Energy Efficient Big Data Networks

Ali Mahdi Ali Al-Salim

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

School of Electronic and Electrical Engineering

June 2018

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

Chapter 4 is based on the work from:

Ali M. Al-Salim; Ahmed Q. Lawey; Taisir El-Gorashi, and Jaafar M. H. Elmirghani, "Energy Efficient Tapered Data Networks for Big Data Processing in IP/WDM Networks," the 17th International Conference on Transparent Optical Networks (ICTON), 2015, pp. 1-5.

Prof Elmirghani, the supervisor, suggested the study of the energy efficiency of big data processing in IP/WDM networks. The co-supervisor Dr Lawey, worked with the student on the MILP model development and revised the paper. The co-supervisor, Dr El-Gorashi, revised the paper, and worked with the student on paper preparation. The PhD student developed the model, obtained and analysed the results, and wrote paper.

And:

Ali M. Al-Salim; Ahmed Q. Lawey; Taisir El-Gorashi, and Jaafar M. H. Elmirghani, "Greening Big Data Networks: Volume Impact," the 18th International Conference on Transparent Optical Networks (ICTON), 2016, pp. 1-6.

Prof Elmirghani, the supervisor, suggested the study of big data's volume impact on greening big data networks. The co-supervisor Dr Lawey, worked with the student on the MILP model development and revised the paper. The co-supervisor, Dr El-Gorashi, worked with the student on paper preparation. The PhD student developed the model, obtained and analysed the results, and wrote paper.

And

Al-Salim AM, Lawey AQ, El-Gorashi TE, Elmirghani JM. "Energy efficient big data networks: impact of volume and variety". IEEE Transactions on Network and Service Management (TNSM), pp. 458-474, 2017 Dec 28.

Prof Elmirghani, the supervisor, suggested the study of big data's volume and variety impact on greening big data networks. The co-supervisor Dr Lawey, worked with the student on the MILP model and heuristic development and revised the paper. The co-supervisor, Dr El-Gorashi, worked with the student on paper preparation. The PhD student developed the model, obtained and analysed the results, and wrote paper.

Chapter 5 is based on the work from:

Al-Salim, A., EL-Gorashi, T., Lawey, A. and Elmirghani, J.M., "Greening Big Data Networks: Velocity Impact". IET Optoelectronics. 2017 Nov 21.

Prof Elmirghani, the supervisor, suggested the study of big data's velocity impact on greening big data networks. The co-supervisor Dr Lawey, worked with the student on the MILP model development and revised the paper. The co-supervisor, Dr El-

Gorashi, worked with the student on paper preparation. The PhD student developed the model, obtained and analysed the results, and wrote paper.

Chapter 6 is based on the work from:

Al-Salim, A., El-Gorashi, T., Lawey, A.Q., and Elmirghani, J.M., "Greening Big Data Networks: Veracity Impact", Springer Journal of Photonic Communications, Accepted, March 2018.

Prof Elmirghani, the supervisor, suggested the study of big data's veracity impact on greening big data networks. The co-supervisor Dr Lawey, worked with the student on the MILP model development and revised the paper. The co-supervisor, Dr El-Gorashi, worked with the student on paper preparation. The PhD student developed the model, obtained and analysed the results, and wrote paper.

Acknowledgements

I would like to sincerely thank my supervisor, Professor Jaafar Elmirghani for his inestimable assistance and patience throughout my PhD journey. His insights and valuable advice paved the road toward accomplishing the goals of this work.

I am truly grateful to my co-supervisor, Dr. Ahmed Lawey, for the patient guidance, encouragement and immeasurable advice he has provided throughout my time as his student.

I would like to acknowledge Dr. T. H. El-Gorashi, my co-supervisor for her advice and useful discussions.

I would like to acknowledge the Higher Committee for Education Development in Iraq (HCED) for funding my Ph.D.

I am indebted to my two angels, my daughter Zainab, my son Mahdi; and my beloved wife Howraa for their everlasting support and love through all stages of my life and especially during my Ph.D.

I wish to express my love and gratitude to all my family, specially my beloved mother, Rajiha, and my wonderful father, Mahdi, my soul mate sister, Anfal, my kind brothers Salwan, Radhwan and Karrar, my father in law, Mahdi, my mother in law, Najat, for their support, patience and understanding during my Ph.D.

Finally, I thank all my friends with special thanks to all my colleagues for their friendship, input and valuable discussions.

Abstract

The continuous increase of big data applications in number and types creates new challenges that should be tackled by the green ICT community. Data scientists classify big data into four main categories (4Vs): Volume (with direct implications on power needs), Velocity (with impact on delay requirements), Variety (with varying CPU requirements and reduction ratios after processing) and Veracity (with cleansing and backup constraints). Each V poses many challenges that confront the energy efficiency of the underlying networks carrying big data traffic. In this work, we investigated the impact of the big data 4Vs on energy efficient bypass IP over WDM networks. The investigation is carried out by developing Mixed Integer Linear Programming (MILP) models that encapsulate the distinctive features of each V. In our analyses, the big data network is greened by progressively processing big data raw traffic at strategic locations, dubbed as processing nodes (PNs), built in the network along the path from big data sources to the data centres. At each PN, raw data is processed and lower rate useful information is extracted progressively, eventually reducing the network power consumption. For each V, we conducted an in-depth analysis and evaluated the network power saving that can be achieved by the energy efficient big data network compared to the classical approach. Along the volume dimension of big data, the work dealt with optimally handling and processing an enormous amount of big data *Chunk*s and extracting the corresponding knowledge carried by those *Chunk*s, transmitting knowledge instead of data, thus reducing the data volume and saving power. Variety means that there are different types of big data such as CPU intensive, memory intensive, Input/output (IO) intensive, CPU-Memory intensive, CPU/IO intensive, and

memory-IO intensive applications. Each type requires a different amount of processing, memory, storage, and networking resources. The processing of different varieties of big data was optimised with the goal of minimising power consumption. In the velocity dimension, we classified the processing velocity of big data into two modes: expedited-data processing mode and relaxed-data processing mode. Expedited-data demanded higher amount of computational resources to reduce the execution time compared to the relaxed-data. The big data processing and transmission were optimised given the velocity dimension to reduce power consumption. Veracity specifies trustworthiness, data protection, data backup, and data cleansing constraints. We considered the implementation of data cleansing and backup operations prior to big data processing so that big data is cleansed and readied for entering big data analytics stage. The analysis was carried out through dedicated scenarios considering the influence of each V's characteristic parameters. For the set of network parameters we considered, our results for network energy efficiency under the impact of volume, variety, velocity and veracity scenarios revealed that up to 52%, 47%, 60%, 58%, network power savings can be achieved by the energy efficient big data networks approach compared to the classical approach, respectively.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACET	Allocated CPU Execution Time
ACET	Allocated CPU execution time of a *Chunk*
AWP	Allocated Processing Workload
CBDN	Classical Big Data Networks
CCC	Couple Congestion Control
CDM	Code Division Multiplexing
CDN	Content Distribution Networks
CET	CPU Execution Time
CHT	Chunk Big Data Traffic
CMS	Compact Muon Solenoid
CPI	Cycles per Instruction
DC	Data Centre
EEBDN	Energy Efficient Big Data Networks
EON	Elastic Optical Networks
EXC	Electronic Cross-connect; Electronic Core Packet Switch
GHG	Green House Gases
IC	Instruction Count
ICT	Information and Communication Technologies
INF	Info Big Data Traffic
IO	Input/Output
IP	Internet Protocol
IPN	Intermediate Processing Node
ISP	Internet Service Provider

LHC	Large Hadron Collider
MILP	Mixed Integer Linear Programming
MPTCP	Multipath Transmission Control Protocol
MR	Malleable Reservation
NFV	Network Function Virtualization
NPC	Network Power Consumption
OCS	Optical Circuit Switching
OLS	Optical Label Switching
OPS	Optical Packet Switching
OSPF	Open Shortest Path First
OXC	Optical Cross Connect
PBS	Optical Burst Switching
PN	Processing Node
PPR	Processing Reduction Ratio
PUE	Power Usage Effectiveness
PW	Processing Workload
RCET	Requested CPU Execution Time
RCET	Requested CPU execution time of a *Chunk*
RSA	Routing and Spectrum Assignment
RWP	Requested processing workload for a *Chunk*
SDH	Synchronous Digital Hierarchy
SONET	Synchronous Optical Networks
SPN	Source Processing Node
TDM	Time Division Multiplexing

WDM	Wavelength Division Multiplexing
Φ	Required processing weight for a *Chunks*

Chapter 1 Introduction

In recent years, energy crises and environmental protection are under the spotlight, specifically, the remarkable growth of energy consumption in ICT (Information and Communication Technologies). According to [1], in the first quarter of 2007, British Telecom consumed about 0.7% of the total UK's power consumption and therefore was the largest single power consumer in the UK. In the same line, about 8% of the total electricity in the United States is being consumed by information and communication technologies [2], and an increase of more than 4% per year has been predicted.

So far, applications such as telework telecom applications, video conferencing, e-commerce, and their impact on human movements, have stamped ICT as an environment-friendly sector, however, there is a downside of ICT. Due to the ICT availability, everywhere and anywhere in our daily life, (both private and professional), the power that is needed to maintain and operate the network is being considered as an essential problem associated with bandwidth growth. Another aspect, energy consumption of computers and network equipment is becoming a significant part of the global energy consumption as a result of the network expansion [3]. For example, during the last decade, the Internet bandwidth has increased by approximately 50 to 100 times [4], and accordingly, the network power consumption has increased simultaneously. Thus, lots of attention is being focused on "energy-aware" ICT solutions [3]. One of the most significant current discussions in today's ICT sector is the increase in energy consumption due to the massive increase in the number of devices accessing the Internet – with around 40% of the world population having an Internet connection [5] – and the huge amount of generated data. Data centre power consumption is in the range of 100-130 GWh per year, as measured by the power usage effectiveness (PUE) index, and computer room air conditioning consumes up to half the total power consumed by the data centre [6]. The amount of data created between the period of the dawn civilisation and 2003 is estimated to be five Exabyte. Currently, the same amount of data is created every two days [7]. This massive increase in generating data and the huge data generated is referred to as big data.

Data scientists classify big data into four main categories (4Vs): volume (implies enormous volumes of data), variety (refers to the many sources and types of data) velocity (deals with the pace at which data flows from source), and veracity (refers to the biases, noise and abnormality in data). Each V carries many challenges that have implications on the power consumption of the underlying networks carrying the big data traffic.

The first challenge facing the Data Centres (DCs) is the enormous **Volume** of data fluxing to them. Voluminous growth in generating big data is causing drops in the percentage of processed data of organisations due to the lack of resources and poor analysis tools [8]. Thus, a large amount of the data that is to be processed is either neglected, deleted or delayed. Hence, there is unnecessary networking power consumption, extra wastage of storage and bandwidth because of transferring raw data, which leads to increasing the financial and environmental costs.

Variety means that there are different types of big data applications such as CPU intensive, memory intensive, Input/output (IO) intensive, CPU-Memory intensive, CPU/IO intensive, and memory-IO intensive applications. Each application requires different amounts of processing, memory, storage, and networking resources. These different types come from the diversity of big data sources, such as healthcare sensors, smart devices, social networks [8].

Velocity is data in motion, which is the speed at which data is fluxing in and processed in the DCs. High-speed processing of such immense data volumes as produced by plentiful data sources calls for new processing and communications methodologies in the big data era. The processing velocity of big data can be classified into two modes: expedited-data processing mode and relaxed-data processing mode. Expedited-data processing mode is used for the CPU hungry applications that need to be processed in real time [9], e.g. remote patient monitoring. Relaxed-data processing mode can tolerate some delay and can be processed in a batch processing mode after being stored inside DCs, such as digital image processing and automated transaction processing.

