesting to study the trade-off between delay and power consumption in embedding delay sensitive DNN VSRs.

### 6.2.2 Scarce energy sources for IoT devices

IoT devices are typically either battery powered or powered by renewable energy. Hence, the scarce energy sources need to be efficiently managed to prolong the lifetime of the service provided by the IoT nodes and avoid service disruption. It would be interesting to consider a constraint on the energy availability of IoT devices when studying the embedding of DNN VSRs.

### 6.2.3 Heuristics Algorithms

The MILP models in this thesis provide the optimal solution however they are very complex to execute, hence deploying these models in real-time scenarios is not practical. Therefore, it would be of interest to design heuristic-based algorithms of reduced computational complexity to enable real-time DNN VSRS embedding.

# References

[1] M. Kumar, X. Zhang, L. Liu, Y. Wang, and W. Shi, “Energy-efficient machine learning on the edges,” *Proc. - 2020 IEEE 34th Int. Parallel Distrib. Process. Symp. Work. IPDPSW 2020*, pp. 912–921, 2020, doi: 10.1109/IPDPSW50202.2020.00153.

[2] A. Yousefpour *et al.*, “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” *J. Syst. Archit.*, no. December 2018, 2019, doi: 10.1016/j.sysarc.2019.02.009.

[3] R. Du, S. Magnússon, and C. Fischione, “The Internet of Things as a Deep Neural Network,” *arXiv*, no. September, pp. 20–25, 2020.

[4] V. Sze, Y. H. Chen, T. J. Yang, and J. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *arXiv*, vol. 105, no. 12, pp. 2295–2329, 2017.

[5] Y. Wang *et al.*, “Benchmarking the Performance and Energy Efficiency of AI Accelerators for AI Training,” *Proc. - 20th IEEE/ACM Int. Symp. Clust. Cloud Internet Comput. CCGRID 2020*, pp. 744–751, 2020, doi: 10.1109/CCGrid49817.2020.00-15.

[6] E. Di Pascale, I. Macaluso, A. Nag, M. Kelly, and L. Doyle, “The Network As a Computer: A Framework for Distributed Computing over IoT Mesh Networks,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2107–2119, 2018, doi: 10.1109/JIOT.2018.2823978.

[7] N. Kaminski *et al.*, “A neural-network-based realization of in-network computation for the Internet of Things,” *IEEE Int. Conf. Commun.*, pp. 1–6, 2017, doi: 10.1109/ICC.2017.7996821.

[8] J. Li, W. Liang, Y. Li, Z. Xu, X. Jia and S. Guo, "Throughput Maximization of Delay-Aware DNN Inference in Edge Computing by Exploring DNN Model Partitioning and Inference Parallelism," in *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2021.3125949.

[9] W. Miao, Z. Zeng, L. Wei, S. Li, C. Jiang and Z. Zhang, "Adaptive DNN Partition in Edge Computing Environments," *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, Hong Kong, 2020,

pp. 685-690, doi: 10.1109/ICPADS51040.2020.00097.

[10] T. Liu and D. Lu, “The application and development of IoT,” in *2012 International symposium on information technologies in medicine and education*, 2012, vol. 2, pp. 991–994.

[11] J. F. Pagel and P. Kirshtein, *Machine dreaming and consciousness*. Academic Press, 2017.

[12] O. Vermesan *et al.*, “Internet of things strategic research roadmap,” *Internet things-global Technol. Soc. trends*, vol. 1, no. 2011, pp. 9–52, 2011.

[13] Cisco.com, “The Internet of Things How the Next Evolution of the Internet Is Changing Everything.’’

[14] D. Online, “IoT will grow to 26 billion units installed by 2020, says Gartner.” Accessed 2022.

[15] W. P. P. 2019, “Population.un.org.” Accessed 2019.

[16] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, “Estimation of energy consumption in machine learning,” *J. Parallel Distrib. Comput.*, vol. 134, pp. 75–88, 2019, doi: 10.1016/j.jpdc.2019.07.007.

[17] L. Tian, H. Wang, Y. Zhou, and C. Peng, “Video big data in smart city: Background construction and optimization for surveillance video processing,” *Futur. Gener. Comput. Syst.*, vol. 86, pp. 1371–1382, 2018, doi: 10.1016/j.future.2017.12.065.