Veracity of big data is a more serious challenge to data scientists since they need to distinguish between the meaningful data and the dirty data [10]. Low-quality data causes the U.S. economy to waste $3.1 trillion each year [10]. Veracity specifies trustworthiness, data protection, data backup, and data cleansing constraints. **Data cleansing** [11] deals with detecting and removing dirty data due to overlaps, errors, duplications, and contradictory materials from big data to improve its quality. It provides easy access to accurate, consistent and consolidated data of different data forms.

Significant efforts have been dedicated to optimise the power consumption of conventional data centres and the power consumption and communication cost of big data networks including energy-efficient data centre designs [12, 13], energy-efficient inter- and intra-data centre network architectures [14, 15], designing energy-efficient cloud computing services and energy-efficient resource provisioning, and virtual network embedding for cloud systems [16, 17], minimisation of overall cost for Big Data placement, processing, and movement across geo-distributed data centres [18-20], and minimisation of the communication cost of big data queries and real-time big data processing on the heterogeneous systems [21, 22].

To understand the usefulness of our proposed Energy Efficient Big Data Networks (EEBDN) concept, consider an example of a Classical Big Data Network (CBDN), where all the big data *Chunk*s Traffic (CHT); which is the unprocessed big data traffic, generated by the source nodes is forwarded to the DCs to be processed there. On the other hand, in the EEBDN, a progressive processing technique is implemented to process the CHT at strategic locations, dubbed Processing Nodes (PNs), built into the network along the path from the data source to the destination. Our progressive processing can be classified into three main stages: edge processing stage, which is implemented in the Source PNs (SPNs), intermediate processing stage, implemented in the Intermediate PNs (IPNs), and central processing stage, implemented in the DCs. During the processing of big data, the extracted information from the CHT raw traffic, referred to as *Info* traffic (INF), is smaller in volume compared to the original big data traffic each time the data is processed, hence, reducing network power consumption.

Typically, the size of *Info* is very small compared to the size of *Chunk*s [23] in many big data applications such as remote patient monitoring, where info is used for example to capture only the abnormality in the heartbeat from a huge amount of measured heartbeat rate time series. Processing Nodes (PNs) are attached to the core nodes. Each PN is composed of internal switches and routers, limited storage, and a limited number of servers depending on the available building space and its structure is similar to the cloud structure in [16]. DCs, however, are assumed to have large enough processing and storage capabilities. PNs are capable of processing different amount of big data traffic depending on their processing and storage capacity.

For energy efficiency in big data networks, the work in this thesis presents detailed analyses of the process of big data applications inside the PNs by considering the 4 Vs of big data and exemplifies its implication in the EEBDN. The approach exploits energy-efficient processing of big data *Chunk*s along the path from the source to the destination, starting from the SPNs, moving through the IPNs, and finally reaching the DCs. This approach showed the impact of increasing the total resources utilisation (such as PNs' and DCs' servers, switches, and routers), which reduced the overall energy consumption. Furthermore, this approach showed the impact of processing the CHT, (which typically contained large data volume, along the way from source data nodes to the destination) on the network power consumption.

The extracted knowledge from the CHT (i.e. the INF) is typically smaller in size, and this led to a significant reduction in big data traffic each time the data is processed, hence, reducing the network power consumption. The benefits were maximised using a mixed integer linear programming (MILP) mathematical optimisation, and a heuristic was developed and used to verify the MILP optimisation. The goal of the optimisation was to ensure that power consumption is minimised by effectively considering the distinct features of each V as mentioned earlier.

1.1 Research Objectives

The following primary objectives were set for the work reported in this thesis:

1. To design a network architecture that supports energy-efficient big data processing by building Processing Nodes (PNs).
2. To develop a scheme for the transmission and processing of enormous **volumes** of big data, starting at processing nodes close to the data source, with significant processing resources provided at centralised data centres.
3. To capture the distinct features of big data's **variety** by evaluating the energy efficiency implications of processing various big data applications in networks where edge, intermediate, and finally central processing in networks is considered.
4. To utilise the proposed network architecture to handle the challenges of the **velocity** dimension of big data where data can have expedited or relaxed processing requirements.
5. To investigate the impact of the **veracity** dimension on energy efficient big data networks by performing the cleansing operation in the proposed network architecture before processing big data applications such that the data is readied for big data processing.

1.2 Original Contributions

The main contributions of this thesis are as follows:

1. We introduced MILP models to minimise the power consumption of networks that include processing nodes considering the impact of volume and variety of big data in IP over WDM core networks. As a result, we made the following contributions:

(i) we evaluated, using MILP and heuristic, the volume dimension by analysing the optimal distribution of big data volumes in different processing locations that have different processing capabilities where *Chunks* demand similar processing and yield similar volume reduction ratios; (ii) we examined, using MILP, the impact of variety by considering the case where different CPU workloads are required to serve different volumes of *Chunks* at different volume reduction ratios; (iii) we used our progressive processing technique to optimise the processing locations of the big data *Chunks* and compared the results to the classical technique where no PNs exist in the network. The processing locations are optimally chosen at either SPNs, inside "location optimised" DCs or at the IPNs. Thus, we jointly minimised the power consumption of the overall network and processing resources since the network elements, e.g., router ports, the routing paths, and the processing resources, have their energy efficiently utilised; (iv) we assessed the impact of the energy efficiency of PNs hardware on the proposed energy efficient big data networking approach where the PNs constituents (LAN switches, routers, and servers) consume higher power compared to data centre equipment; (v) we considered a software matching problem to evaluate the performance limits of our approach where each PN contains different software packages used to process different big data applications.

2. We developed a MILP model to jointly minimise the power consumption of the network and the power consumption of the processing of information given the velocity dimension of big data. We considered bypass IP over WDM core

networks with our progressive processing approach. As a result, we made the following contributions: (i) we explored the impact of the big data processing velocity dimension on the IP over WDM network energy efficiency considering an expedited-data processing mode and a relaxed-data processing mode, (ii) we used our progressive processing technique to process big data *Chunks* and compared the results to the classical approach where progressive processing is not allowed. In our approach, the processing locations are optimally selected at the SPNs, at the IPNs or inside the centralised data centres (DCs). As a result, a significant reduction in the network power consumption is achieved each time the data is processed along the journey from the source to the DCs.

3. We developed a MILP model to jointly minimise the power consumption of the network and the power consumption of the processing of information given the veracity dimension of big data. Here we considered the power consumption of bypass IP over WDM core networks and the power consumption of processing nodes. As a result, we made the following contributions: (i) we studied the influence of *Veracity* by performing cleansing and backup for big data *Chunks* before processing, where a Backup Node (BN) location is optimally selected to store a copy of the cleansed big data *Chunks*; (ii) we employed our energy efficient edge, intermediate, and centralised processing technique to process the cleansed *Chunks* in optimal locations in the core network and compared the results to the classical approach that lacks progressive processing. The optimally selected processing locations for the energy efficient approach are either the SPNs where

Chunks are generated, inside location optimised DCs, or at the IPNs, between the SPNs and the DCs. Accordingly, the network elements, e.g., router ports, the routing paths, and the processing resources, are efficiently utilised to jointly minimise the power consumption of the overall network and processing resources.

1.3 Related Publications

The original contributions in this thesis are supported by the following publications:

- **Journals**

1. Al-Salim, A., EL-Gorashi, T., Lawey, A. and Elmirghani, J.M., “Greening Big Data Networks: Velocity Impact”. IET Optoelectronics. 2017 Nov 21.
2. Al-Salim AM, Lawey AQ, El-Gorashi TE, Elmirghani JM. “Energy efficient big data networks: impact of volume and variety”. IEEE Transactions on Network and Service Management (TNSM), pp. 458-474, 2017 Dec 28.
3. Al-Salim, A., El-Gorashi, T., Lawey, A.Q., and Elmirghani, J.M., “Greening Big Data Networks: Veracity Impact”, Springer Journal of Photonic Communications, Accepted, March 2018.

- **Conferences**

4. A. M. Al-Salim, A. Q. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy Efficient Tapered Data Networks for Big Data Processing in IP/WDM Networks," in Transparent Optical Networks (ICTON), 2015 17th International Conference on, 2015, pp. 1-5.

5. A. M. Al-Salim, H. M. Mohammad Ali, A. Q. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Greening Big Data Networks: Volume Impact," in Transparent Optical Networks (ICTON), 2016 17th International Conference on, 2016, pp. 1-6.

1.4 Thesis Structure

Following the introduction in Chapter 1, the rest of the thesis is organised as follows:

Chapter 2 reviews optical networks, energy efficient optical networks, mixed integer linear programming (MILP), IP over WDM networks, and provides an overview of the main topics related to the work on energy efficiency in optical networks.

Chapter 3 discusses the main sources of big data, big data characteristics, Hadoop-MapReduce scheme, and reviews the work on big data processing and networking. Attention is given to the related work on minimising communication cost and energy efficiency in big data networks. A special section is dedicated to introducing the proposed topics on energy efficient big data networks and comparison with the classical networks with an illustrated example.

Chapter 4 introduces a MILP model and a heuristic to examine the impact of big data's volume on energy efficient big data networks. It also introduces a MILP model to examine the impact of the big data's variety dimension on energy efficient big data networks.

Chapter 5 introduces a MILP model that focuses on power minimisation in big data networks given the big data's velocity requirements.

Chapter 7 introduces a MILP model to minimise big data networks and processing power consumption given the veracity dimension of big data.

The thesis concludes in Chapter 7, which summarises this work's main contributions and gives recommendations for future work.

Chapter 2 Optical Networks

The globalisation of the Internet and the boom in high bandwidth applications such as video streaming and Big Data applications has greatly accelerated the call to develop the communications infrastructure. Accordingly, such huge bandwidth requirements have gradually resulted in developing the bandwidth supported by the optical fibre to reach its current maximum of 1.050 petabit/s over 52.4 km of 12-core (light paths) optical fibre [24]. Nowadays, the optical fibre is the dominant medium which can handle such immensely growing traffic. This is because of its many advantages compared to copper and wireless mediums such as high reliability and low signal attenuation.

2.1 The Evolution of Optical Networks

In 1965 fibre optics has come to the light and this has been considered as the start of networks revolution [25]. Since then to present, the development in optical networks has undergone two main phases: Phase one is related to the transmission where the optical fibre is used only to transmit huge traffic in peer to peer fashion while the routing and switching are electronic devices responsibility. That was the era of Synchronous Optical Networks (SONET) and Synchronous Digital Hierarchy (SDH) [26].

Phase two has witnessed the revolution with the introduction of multiplexing techniques such as Time Division Multiplexing (TDM), Frequency Division Multiplexing (FDM), Code Division Multiplexing (CDM), and Wavelength Division Multiplexing (WDM) [26]. TDM specifies time slots for users to transmit data over a single communication channel with fixed bit rate as shown in Figure 2-1 [27]. This technique helps to utilise all the available bandwidth but users' data may be delayed depending on the time slot window.

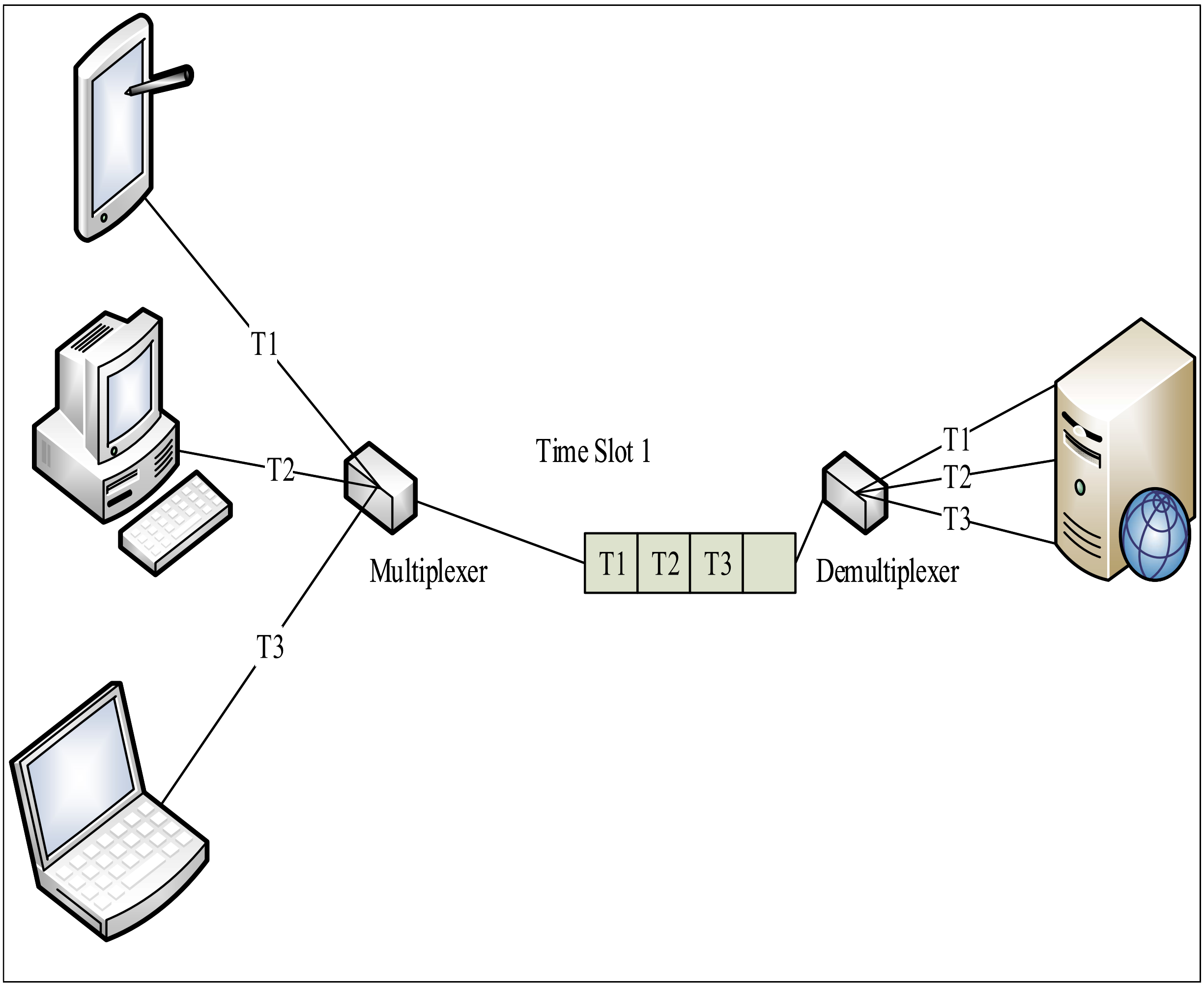


Figure 2-1: Time division multiplexing (TDM).

FDM [28] divides the total bandwidth of the system into different sub-channels without overlapping and this technique is useful when transmitting different requests at the same time but at a lower bit rate per request as shown in Figure 2-2.

WDM [26, 29, 30] is a technique that enables multiple data stream to be sent on multiple wavelengths as shown in Figure 2-3. In WDM, the spectrum of the transmitted signals is divided into a number of non-overlapping wavelengths (or frequency) slots, with each wavelength serving a specific communication channel operating at maximum electronic speed, and this allows the huge bandwidth of optical fibre to be utilised [30].

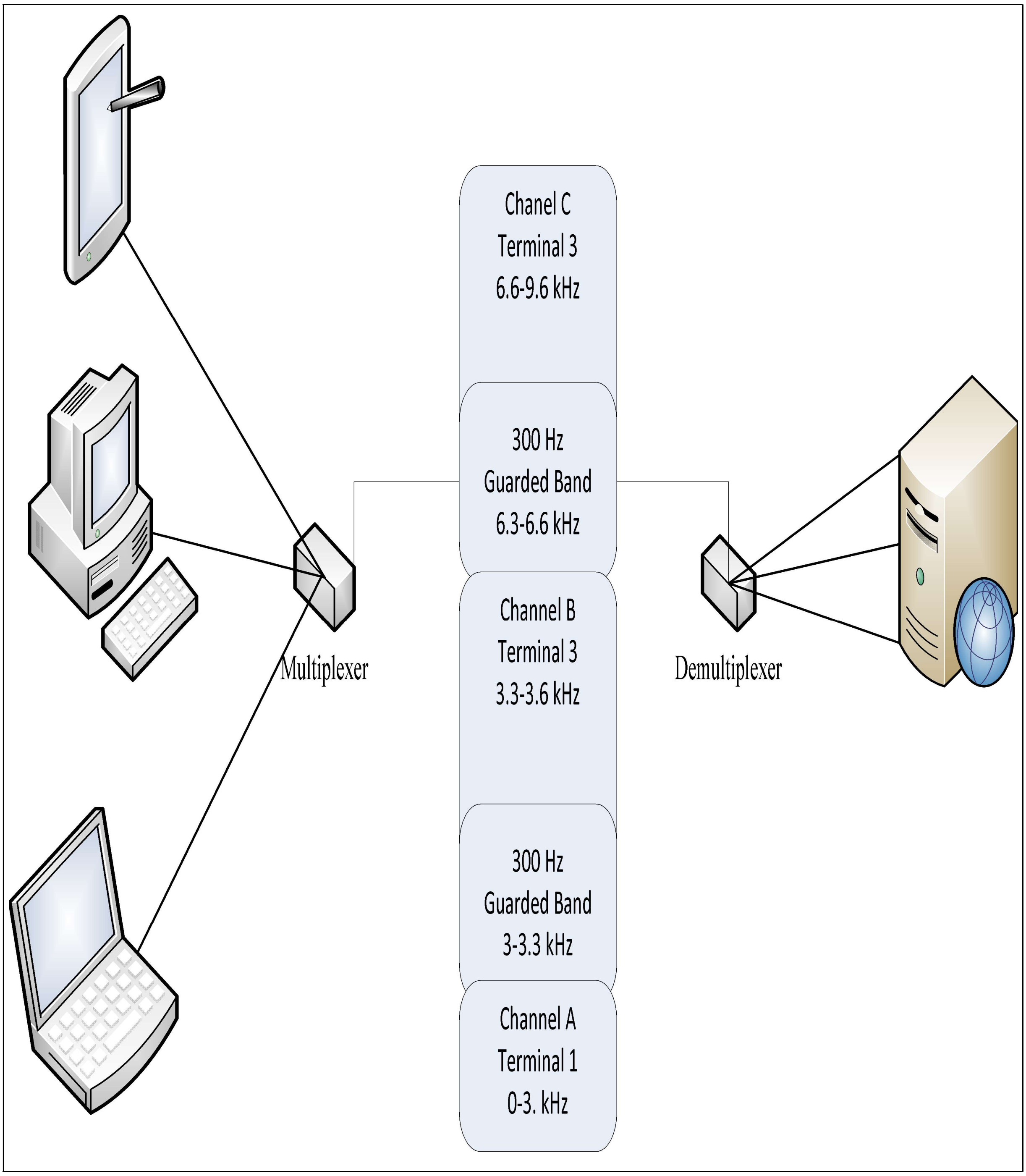


Figure 2-2: Frequency division multiplexing (FDM).

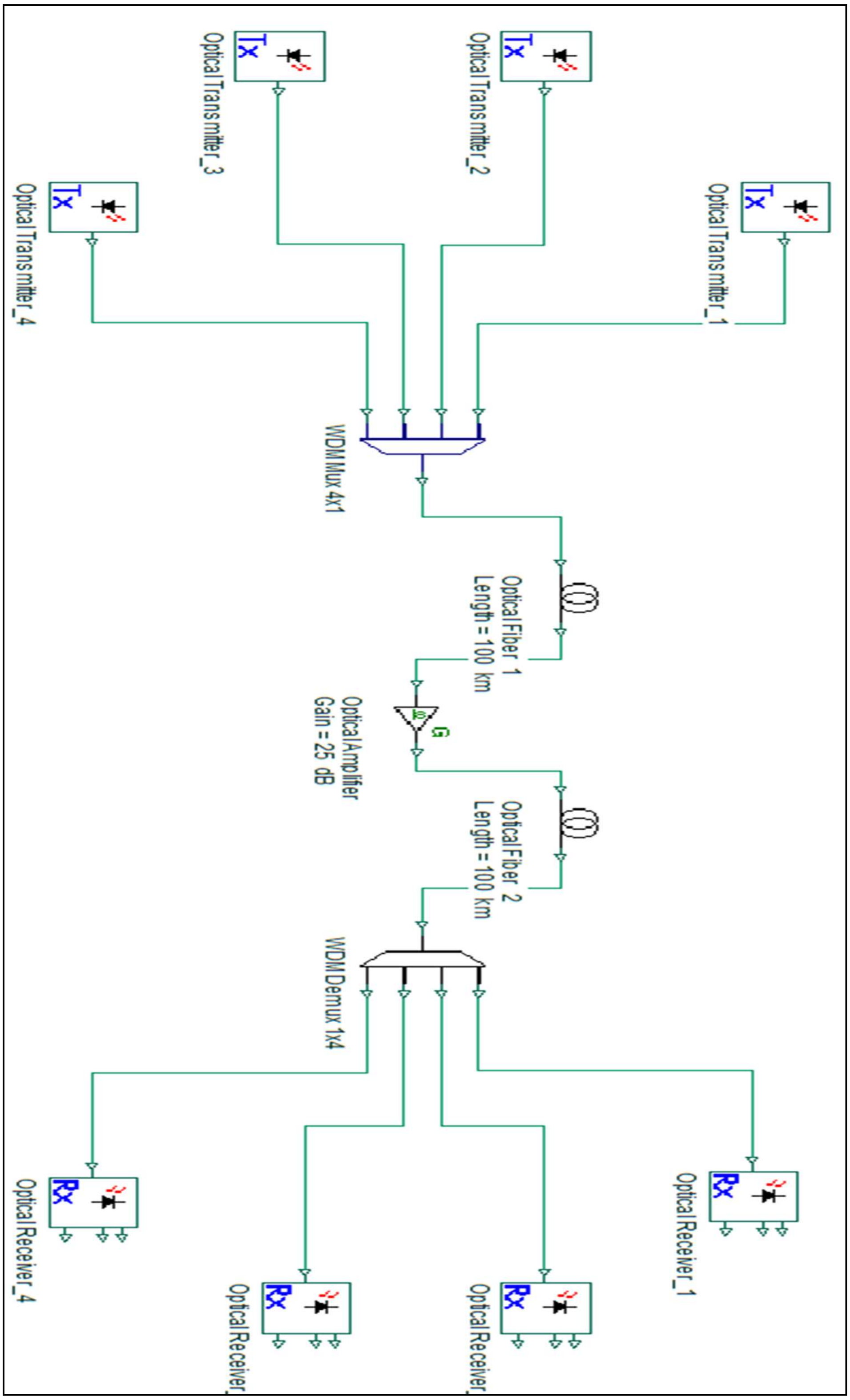


Figure 2-3: Wavelength division multiplexing (WDM).

2.2 WDM Switching Technology

Basically, there are four types of switching technology in WDM networks which are Optical Circuit Switching (OCS), Optical Packet Switching (OPS), Optical Burst Switching (OBS), and Optical Label Switching (OLS).