[18] S. Sarkar, S. Chatterjee, and S. Misra, “Assessment of the Suitability of Fog Computing in the Context of Internet of Things,” *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, 2015.

[19] M. G. Institute, “By 2025, Internet of things applications could have $11 trillion impact” . [Online]. Available: https://www.mckinsey.com/mgi/overview/in-the-news/by-2025-internet-of-things-applications-could-have-11-trillion-impact.”

[20] E. Newton, “How to Optimise Your IoT Device’s Power Consumption,” vol. 0, no. 0.

[21] S. Zeadally, S. U. Khan, and N. Chilamkurti, “Energy-efficient networking: past, present, and future,” *J. Supercomput.*, vol. 62, no. 3, pp. 1093–1118, 2012, doi: 10.1007/s11227-011-0632-2.

[22] R. Deng, R. Lu, C. Lai, and T. H. Luan, “Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing,” *IEEE Int. Conf. Commun.*, vol. 2015-Septe, pp. 3909–3914, 2015, doi: 10.1109/ICC.2015.7248934.

[23] M. Burhan, R. A. Rehman, B. Khan, and B.-S. Kim, “IoT elements, layered architectures and security issues: A comprehensive survey,” *Sensors*, vol. 18, no. 9, p. 2796, 2018.

[24] J. Jimenez, M. Koster, and H. Tschofenig, “IPSO smart objects,” 2016.

[25] E. Borgia, “The Internet of Things vision: Key features, applications and open issues,” *Comput. Commun.*, vol. 54, pp. 1–31, 2014.

[26] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[27] J. Cockerham, “What Are Sensors & Actuators? | Differences & Use-Cases.” 2022.

[28] A. F. Molisch *et al.*, “IEEE 802.15. 4a channel model-final report,” *IEEE P802*, vol. 15, no. 04, p. 662, 2004.

[29] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, “Choices for interaction with things on Internet and underlying issues,” *Ad Hoc Networks*, vol. 28, pp. 68–90, 2015.

[30] “iPhone 13 - Apple (UK).”, “Apple Storeg,” *Apple*. 2022.

[31] D. De Donno, L. Catarinucci, A. Di Serio, and L. Tarricone, “A long-range computational RFID tag for temperature and acceleration sensing applications,” *Prog. Electromagn. Res. C*, vol. 45, pp. 223–235, 2013.

[32] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, “Internet of Things (IoT): A vision, architectural elements, and security issues,” in *2017*

*international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017, pp. 492–496.

[33] P. Hooda, “Comparison - Centralized, Decentralized and Distributed Systems - GeeksforGeeks.” geeksforgeeks.org, 2022.

[34] M. Aazam and E.-N. Huh, “Fog computing and smart gateway based communication for cloud of things,” in *2014 International Conference on Future Internet of Things and Cloud*, 2014, pp. 464–470.

[35] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, “Semantics for the Internet of Things: early progress and back to the future,” *Int. J. Semant. Web Inf. Syst.*, vol. 8, no. 1, pp. 1–21, 2012.

[36] L. Tan and N. Wang, “Future internet: The internet of things,” in *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, 2010, vol. 5, pp. V5----376.

[37] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, “Research on the architecture of Internet of Things,” in *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, 2010, vol. 5, pp. V5----484.

[38] Z. Abbas and W. Yoon, “A survey on energy conserving mechanisms for the internet of things: Wireless networking aspects,” *Sensors (Switzerland)*, vol. 15, no. 10, pp. 24818–24847, 2015, doi: 10.3390/s151024818.

[39] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, “A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 416–464, 2018, doi: 10.1109/COMST.2017.2771153.

[40] P. Hu, S. Dhelim, H. Ning, and T. Qiu, “Survey on fog computing: architecture, key technologies, applications and open issues,” *J. Netw. Comput. Appl.*, vol. 98, no. September, pp. 27–42, 2017, doi: 10.1016/j.jnca.2017.09.002.

[41] A. Botta, W. de Donato, V. Persico, and A. Pescapé, “Integration of Cloud computing and Internet of Things: A survey,” *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016, doi: https://doi.org/10.1016/j.future.2015.09.021.