2.2.1 Optical circuit switching (OCS)

OCS can be defined as a source-destination permanent connection. Each connection requires dedicated path or a guaranteed amount of bandwidth and a wavelength all time [31, 32].

The sum of the bandwidth of all connections should be less or equal to the total link bandwidth. Before a source starts sending data to a destination, a Routing and Wavelength Assignment [26] algorithm, [31] assigns specific wavelengths and route to the lightpath.

The pros of OCS lie in assigning dedicated wavelengths during the transmission process, which is useful in terms of reliability and security for real-time applications. While the cons of this technique lie in utilising a portion of the total bandwidth which means that the remaining bandwidth is useless until the circuit is released. Moreover, there is delay in establishing the connection before the transmission process, which might be inappropriate for delay-sensitive applications.

2.2.2 Optical packet switching (OPS)

OPS is a technique in which different data segments from different users can share the same wavelength and there is no wavelength reservation and no wasted connection capacity. OPS can be divided into two types: synchronous and asynchronous. The size of packets in the synchronous approach are fixed and each packet should contain synchronisation bits. While in the asynchronous technique the size of packets is flexible with no synchronisation clock required between sender and receiver [26, 33, 34]. The first approach is used for delay sensitive applications while the second one is not.

2.2.3 Optical burst switching (OBS)

Traffic in the optical network is greatly increasing and this requires very fast switching in the order of few nanoseconds [33, 35, 36]. OBS combines the merits of both OCS and OPS and may be efficient as it first does not need packet switching buffering but it may also not utilise the whole wavelength band resulting in wastage. OBS includes two packets, control packet, and data burst packet. The control packet is used to establish the path for the data burst. This is done by sending this packet along the route to the destination node. It is processed electronically at the core network nodes to decide a specific route for each data burst packet. The ingress OBS node then gathers these data burst packets, which have different size and sends them to the destination as shown in Figure 2-4.

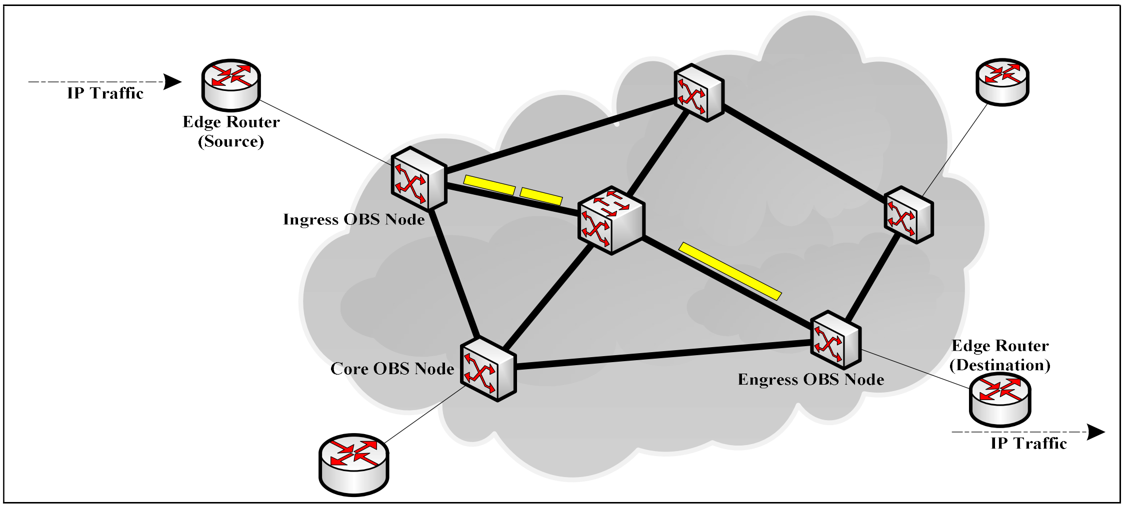


Figure 2-4: Optical burst switching network.

2.2.4 Optical label switching (OLS) [16]

OLS combines the benefits of optical burst switching OBS and optical packet switching OPS and overcomes their disadvantages [37-39]. It can hold-up both packets and bursts by assigning different labels for each of them. Before sending data, OLS ensures a lightpath between source and destination is established. This is done by sending a signal to the control layer to establish a Label Switching Path (LSP). Generally, all packets carrying the same label should be transmitted through the same LSP. When a packet passes through intermediate switching nodes and reaches the edge router, it is assigned a new label by this router according to the labels stored in the router using the forwarding labels table. The packets with new labels are forwarded to their destinations according to the label forwarding table.

2.3 IP over WDM

The IP over WDM network is comprised of two layers the IP layer and the optical layer as shown in Figure 2-5. The IP layer consists of a core IP router connected to an optical switch. The IP router aggregates data traffic from low-end access routers. The optical layer provides the needed huge capacity and bandwidth for the communications between IP routers [40]. Optical switches are connected to physical fibre links, and each link may contain multiple fibre strands. Each fibre is supplemented by: pair of wavelength multiplexer/demultiplexers required to multiplex/demultiplex wavelengths; transponders that can provide optical/electrical/optical (OEO) conversion and hence also full wavelength conversion at each core node; and, for long distance transmission, Erbium Doped Fiber Amplifiers (EDFAs) are used to amplify the optical signal. An automatically controllable optical cross-connect (OXC) switch is used as the core optical switch box or a dumb optical patch panel can be used instead [25, 41].

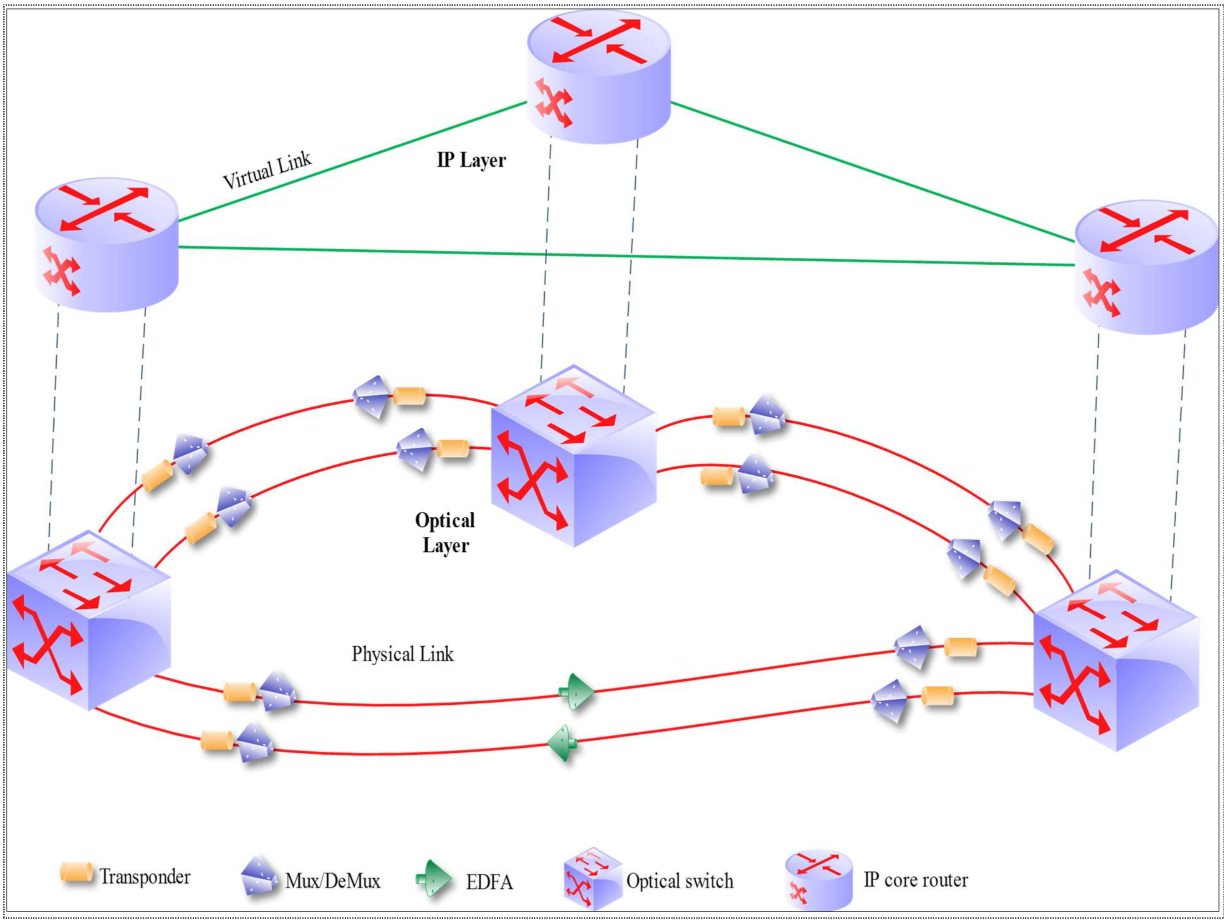


Figure 2-5: IP over WDM network.

IP over WDM networks can be realised in two ways: Lightpath non-bypass and lightpath bypass. Considering lightpath non-bypass, the lightpath must be dropped at each intermediate node and all the data carried by a lightpath must be processed and forwarded by all IP routers on its path to the destination node. In contrast, the lightpath bypass approach uses a cut-through lightpath, where a lightpath directly bypasses intermediate nodes. Only destination node IP router processes the lightpath data. This requires the optical nodes to have intelligence to bypass lightpaths destined to other nodes. The implementation of lightpath bypass however has the advantage of introducing a significant reduction in the number of working IP router ports [42, 43]. The communications between core routers are directly over lightpaths, where each lightpath joins a pair of router ports, and lightpaths are considered virtual links for the IP layer. IP routers play a major role in the total energy consumption in an IP over WDM transport network. Thus, minimising the required number of IP router ports can potentially maximise the energy consumption saving in an IP over WDM network [42, 43].

2.4 Energy-Efficiency in Optical Networks

2.4.1 Introduction

Today, most of the energy need is being provided by traditional energy sources, such as hydrocarbon energy sources. According to [3], about 85 percent of the primary energy of USA's electricity is provided using this source, however, this energy is not renewable, and its use is expected to be finally minimised. Also, large amounts of Green House Gases (GHG), the main reason for Global Warming, are emitted because of the combustion of hydrocarbon materials. Thus, hydrocarbon energy sources utilisation should be minimised. If traditional energy sources such as coal or natural gas are used, a network component that consumes 1 kWh of such traditional electrical energy emits approximately 228 grams of CO_2 to nature [4]. The latest report of SMARTer2030 estimates that the ICT emissions could be reaching 1.25Gt CO2e in 2030 or 1.97% of global emissions [44]

In order to address this important problem, mutual responsibility requires both network operators and system vendors, alike, to corporate in order to reduce the carbon footprint and reduce the environmental impact of communication networks [45]. In [4], the use of renewable energy has been proposed to reduce the CO_2 emissions at a given energy consumption level. A Linear Programming (LP) model was also developed for energy minimisation in the network when renewable energy is used and a novel heuristic was proposed for improving renewable energy utilisation. While the routing in the electronic layer consumes a large amount of power, routing in the optical layer coupled with renewable energy nodes significantly reduces the CO_2 emission of the IP over WDM network considered by 47% to 52%, and the new heuristic introduced hardly affects the QoS. Secondly, in many science and technology areas, energy-aware ICT solutions are being proposed [43], low-energy equipment and components are being developed, not only to decrease the energy cost but also to help save our environment [3].

For example, if 1kW non-renewable power consumption can be eliminated by changing the network design, then a significant reduction, about 2 tons, in CO_2 emission may be achieved every year. As a family vehicle typically emits 150 g/km of CO_2; therefore, in a year a 1 kW router port emits CO_2 amount approximately equivalent to 13 k journeys in a family vehicle. It was shown that ICT is one of the most promising areas for exploring energy conservation.