[42] M. Chiang and T. Zhang, “Fog and IoT: An Overview of Research Opportunities,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, 2016, doi: 10.1109/JIOT.2016.2584538.

[43] P. Sethi and S. R. Sarangi, “Internet of things: architectures, protocols, and applications,” *J. Electr. Comput. Eng.*, vol. 2017, 2017.

[44] L. Zhang, Y.-C. Liang, and M. Xiao, “Spectrum sharing for Internet of Things: A survey,” *IEEE Wirel. Commun.*, vol. 26, no. 3, pp. 132–139, 2018.

[45] M. Aazam, I. Khan, A. A. Alsaffar, and E. N. Huh, “Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved,” *Proc. 2014 11th Int. Bhurban Conf. Appl. Sci. Technol. IBCAST 2014*, pp. 414–419, 2014, doi: 10.1109/IBCAST.2014.6778179.

[46] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future internet: The internet of things architecture, possible applications and key challenges,” *Proc. - 10th Int. Conf. Front. Inf. Technol. FIT 2012*, pp. 257–260, 2012, doi: 10.1109/FIT.2012.53.

[47] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[48] D. Macedo, L. A. Guedes and I. Silva, "A dependability evaluation for internet of things incorporating redundancy aspects," in Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference On, 2014, pp. 417-422.

[49] I. Silva, R. Leandro, D. Macedo and L. A. Guedes, "A dependability evaluation tool for the Internet of Things," Comput. Electr. Eng., vol. 39, pp. 2005-2018, 2013.

[50] N. Maalel, E. Natalizio, A. Bouabdallah, P. Roux and M. Kellil, "Reliability for emergency applications in internet of things," in Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference On, 2013, pp. 361-366.

[51] J. Kempf, J. Arkko, N. Beheshti and K. Yedavalli, "Thoughts on reliability in the internet of things," in Interconnecting Smart Objects with the Internet Workshop, 2011, pp. 1-4.

[52] L. Li, Z. Jin, G. Li, L. Zheng and Q. Wei, "Modeling and analyzing the reliability and cost of service composition in the IoT: A probabilistic approach," in Web Services (ICWS), 2012 IEEE 19th International Conference On, 2012, pp. 584-591.

[53] F. Ganz, Ruidong Li, P. Barnaghi and H. Harai, "A resource mobility scheme for service-continuity in the internet of things," in Green Computing and Communications (GreenCom), 2012 IEEE International Conference On, 2012, pp. 261-264.

[54] Z. Zhu, L. Zhang and R. Wakikawa, "Supporting mobility for internet cars," Communications Magazine, IEEE, vol. 49, pp. 180-186, 2011.

[55] S. Misra and P. Agarwal, "Bio-inspired group mobility model for mobile ad hoc networks based on bird-flocking behavior," *Soft Computing,* vol. 16, pp. 437-450, 2012.

[56] O. Vermesan, P. Friess, and Others, *Internet of things-from research and innovation to market deployment*, vol. 29. River publishers Aalborg, 2014.

[57] W. Leister and T. Schulz, “Ideas for a Trust Indicator in the Internet of Things,” 2012.

[58] S. F. Abedin, M. G. R. Alam, R. Haw, and C. S. Hong, “A system model for energy efficient green-IoT network,” *2015 Int. Conf. Inf. Netw.*, pp. 177–182, 2015, doi: 10.1109/ICOIN.2015.7057878.

[59] A. Galanopoulos, T. Salonidis and G. Iosifidis, "Cooperative Edge Computing of Data Analytics for the Internet of Things," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1166-1179, Dec. 2020, doi: 10.1109/TCCN.2020.3019610.

[60] C. Perera, D. S. Talagala, C. H. Liu, and J. C. Estrella, "Energy-Efficient Location and Activity-Aware On-Demand Mobile Distributed Sensing Platform for Sensing as a Service in IoT Clouds," *IEEE Transactions on Computational Social Systems,* vol. 2, no. 4, pp. 171-181, 2015.

[61] J.-M. Liang, J.-J. Chen, H.-H. Cheng, and Y.-C. Tseng, "An energy-efficient sleep scheduling with QoS consideration in 3GPP LTE-advanced networks for internet of things," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 3, no. 1, pp. 13-22, 2013.

[62] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol

for wireless sensor networks," in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002, vol. 3: IEEE, pp. 1567-1576.

[63] M. Moreno, B. Úbeda, A. Skarmeta, and M. Zamora, "How can we tackle energy efficiency in iot basedsmart buildings?," *Sensors,* vol. 14, no. 6, pp. 9582-9614, 2014.

[64] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express,* vol. 5, no. 1, pp. 1-7, 2019.

[65] C. Gray, R. Ayre, K. Hinton, and R. S. Tucker, “Power consumption of IoT access network technologies,” *2015 IEEE Int. Conf. Commun. Work.*, pp. 2818–2823, 2015, doi: 10.1109/ICCW.2015.7247606.

[66] T. G. Orphanoudakis, C. Matrakidis, and A. Stavdas, “Next generation optical network architecture featuring distributed aggregation, network processing and information routing,” *2014 Eur. Conf. Networks Commun.*, pp. 1–5, 2014, doi: 10.1109/EuCNC.2014.6882669.

[67] J. M. NERI Fabio; FINOCHIETTO, “On the Energy Consumption of Relay Networks,” *Passive Optical Networks*.

[68] L. M. Camarinha-Matos, S. Tomic, and P. Graça, *Technological Innovation for the Internet of Things: 4th IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2013, Costa de Caparica, Portugal, April 15-17, 2013, Proceedings*, vol. 394. Springer, 2013.

[69] A. R. Biswas and R. Giaffreda, “IoT and Cloud Convergence: Opportunities and Challenges,” *2014 IEEE World Forum Internet Things*, pp. 375–376, 2014, doi: 10.1109/WF-IoT.2014.6803194.

[70] F. Lin, Q. Liu, X. Zhou, Y. Chen, and D. Huang, “Cooperative differential game for model energy-bandwidth efficiency tradeoff in the Internet of Things,” *China Commun.*, vol. 11, no. 1, pp. 92–102, 2014, doi: 10.1109/CC.2014.6821311.

[71] F. Jalali, S. Khodadustan, C. Gray, K. Hinton, and F. Suits, “Greening IoT with Fog: A Survey,” in *Proceedings - 2017 IEEE 1st International Conference*

*on Edge Computing, EDGE 2017*, Sep. 2017, pp. 25–31, doi: 10.1109/IEEE.EDGE.2017.13.

[72] S. H. Mohamed, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Energy efficiency of server-centric PON data center architecture for fog computing,” in *2018 20th International Conference on Transparent Optical Networks (ICTON)*, 2018, pp. 1–4.

[73] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017, doi: 10.1109/COMST.2017.2705720.

[74] B. J. Baliga, R. W. A. Ayre, K. Hinton, R. S. Tucker, and F. Ieee, “Green Cloud Computing : Balancing Energy in Processing , Storage , and Transport,” 2011.

[75] Z. Á. O. problems in fog MANN and E. computing, “Fog and edge computing: principles and paradigms,” pp. 103–121, 2019.

[76] et al ROCHA NETO Aluizio F., “Distributed machine learning for iot applications in the fog, Fog Computing: Theory and Practice.

[77] yourtechdiet, “‘AI vs. Machine Learning vs. Deep Learning’.[Online]. Available:https://yourtechdiet.com/blogs/ai-vs-machine-learning-vs-deep-learning.” Accessed 2022.

[78] IBM, “AI vs. Machine Learning vs. Deep Learning vs. neural networks: What’s the difference?,” IBM. [Online]. Available: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks,” Accessed 2022.

[79] Smartboost, “"Deep learning vs neural network: What’s the difference?,” smartboost.” .

[80] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.

[81] WiKi,AI “Supervised, Unsupervised,& Reinforcement Learning” https://machine-learning.paperspace.com/wiki/supervised-unsupervised-

and-reinforcement-learning, Accessed 2022.

[82] H. A. Alharbi, T. E. H. Elgorashi, and J. M. H. Elmirghani, “Energy efficient virtual machines placement over cloud-fog network architecture,” *IEEE Access*, vol. 8, pph. 94697–94718, 2020, doi: 10.1109/ACCESS.2020.2995393.