2.4.2 Power consumption problem and energy efficiency solutions

So far, the main research focus in ICT was to achieve higher data rates, without much consideration of energy efficiency. However, one of the main drawbacks of many of these new techniques is that these approaches significantly increase system complexity and energy consumption [46].

Although, the power consumption of ICT networks can be reduced on one of three levels, namely circuit, equipment and network level; a combination of approaches can be used. For example, energy efficiency approaches can be implemented at the equipment level, i.e. energy-efficient components, as well as at the network level, ie energy-efficient routing and traffic grooming, in the network at the same time [47]. To reduce the networks power consumption, various methods and approaches have been proposed and investigated comprehensively. At the circuit level, for example; components are being designed such that their core processors and electronic modules work and are managed at very low operational power. On the transmission side, the deployment of Long-reach WDM transceivers and low attenuation low-dispersion fibers increases the transmission efficiency effectively, whereas at the system level, sleep-wake cycles can effectively be used in power saving in various network equipment [48]. In the following sections, the energy minimisation in the different network levels is being discussed.

2.5 Energy Minimisation in Core Networks

Energy consumption in core networks is primary due to data transmitters and switching equipment such as transponders, OXC (Optical Cross Connects), EDFAs (Erbium Doped Fibre Amplifiers), and routers. The energy consumed in core networks is large [3], and the percentage of the total network power consumption that the core network is responsible for is expected to increase significantly with the growing demand for data-intensive applications from the Internet. Therefore, power consumption in backbone networks has received increased attention. In addition, heat dissipation has attracted increased attraction.

Due to the fact that power consumption of the backbone network is often limited to a few locations, minimising the power consumption of the IP over wavelength-division-multiplexing (WDM) backbone network is essential [43]. For present and future Internet, WDM networks will continue to be employed as they are able to provide a huge amount of bandwidth. The formation of the backbone networks relies on the use of these networks in a large scale [2]. The invention of OXC nodes, which can switch the wavelengths completely in the optical domain, has enabled new dynamic optical capabilities in WDM extending the use and utility of WDM into the future. A future promising technique for managing the increased power consumption in the core network is Optical bypass [49]. Processes like traffic engineering or power consumption minimisation in optical-to-electrical-to-optical conversion by optically bypassing the energy exhausting nodes in the network can be deployed to reduce the network power consumption. In [48] a dynamic routing protocol has been proposed which minimises the power consumption in the core optical WDM network.

As WDM optical networks have the ability to route each optical circuit, i.e. lightpath, on a dedicated wavelength passing a series of optical fibre links from source to destination without the need for intermediate data processing, the deployment of optical technologies based on WDM continues to be one of the promising techniques to reduce core networks energy consumption. The use of WDM based optical core networks significantly reduces the need for optical/electronic/optical (OEO) conversions, and hence the extra power consumption in the optoelectronic devices.

Consequently, in the past few years, the energy efficiency of the transport layer of IP/WDM networks has received increased research attention [50].

Mixed integer linear programming (MILP), models have been built to study the optimisation of core networks to minimise the embodied energy and the operational energy of networks. Two approaches have been investigated for energy efficiency in core content distribution networks: data compression in optical networks and locality in P2P networks [51]. The physical topology of IP over WDM networks has been optimised considering the embodied energy of the network devices in addition to the operational energy and it has been compared to optimising the physical topology considering operational energy only. In addition, investigations have been carried out for the power consumption savings achieved by optimising the data compression ratio of traffic demands in IP/WDM networks. Moreover, the energy consumption of BitTorrent, the most popular P2P content distribution protocol, has been compared to client/server (C/S) systems over IP over WDM networks.

Generally, in the past, energy efficiency has received little attention by network architects and operators. With the increase in energy prices and environmental concerns, energy-efficiency has become a significant design metric in recent research efforts. The current research approaches to reduce core networks energy consumption can be summarised in four categories: (1) selectively turning off network elements, (2) energy-efficient network design, (3) energy-efficient IP packet forwarding, and (4) green routing [3].

2.5.1 Selectively turning off network elements

This approach aims to save energy in the core network by selectively switching off idle network components during low load periods as in [52], for example during the night, while adjusting the network vital and essential functions which enable it to serve the remaining traffic.

As mentioned in [3], node turn off can be executed in the following situations (i) at the node idle time, i.e. when it is totally unused, (ii) when the traffic passing the node goes below a given threshold, the remaining traffic routing responsibility is left to upper layers, and (iii) after proactively rerouting the traffic along other routes, in order to avoid traffic loss and disruptions.

The above-mentioned approaches add extra burdens in addition to control, management, and operation of the network. Regarding the first approach, it requires no or minimal additional network control and the second only requires gathering congestion information, while the third approach can be applied only in a network that has some form of automatic provisioning and/or adaptive provisioning in place.

On one hand, links can be switched off in the same manner, i.e. when there is no traffic passing through them, or when traffic goes below a specific level, or if it is possible to perform traffic rerouting. A drawback to this approach is that most of the core network components, unfortunately, cannot be shut down without affecting the overall network performance. So, to perform this approach it is important to evaluate it carefully under QoS and connectivity constraints, as shutting off a core node requires rerouting of the connection and this may cause congestion or the traffic may be routed over a longer route which is unacceptable for different reasons as it may cause extra delay.

2.5.2 Energy-efficient network design

The second proposed solution to achieve energy efficiency, [49], is the possibility of devising energy efficient network components and architectures. For instance, an IP/WDM network design, where the IP routers, EDFAs, and transponders energy consumption is minimised jointly, is found to have a significant impact on network energy efficiency. In [53], heuristics have been proposed to minimise the energy consumption of the network. The IP/WDM implementation has been considered in two ways, non-bypass and bypass. As mentioned above, the results in the literature show that lightpath bypass achieves higher energy saving compared to non-bypass, relying on the fact that in lightpath-bypass the number of IP routers can be decreased. The power consumption of routers, EDFAs, and transponders has been measured and reported in [54]. It was showed that the total power consumption of routers is much more than that of EDFAs and transponders in IP/WDM networks.

Furthermore, in order to reduce the IP layer energy consumption, energy-aware packet switching has been proposed in [48] and [55]. It was showed that the main parameter that affects the power consumption in the router is the IP packet size. In general, there is an inverse relationship between packet size and router power consumption, i.e. the smaller the IP packet the higher the energy consumed by the router. This is attributed to the small ratio of payload to header size in small packets. Therefore, small packets call for frequent routing table look up, path computation and processing in general in routers [49]. As a result, by optimising the data packet size a new energy efficient routing paradigm for IP packets can be achieved. On the other hand, there exists a relationship between packet switching delay and energy-efficient IP packet forwarding. So, another energy efficient IP packet switching mechanism is being evaluated, namely pipeline forwarding. This approach is a time-based IP packet switching scheme. In this scheme, energy efficient IP packet switching is carried out along all the path to the network edges [56].

2.5.3 Green routing

The final solution considered here is the energy-aware routing (green routing) scheme which has been proposed in [57] as a novel routing scheme in core networks. In this scheme, the network energy consumption is considered as the design optimisation objective. Several other energy-aware routing schemes have been proposed, for example, an energy-aware routing scheme, which considers line card/chassis reconfiguration in IP routers. Comparing this scheme to the traditional shortest path or non-energy-aware routing scheme, significant energy saving is obtained [58]. Due to the fact that line cards and chassis are essential energy consuming components in the core network and in traditional routing schemes, and the fact that they are not configured to utilise energy efficiently, the energy saving of this energy-aware scheme was remarkable.

In addition, in the future energy efficient routing needs to be dynamic such that the traffic rerouting and the energy saving are done according to traffic variation. However, recent research has raised a concern that with current ever increased demand for the Internet and ICT services and products, the increase in energy efficiency is not able to counter the current huge growth in the deployment of new services and applications.

2.6 Linear Programming

The recent development of linear programming is attributed to World War II when a system that can maximise the efficiency of available resource was highly required and was of utmost importance [59]. Linear programming has been considered as one of the most important scientific achievements in the mid-20th century as it has had a very remarkable impact on all society sectors since 1950. Nowadays linear programming is saving thousands or millions of dollars for most companies and businesses.

2.7 Linear Programming Capabilities

The most common type of problems that linear programming can solve is the general problem that involves the allocation of limited resources in the best (or optimal) way to achieve the goal of gaining maximum profit or minimum cost. The range of activities that this definition can be applied to is diverse indeed, ranging from allocating production facilities to products to the national resource allocation to domestic needs, from portfolio selection to shipping patterns selection, from agricultural planning to computerised and networked systems designing, and so on [60].

Linear programming uses mathematical modeling to characterise the problem under consideration. As the name implies, all the mathematical functions in the model have to be linear functions and the word programming was a military term that is a synonym to the word 'planning' of schedules efficiently or deploying men optimally and does not refer to computer programming. Thus, linear programming refers to the planning of activities to gain optimal results that achieve the specified goal among all feasible alternative results. In addition to the most common application of allocating resources, linear programming has numerous important applications. Any programme whose mathematical model fits the very general linear programming format is a linear program as well [59].

2.8 A Linear Program General Form

A linear program consists of two parts. The first part is the expression being optimised, which is called the objective function. This function must be optimised under the restriction of a given set of constraints. The second part is the variables $x_1, x_2 \ldots x_n$, which are called decision variables, and their values are subject to $m + 1$ constraints, every equation ending with a b_i, in the example below, plus the non-negative equation. A point consists of a set of $x_1, x_2 \ldots x_n$ satisfying all the constraints is called a feasible point and the set of all such points is called the feasible region. Thus, the solution of a linear program must be a point in this feasible region and any other solution outside this region does not satisfy all the constraints [61]. An example of a linear programme is given below:

Objective Function:

$$
\begin{aligned}
&\text{Minimise} && c_1x_1 + c_2x_2 + \ldots + c_nx_n = z \\
&\text{Subject to:} && a_{11}\,x_1 + a_{12}\,x_2 + \ldots + a_{1n}\,x_n = b_1 \\
& && a_{21}\,x_1 + a_{22}\,x_2 + \ldots + a_{2n}\,x_n = b_2 \\
& && \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad \vdots \\
& && a_{m1}\,x_1 + a_{m2}\,x_2 + \ldots + a_{mn}\,x_n = b_m \\
& && x_1 \;,\; x_2\,,\; \ldots\,,\; x_n \geq 0
\end{aligned}
$$

2.9 Simplex Optimisation

In 1947, the Simplex method of optimisation was developed by George Dantzig, a member of the U.S. Air Force, in order to provide an efficient algorithm for solving linear programming problems of enormous size [59].

The simplex method is a mathematical procedure that uses iterations to improve the solution at each step and when there is no possibility of further improvement, the procedure stops. The procedure starts with a random vertex of the objective function and repeatedly tries to find a new vertex value which improves the solution compared to its predecessor. The search continues till the best result, which cannot be improved any more, is obtained. The solution of a linear program is accomplished in two steps. In the first step, known as Phase I, a starting extreme point is found. Depending on the nature of the program this may be trivial, but in general it can be solved by applying the simplex algorithm to a modified version of the original program. The possible results of Phase I are either that a basic feasible solution is found or that the feasible region is empty. In the latter case the linear program is called infeasible. In the second step, Phase II, the simplex algorithm is applied using the basic feasible solution found in Phase I as a starting point. The possible results from Phase II are either an optimum basic feasible solution or an infinite edge on which the objective function is unbounded below [62]. The Simplex method is based on the following property: If the objective function, z, does not take the max value in the A vertex, then there is an edge starting at A, along which the value of the function grows [63].