[83] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Distributed energy efficient clouds over core networks,” *J. Light. Technol.*, vol. 32, no. 7, pp. 1261–1281, 2014, doi: 10.1109/JLT.2014.2301450.

[84] 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 Networking and Processing,” *IEEE Access*, vol. 7, pp. 50697–50709, 2019, doi: 10.1109/ACCESS.2019.2911117.

[85] 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,” *J. Light. Technol.*, vol. 30, no. 12, pp. 1931–1942, 2012, doi: 10.1109/JLT.2012.2186557.

[86] X. Dong, A. Lawey, T. E. H. El-Gorashi and J. M. H. Elmirghani, "Energy-efficient core networks," *2012 16th International Conference on Optical Network Design and Modelling (ONDM)*, 2012, pp. 1-9, doi: 10.1109/ONDM.2012.6210196.

[87] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani, “Green IP over WDM networks with data centers,” *J. Light. Technol.*, vol. 29, no. 12, pp. 1861–1880, 2011, doi: 10.1109/JLT.2011.2148093.

[88] J. M. H. Elmirghani *et al.*, “GreenTouch GreenMeter Core Network Energy Efficiency Improvement Measures and Optimization [Invited],” *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 2, 2018, doi: 10.1364/JOCN.10.00A250.

[89] M. O. I. Musa, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Bounds on GreenTouch GreenMeter Network Energy Efficiency,” *J. Light. Technol.*, vol. 36, no. 23, pp. 5395–5405, Dec. 2018, doi: 10.1109/JLT.2018.2871602.

[90] M. Musa, T. Elgorashi, and J. Elmirghani, “Bounds for energy-efficient survivable IP over WDMnetworks with network coding,” *J. Opt. Commun. Netw.*, vol. 10, no. 5, pp. 471–481, May 2018, doi: 10.1364/JOCN.10.000471.

[91] M. Musa, T. Elgorashi, and J. Elmirghani, “Energy efficient survivable IP-Over-WDM networks with network coding,” *J. Opt. Commun. Netw.*, vol. 9, no. 3, pp. 207–217, Mar. 2017, doi: 10.1364/JOCN.9.000207.

[92] T. E. H. El-Gorashi, X. Dong, and J. M. H. Elmirghani, “Green optical orthogonal frequency-division multiplexing networks,” *IET Optoelectron.*, vol. 8, no. 3, pp. 137–148, 2014, doi: 10.1049/iet-opt.2013.0046

[93] B. G. Bathula, M. Alresheedi, and J. M. H. Elmirghani, “Energy Efficient Architectures for Optical Networks,” in *Proceedings IEEE London Communications Symposium, London,* 2009, pp. 5–8, Accessed: Jan. 08, 2020. [Online]. Available: http://www.ee.ucl.ac.uk/lcs/previous/LCS2009/LCS/lcs09_33.pdf

[94] B. G. Bathula and J. M. H. Elmirghani, "Energy Efficient Optical Burst Switched (OBS)Networks," 2009 IEEE Globecom Workshops, 2009, pp. 1-6, doi: 10.1109/GLOCOMW.2009.5360734.

[95] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani, “IP over WDM networks employing renewable energy sources,” *J. Light. Technol.*, vol. 29, no. 1, pp. 3–14, 2011, doi: 10.1109/JLT.2010.2086434

[96] N. I. Osman, T. El-Gorashi, L. Krug, and J. M. H. Elmirghani, “Energy-efficient future high-definition TV,” *J. Light. Technol.*, vol. 32, no. 13, pp. 2364–2381, Jul. 2014, doi: 10.1109/JLT.2014.2324634.

[97] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “BitTorrent content distribution in optical networks,” *J. Light. Technol.*, vol. 32, no. 21, pp. 3607–3623, Nov. 2014, doi: 10.1109/JLT.2014.2351074

[98] H. A. Alharbi, T. E. H. Elgorashi, and J. M. H. Elmirghani, “Impact of the Net Neutrality Repeal on Communication Networks,” *IEEE Access*, vol. 8, pp. 59787–59800, 2020, doi: 10.1109/ACCESS.2020.2983314

[99] A. M. Al-Salim, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Energy Efficient Big Data Networks: Impact of Volume and Variety,” *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 1, pp. 458–474, Mar. 2018, doi: 10.1109/TNSM.2017.2787624

[100] A. M. Al-Salim, T. E. H. El-Gorashi, A. Q. Lawey, and J. M. H. Elmirghani, “Greening big data networks: Velocity impact,” *IET Optoelectron.*, vol. 12, no. 3, pp. 126–135, Jun. 2018, doi: 10.1049/iet-opt.2016.0165.

[101] H. M. M. Ali, T. E. H. El-Gorashi, A. Q. Lawey, and J. M. H. Elmirghani, “Future Energy Efficient Data Centers with Disaggregated Servers,” *J. Light. Technol.*, vol. 35, no. 24, pp. 5361–5380, Dec. 2017, doi: 10.1109/JLT.2017.2767574.

[102] M. S. Hadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Patient-Centric HetNets Powered by Machine Learning and Big Data Analytics for 6G Networks,” *IEEE Access*, vol. 8, pp. 85639–85655, 2020, doi: 10.1109/ACCESS.2020.2992555.

[103] M. S. Hadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Patient-Centric Cellular Networks Optimization Using Big Data Analytics,” *IEEE Access*, vol. 7, pp. 49279–49296, 2019, doi: 10.1109/ACCESS.2019.2910224.

[104] I. S. B. M. Isa, T. E. H. El-Gorashi, M. O. I. Musa, and J. M. H. Elmirghani, “Energy efficient fog-based healthcare monitoring infrastructure,” *IEEE Access*, vol. 8, pp. 197828–197852, 2020, doi: 10.1109/ACCESS.2020.3033555.

[105] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Energy Efficient Virtual Network Embedding for Cloud Networks,” *J. Light. Technol.*, vol. 33, no. 9, pp. 1828–1849, 2015, doi: 10.1109/JLT.2014.2380777.

[106] H. Q. Al-Shammari, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Resilient Service Embedding in IoT Networks,” *IEEE Access*, vol. 8, pp. 123571–123584, 2020, doi: 10.1109/ACCESS.2020.3005936.

[107] H. Q. Al-Shammari, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Service Embedding in IoT Networks,” *IEEE Access*, vol. 8, pp. 2948–2962, 2020, doi: 10.1109/ACCESS.2019.2962271.

[108] J. C. Smith and Z. C. Taskin, “A Tutorial Guide to Mixed Integer Programming Models and Solution Techniques,” 2007.

[109] M. Pi´oro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[110] AMPL, “A Mathematical Programming Language - AMPL.” [Online]. Available: https://www.ampl.com/REFS/amplmod.pdf. [Accessed: 01-Oct-2022].

[111] Gros, S., Zanon, M., 2020. Reinforcement learning for mixed-integer problems based on mpc. IFAC-papersOnLine 53 (2), 5219-5224.

[112] G. S. G. Shen and R. S. Tucker, “Energy-Minimized Design for IP Over WDM Networks,” *IEEE/OSA J. Opt. Commun. Netw.*, vol. 1, no. 1, pp. 176–186, 2009, doi: 10.1364/JOCN.1.000176.

[113] V. Sze, Y. H. Chen, T. J. Yang, and J. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *arXiv*, vol. 105, no. 12, pp. 2295–2329, 2017.

[114] B. A. Yosuf, M. Musa, T. Elgorashi, and J. Elmirghani, “Energy Efficient Distributed Processing for IoT,” *IEEE Access*, vol. 8, pp. 161080–161108, 2020, doi: 10.1109/ACCESS.2020.3020744.

[115] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, “IoT-Cloud Service Optimization in Next Generation Smart Environments,” vol. 34, no. 12, pp. 4077–4090, 2016.

[116] “Research Computing, University of Leeds (ARC3).” https://arcdocs.leeds.ac.uk/welcome.html (accessed May 06, 2021).

[117] The GFLOPS/W of the various machines in the VMW Research Group.” http://web.eece.maine.edu/~vweaver/group/green_machines.html (accessed May 06, 2021).

[118] W. Y. B. Lim *et al.*, “Federated Learning in Mobile Edge Networks: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020, doi: 10.1109/COMST.2020.2986024.