2.10 Adapting Model to the Simplex Method [63]

The following points must be taken into consideration to set a linear programming model in the standard form:

- The objective must be maximising or minimising the function.
- All restrictions must be equal.
- All variables are not negative.
- The independent terms are not negative.

The non-negativity condition can be achieved by adding slack variables, for example:

Given the linear program:

$$\begin{array}{ll} \text{Maximise:} & -x_1 + 3x_2 - 3x_3 \\ \text{Subject to:} & 3x_1 - x_2 - 2x_3 \leq 7 \\ & -2x_1 - 4x_2 + 4x_3 \leq 3 \\ & x_1 - 2x_3 \leq 4 \\ & -2x_1 + 2x_2 + x_3 \leq 8 \end{array}$$

$$3\,x_1 \qquad\qquad \leq 5$$

$$x_1, x_2, x_3 \geq 0$$

By adding the Slack variables, we get:

Maximise: $\delta \;=\; -\,x_1 + 3\,x_2 - 3\,x_3$

Subject to: $w_1 = 7 - 3\,x_1 - x_2 - 2\,x_3$

$$w_2 = 3 + 2\,x_1 + 4x_2 - 4\,x_3$$

$$w_3 = 4 - x_1 \qquad + 2\,x_3$$

$$w_4 = 8 - 2\,x_1 - 2\,x_2 - x_3$$

$$w_5 = 5 - 3\,x_1$$

$$x_1, x_2, x_{3,}\, w_1,\, w_2,\, w_3,\, w_4,\, w_5 \geq 0$$

2.11 Network Design Problems with Linear Programming

Linear programming in computer networks is very popular and is considered a very efficient design tool. Below are illustrative simple numerical examples that represent some network design problems [64]. Network design problems can be formally formulated using mathematical notations. A good mathematical notation can represent a specific design problem in a compact and clear way and it simplifies the understanding of the given design problem. Furthermore, network design problems can be presented in two ways: link-path formulation and node-link formulation.

2.11.1 Link-path formulation

Let us consider a simple network consisting of three nodes where each node is connected to the other two nodes, i.e., the network topology looks like a triangle, see Figure 2-6. Given the node, link, path, demand, demand path-flow variables, Constraints, and the objective function are defined in [64].

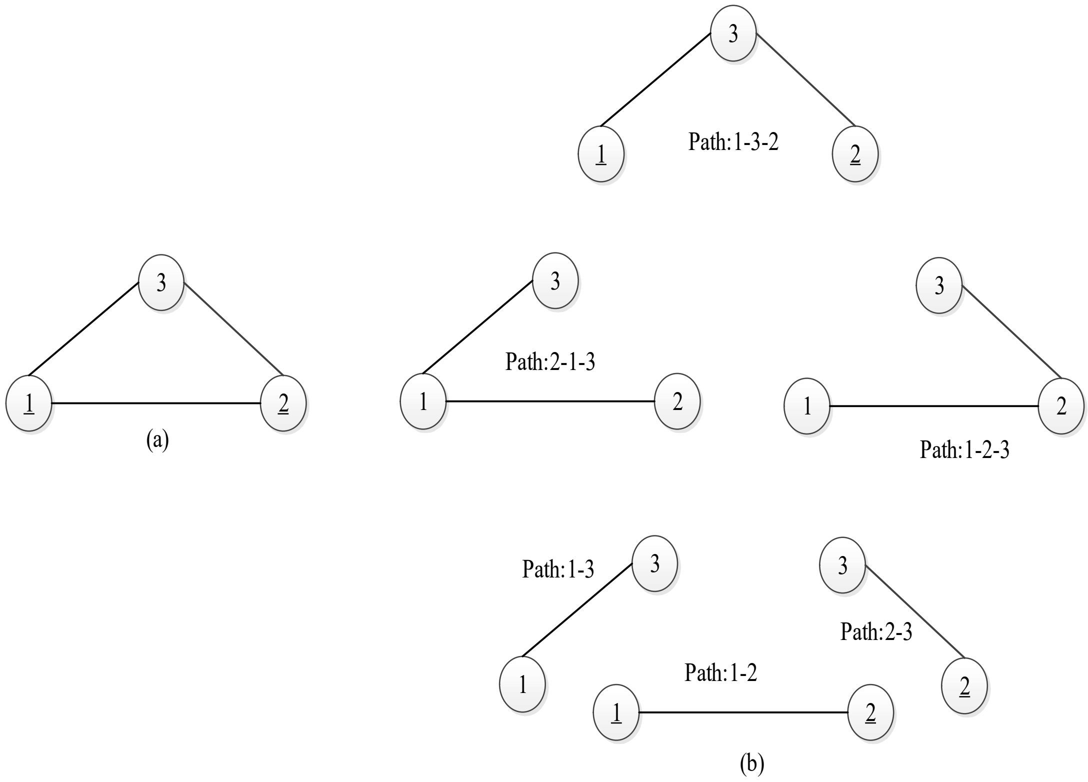


Figure 2-6: (a) A three node network example (b) All possible paths for the three-node example.

2.11.1.1 Network flow example in link-path formulation

Example description:

Suppose that the demand volume between nodes 1 and 2 is 5, between nodes 1 and 3 is 7, and between nodes 2 and 3 is 8 (units), and the demand is assumed to be bi-directional.

$$\hat{h}_{12} = 5, \quad \hat{h}_{13} = 7, \quad \hat{h}_{23} = 8$$

The demand volume for the given network for a pair of nodes can be routed over two paths. For instance, the demand pair with end nodes 1 and 2, its demand volume is routed over the direct-link route 1-2 and the alternate route 1-3-2 via node 3 as shown in Figure 2-6-b. So, if we use $\hat{x}$ with an appropriate subscript identifier to denote the unknown demand path-flow variables, then for demand pair (1,2), we can write:

$$\hat{x}_{12} + \hat{x}_{132} = 5\ (= \hat{h}_{12})$$

In any communication system, the total link load must not exceed the total link capacity. Thus, we have the following inequality for link 1-2:

$$\hat{x}_{12} + \hat{x}_{123} + \hat{x}_{213} \leq \hat{C}_{12}$$

In this example, we assume that the capacity of the first two links is 10 and the third is 15 (units);

$$\text{thus: } \hat{C}_{12} = \hat{C}_{13} = 10, \quad \hat{C}_{23} = 15$$

Suppose the goal of this example is to minimise the total routing cost. Assuming that the cost of routing one unit of flow on every link along its path is simply set to 1, the total routing cost for all the flow variables is:

$$\boldsymbol{F} = \hat{x}_{12} + 2\hat{x}_{132} + \hat{x}_{13} + 2\hat{x}_{123} + \hat{x}_{23} + 2\hat{x}_{213}$$

Put all together:

Minimise: $\boldsymbol{F} = \hat{x}_{12} + 2\hat{x}_{132} + \hat{x}_{13} + 2\hat{x}_{123} + \hat{x}_{23} + 2\hat{x}_{213}$

Subject to:

$$\begin{array}{ccccccccc}
\hat{x}_{12} & + \hat{x}_{132} & & & & & = & 5 \\
& & \hat{x}_{13} & + \hat{x}_{123} & & & = & 7 \\
& & & & + \hat{x}_{23} & + \hat{x}_{213} & = & 8 \\
\hat{x}_{12} & & & + \hat{x}_{123} & & + \hat{x}_{213} & \leq & 10 \\
& \hat{x}_{132} & + \hat{x}_{13} & & & + \hat{x}_{213} & \leq & 10 \\
& \hat{x}_{132} & & + \hat{x}_{123} & + \hat{x}_{23} & & \leq & 15
\end{array}$$

$$\hat{x}_{12},\ \hat{x}_{132},\ \hat{x}_{13},\ \hat{x}_{123},\ \hat{x}_{23},\ \hat{x}_{213} \geq 0$$

Optimal solution/optimal cost is:

$$\hat{x}_{12}{}^* = 5, \qquad \hat{x}_{13}{}^* = 7, \qquad \hat{x}_{23}{}^* = 8, \qquad F^* = 20.$$

2.11.2 Node-link formulation

The notions of link and path was used in the mathematical formulation presented in the previous network optimisation problem example. However, there is still another way to represent the same problem. Consider directed demands on directed links and fixed demand pairs and fixed nodes. Here we considered the total link flow for a specific demand on each link, which is zero for most links. Now from the point of view of a fixed node which is not the destination or the end node of the considered demand, called transit or an intermediate node, the flows come into this node on the incoming links and go out on its outgoing links un-altered, provided the node is not the source node or the destination node of the demand. This is called flow conservation law which is described in Figure 2-7.

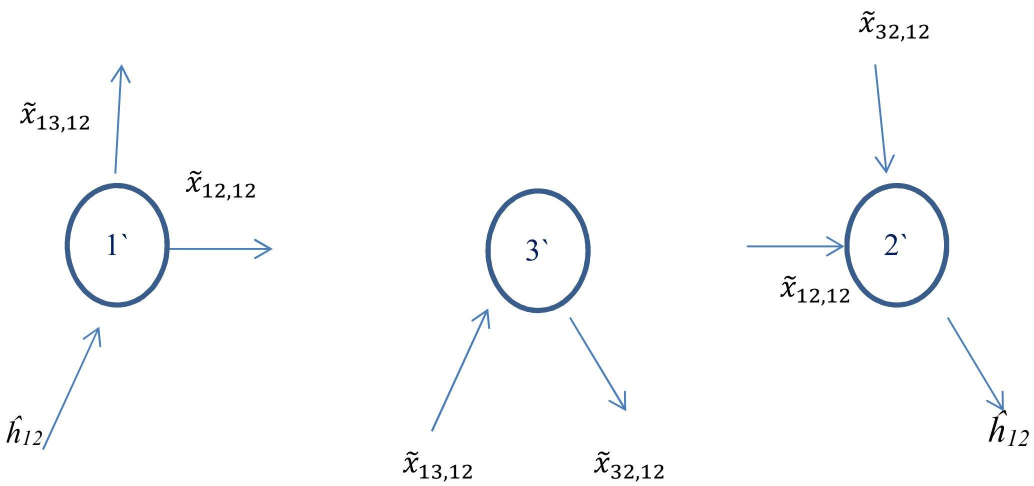


Figure 2-7: Demand flow view between nodes 1 and 2.

- link flow: traffic of one demand on each link
- flow conservation: Source, Destination, Transit node.

2.11.2.1 Optimisation in node-link formulation example

If we consider demand (1:2) and according to the flow conservation law, with the use of the convention that anything entering the node is negative and anything leaving is positive, we may write the following equation for node 1:

$$-\hat{h}_{12} - \tilde{x}_{21,12} - \tilde{x}_{31,12} + \tilde{x}_{12,12} + \tilde{x}_{13,12} = 0$$

It is important to consider that for each undirected link, one of its two flows is always equal to 0. This is because what really matters is the net flow on a link, which means:

$$\hat{h}_{12} - \tilde{x}_{12,12} - \tilde{x}_{32,12} + \tilde{x}_{21,12} + \tilde{x}_{23,12} = 0$$

Making use of the above observations we can write the set of flow conservation equations for demand (1:2) as:

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

If we now consider the demand (1:3) from node 1 to node 3, and demand (2:3) from node 2 to node 3, we obtain the following equalities:

$$\begin{aligned}
\tilde{x}_{12,13} + \tilde{x}_{13,13} \qquad &= \hat{h}_{13} \\
-\tilde{x}_{12,13} \qquad + \tilde{x}_{23,13} &= 0 \\
-\tilde{x}_{13,13} - \tilde{x}_{23,13} &= -\hat{h}_{13} \\
\tilde{x}_{21,23} + \tilde{x}_{23,23} &= \hat{h}_{23} \\
-\tilde{x}_{21,23} + \tilde{x}_{13,23} &= 0 \\
-\tilde{x}_{13,23} + \tilde{x}_{23,23} &= -\hat{h}_{23}
\end{aligned}$$

Finally, considering the links, the sum of all flows routed on a link must not exceed its capacity, which leads to a new set of (capacity) constraints. For example, for link 1→2 the capacity constraint is:

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

On the other hand, for link 1→3, we have the following capacity constraint:

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

In the same manner, the capacity constraints for the other links in the network can be formulated. Putting everything together along with the objective function, and eliminating those arc flow variables which are assumed to be equal to 0, we get the following formulation:

Minimise:

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

Subject to:

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

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

$$\tilde{x}_{21,23} \leq \hat{C}_{21}$$

$$\tilde{x}_{13,12} + \tilde{x}_{13,13} + \tilde{x}_{13,23} \leq \hat{C}_{13}$$

$$\tilde{x}_{23,13} + \tilde{x}_{23,23} \leq \hat{C}_{23}$$

$$\tilde{x}_{23,12} \leq \hat{C}_{32}$$

2.12 Summary

This chapter presented an overview of optical networks, and the evolution of the optical networks, as well as a general overview of the WDM switching technologies. It also considered special attention to the IP over WDM networks and reviewed the main work done in the literature on the energy efficiency of the optical core networks, such as green routing and energy efficient network design. This chapter also presented an overview of the linear programming and showed its general form as well as its capabilities. Furthermore, it presented the Simplex method of optimisation as an efficient algorithm for solving linear programming problems, and highlighted the main points that must be taken into consideration to set a linear programming model. Finally, this chapter showed network design problems with linear programming and provided illustrative examples.

Chapter 3 Energy Efficient Big Data Networks

3.1 Understanding Big Data

The term Big Data refers to the large-scale data sets collected from everywhere that are very complex and complicated, which became a problem for traditional data processing technologies. This immense amount of data brings the challenge of how, where, and when to handle, process, analyse, store, and transfer Big Data [65]. Big Data can be defined as "datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyse."[66].

3.1.1 Big data sources [67]

A huge amount of information comes from various sources in Big Data environments. Each one of the high-level categories of **Process-mediated**, **Machine-generated,** and **Human-sourced** information has different type data source. Five of the eleven various data types, outlined in Figure 3-1, are representing the Machine-generating category. Two of them which are **Application Servers log information** and **Network Activity and Research Information** has a significant contribution to Big Data environments. This is because they can be considered as the backbone to provide a reliable access to cloud-based environments, beside supporting access to the external and internal data centres.

The other three types that represent a considerable contribution to the Big Data environments are **Click-Stream information** from the online applications and mobile apps, **Geo-location information** from mobile devices and telematics, and **Sensors information** from telematics and manufacturing machinery. Examples of these data sources can be seen in XML and JSON formats and can be described as a multi-structured format. Three main data sources are associated with **Process-mediated data** in Big Data environments which are **Operational Application Data** such as point of sale, customer care, and supply chain, **External Augmentation Data** such as demographic or psychographic, and **Curated Business Information** such as a single version of truth customer or product information. Relation structured format is the general structure for each one of these types.

Human-sourced information is generated from four data sources. The top one that generates a huge amount of data in Big Data environments **is Human generated documents** such as emails and applications form documents. According to [68], there are more than 210 billion emails sent in a single day. The second type is Social Media data (e.g., Twitter, Facebook, Forum, Blogs, etc.). **Image content** such as pictures and videos come in the third place and finally, **Audio** information like streaming audio, call centre voice logs is the fourth type of Humana-sourced information sources. So, many researchers describe Big Data as "unstructured data" because of these four types. The reason is, for example, it is easy for people to read and recognise scanned images, forms, hand-writing documents while analysing such documents is difficult for the automated systems and database servers. Figure 3-1 illustrates big data sources.

3.1.2 The main of characteristics big data

Big Data is characterised by four main properties: volume, variety, velocity, and veracity as indicated in Figure 3-2 [69]. The need for extracting knowledge from Big Data has led to the creation of platforms that can process big data to deduce useful information insight from the huge amount of data that has immense volume, a wide range of variety, different velocity, and a different ratio of veracity.

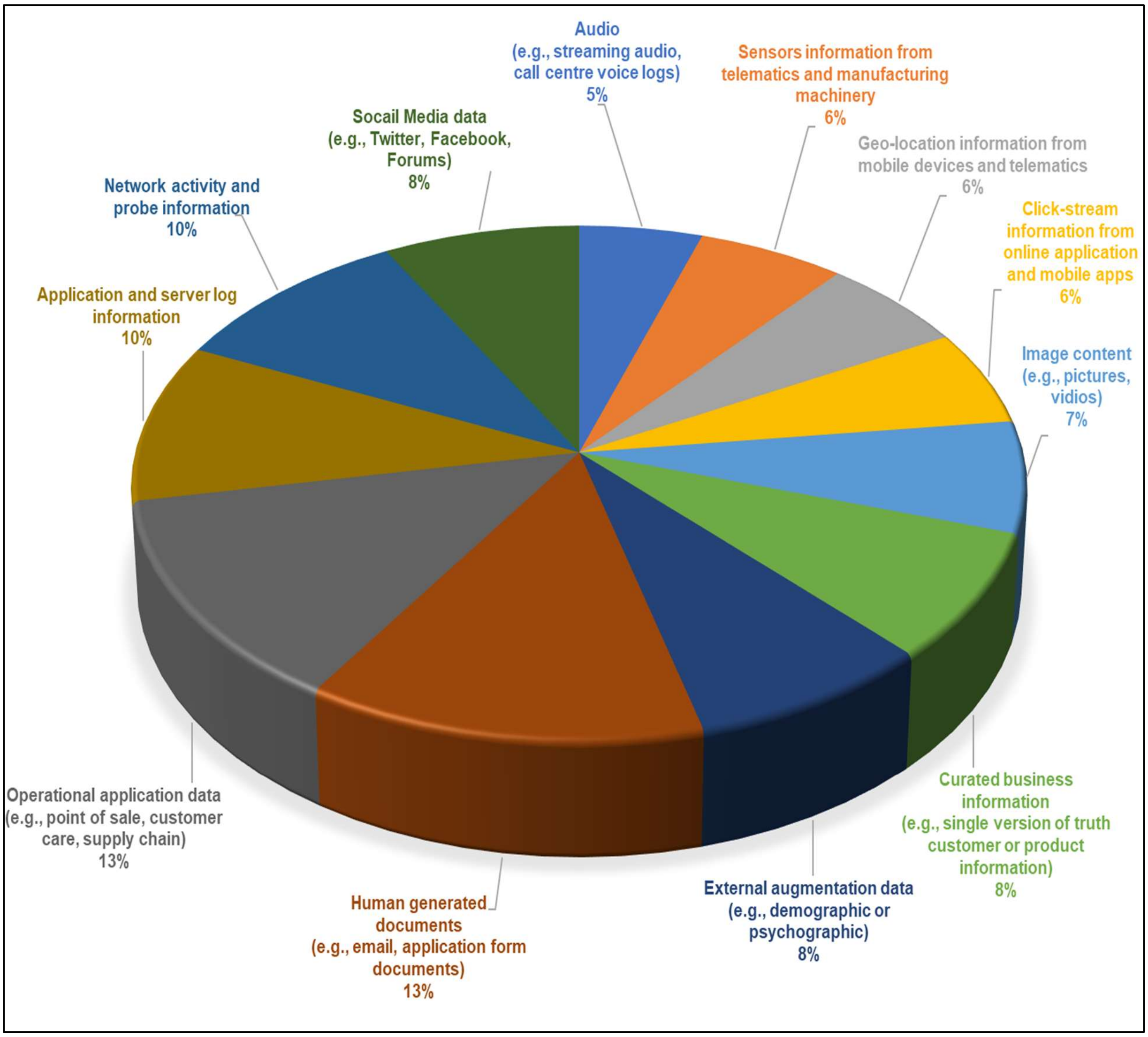


Figure 3-1. Big data sources [67].

3.1.2.1 Volume

By 2020, the foreseen amount of data that will be stored in the world is 35 Zettabyte (ZB), while the actual stored data was 800,000 Petabytes (PB) in the year 2000. Facebook processes more than 500 Terabytes (TB) every single day [70]. More than 210 billion emails are sent every day [68].

The daily Internet traffic is more than 2 Exabyte per second [71]. NASA data warehouse of Climate Change alone is projected to store more than 350 PB by 2030 [72]. The Square Kilometre Array is another project of NASA which opened in 2016 and streams 700TB/second of data from a single square kilometre.

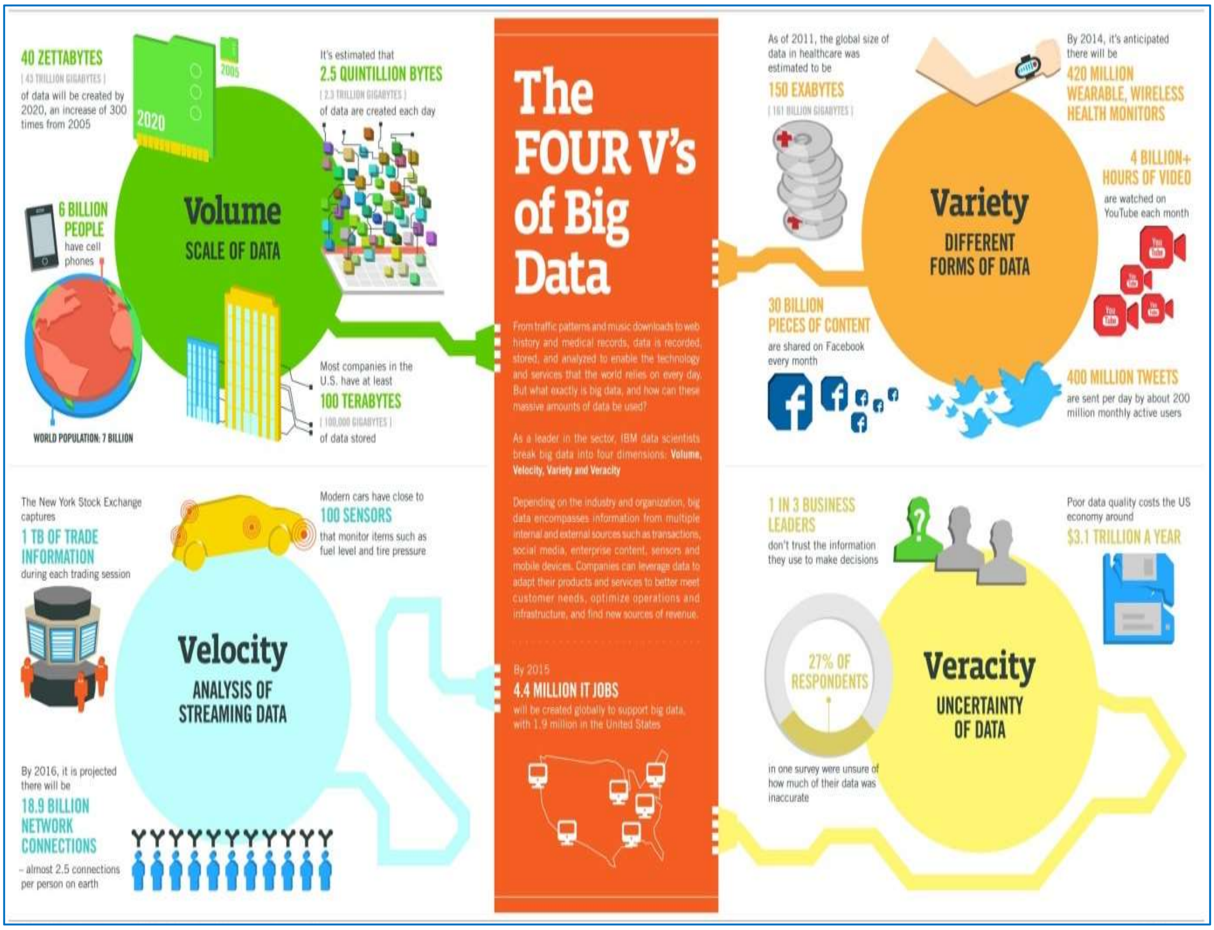


Figure 3-2: The four Vs of big data [69].

This amount of data is created from only three projects of NASA [72]. Many enterprises and scientific research generate hundreds of terabytes of data every hour during the year. The challenge of Big Data is that a lot of generated data nowadays is not analysed at all or addressed to extract insight. Figure 3-3 [65] shows that the amount of processed data is decreasing compared to the dramatic increase in the amount of data being generated. Massive data volumes need analysis, management, and processing technologies, otherwise, organisations could be overwhelmed by the Big Data.

Figure 3-3: Volume of data is increasing, while the percentage of data that can be processed is declining [65].

3.1.2.2 Variety [73]

The new challenge that data centres should deal with is variety. Variety simply means all types of data. The boom in sensors industry, smart devices, and social networks technologies has led to dramatic increase in data volume and complexity. The complexity of data comes from not only traditional data as there are many unstructured, semistructured, and raw data sources that contribute a lot in increasing data complexity. Emails, sensors data, blogs, social media, forums, video and audio streaming, log files, search indexes, and more can make tradition database systems struggle to store, process and analyse to obtain the useful information being generated form such types of data since they are not related to the relational database technologies. The successful organisation should be able to handle the variety of data which includes traditional and non-traditional data such as text, video, audio, sensors data, etc.

3.1.2.3 Velocity [73]

The conventional definition of the velocity of data is that it is the speed of creating, collecting, archiving and storing all data types and how quickly can organisations manage all these together. That is true when enterprises were dealing with terabytes of data. Today things change: enterprises are dealing with petabytes and Zettabyte and will deal with Exabytes soon. The dramatic increase in the velocity and volume of data being created from streaming systems such RFID sensors had made it impossible to be handled by the traditional systems. This led to thinking differently when dealing with the velocity problem. A simple example of processing data in real time is GPS because it updates the location information frequently and this can give an accurate statistical picture of the number of people living now in a city in space and time.

A new definition of velocity relates to data in motion and the speed at which the data is fluxing. Effective handling of Big Data requires analysis, in this case, while the data is flowing and not only after the data storage. This is because there is an immense amount of produced data which has a very short life cycle and this requires extracting the useful knowledge from Big Data in near real time.

3.1.2.4 Veracity

The veracity of Big Data is the most serious challenge to data scientists compared to volume and velocity since they need to distinguish between meaningful data and noise and meaningless /

inconsistent data [74]. Significant efforts are need to keep dirty data out from organisations databases. A key motivation here is that low quality of data is costing the U.S. economy $3.1 trillion each year [74].

3.2 Hadoop-MapReduce: The Storage and Processing Platform for Big Data

Map Reduce [75] is a software model used to perform certain processes on huge volumes of data such as word counting, text searching, sorting, merging etc. Two main tasks are performed in patch processing of the data sets: Computing and combining. Computing is the process of assigning a specific computational task on each record. It requires mapping an operation on each record. Combining is the process of consolidating and merging the output results from mapping processes. Having divided the large input volume data into many independent different *Chunk*s, the Map task is executed in a parallel manner on each *Chunk* to produce intermediate key-value pairs. This intermediate data is grouped by performing shuffle processes and is served as input to the user Reduce function which is responsible for reducing/merging all these intermediate key-value pairs to produce the final associated key-value result.

3.2.1 Hadoop-MapReduce

Hadoop [76] framework is an open source Linux based platform written in Java and distributed under Apache License. The basic idea of Hadoop has come out from Google File System GFS [77] and Google MapReduce [75]. It operates under commodity hardware which is relatively inexpensive compared to special expensive hardware such RAID systems. Its high-level structure consists of two main parts to address Big Data challenges, the computational part which is a software programming framework called MapReduce that is responsible for analysing and processing efficiently petabytes of data sets; and a storage part which is the Hadoop Distributed File System (HDFS). HDFS handles consistently an immense amount of data and transforms it to the user application at high bandwidth in a reliable way. The utility and efficiency of Hadoop comes from its capability of distributing both data processing and storage units across multiple hosts to perform and execute an efficient parallel segmentation, processing, and reassembly on the huge amount of data. Moreover, it has an effective redundancy technique that is able to overcome machine failure and recovers data automatically and immediately by performing automatic replication to user data [78].

3.2.2 Implementing Hadoop-MapReduce [79]

Architecturally, Hadoop-MapReduce clusters consist of three major layers which control the parallel patch processing and computational units (i.e., the MapReduce framework), and the patch storage unit which is the HDFS. These three layers are Client machines layer, Master nodes layer, and Salve nodes layer as shown in Figure 3-4. The master layer includes servers that supervise two main functions: storing huge volumes of data in the HDFS and this job is done by a unit called Name Node; secondly, performing parallel processing for all HDFS data and this job is done by a node called Job Tracker that runs MapReduce. The slave layer consists of most cluster machines and is responsible for doing the hard work. In each slave machine of the small cluster, there are two major key elements that maintain the communication with the master layer: Data Node that handshakes, communicates, and receives orders from the Name Node; and Task Tracker that handshakes, communicates, and receives orders from Job Tracker. In medium and large clusters, the Task Tracker and the Data Node are separated. The Client machines layer is located above the master and slave layers. Hadoop platform is installed on the client. All Hadoop parameters such as replication factor, cluster size, file size, number of mappers, number of reducers, etc. are configured by the user to draw the roadmap for the Hadoop cluster. Moreover, it is responsible for loading data into the cluster, and assigning MapReduce jobs, and how these data should be treated. Also, it observes and oversees the results that come out from the underlying layers.

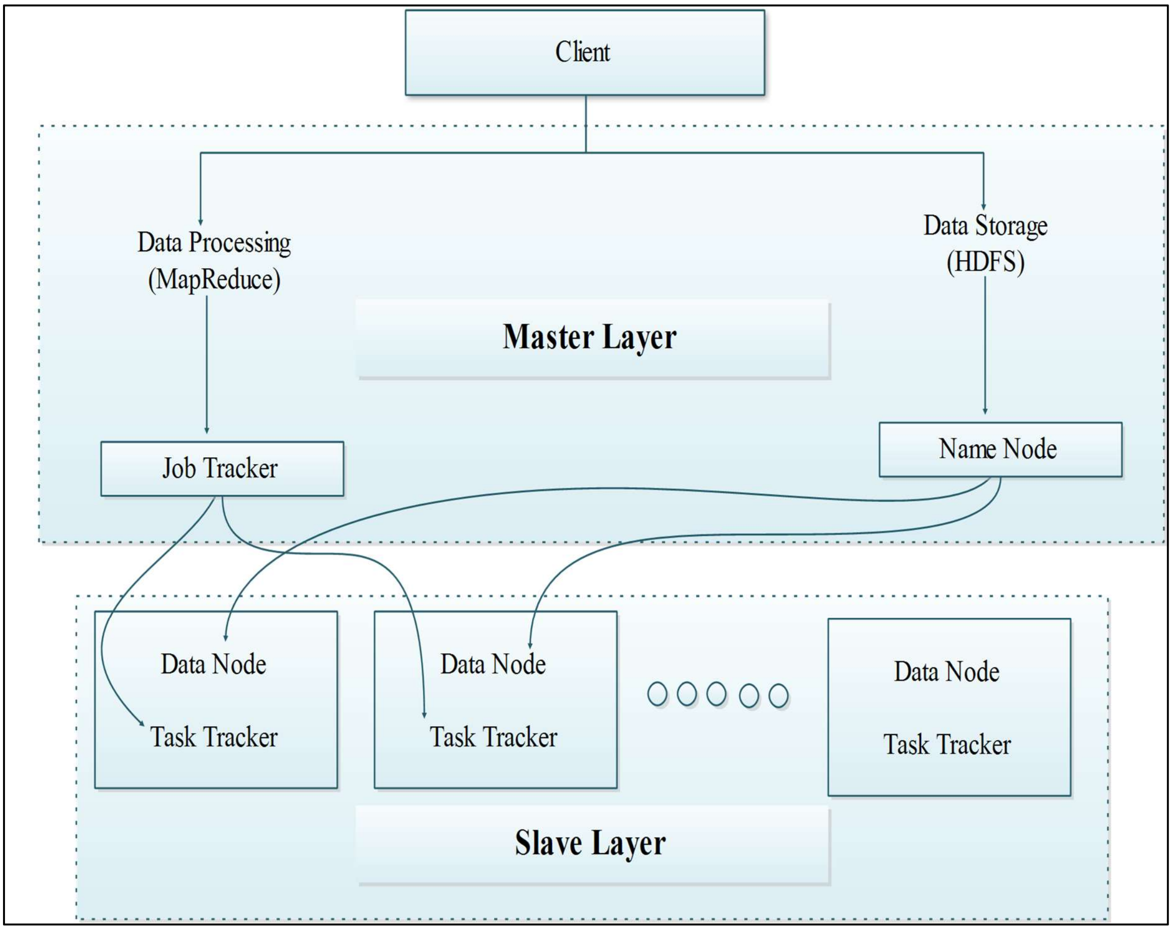


Figure 3-4: Hadoop MapReduce architecture.

- **Job Tracker and Task Tracker (MapReduce server)**

After finishing the process of loading the file and replicating the blocks related to that file, the client now signals the JobTracker to start the computation process. JobTracker consults the NameNode about the locations of these blocks in the DataNodes. Then it assigns a MapReduce task to all TaskTrackers running on those DataNodes.

Like DataNodes, TaskTracker heartbeats JobTracker informing the status of their Map task. Having finished each Map task, the results are stored locally and temporary in its node. This is called the intermediate or shuffle data. This data then is aggregated over the network to other nodes that execut Reduce tasks. Finally, the MapReduce task result is provided back to the client.

- **Replica and Rack Awareness**

When a block of an HDFS file is to be written into DataNode, it is replicated 3 times (if the user has configured the replication factor to be equal 3 which is the default Hadoop replication value) and is stored in a way that no DataNode in a specific rack has more than one copy of the same block of a file, and no more than two copies are stored in the other rack. The result is one replica in a rack, and two replicas are stored in two DataNodes in another rack. The strategy in copying the block from node to node in the same rack is done by the NameNode by sending a signal to one of the nodes to grab a copy of the block from the other node. This replication scenario is pipelined according to the minimum distance from the client. This is useful to improve the cluster performance by minimising the writing cost and maximising data protection in case of DataNode and/or in case the rack switch goes off for any reason such as power failure.

3.3 Network Architecture Utilised by Big Data

Typically, big data relies on a network architecture that compromises three network levels as shown in Figure 3-5. The access networks (User-Level) are directly connected to the end user devices and are responsible for collecting vast amounts of unprocessed data generated by many Internet-connected devices. Data centre networks (Destination-Level) are in charge of storing and processing big data using servers (up to 1 million servers are currently typical in a large data centre) that execute high performance parallel computational jobs such as MapReduce [75]. Aggregation networks (Bridge-Level) act as a bridge that connects data sources with remote data centres. This intermediate level receives the large volumes of raw data from the access network and forwards it to the remote data centres which in turn send back the results of the processed big data to the users. In this section, we introduced the main challenges associated with big data networks and review technical solutions that aim to reduce the communication cost in big data networks. Then, we introduce recent work we carried out to improve energy efficiency in big data networks.

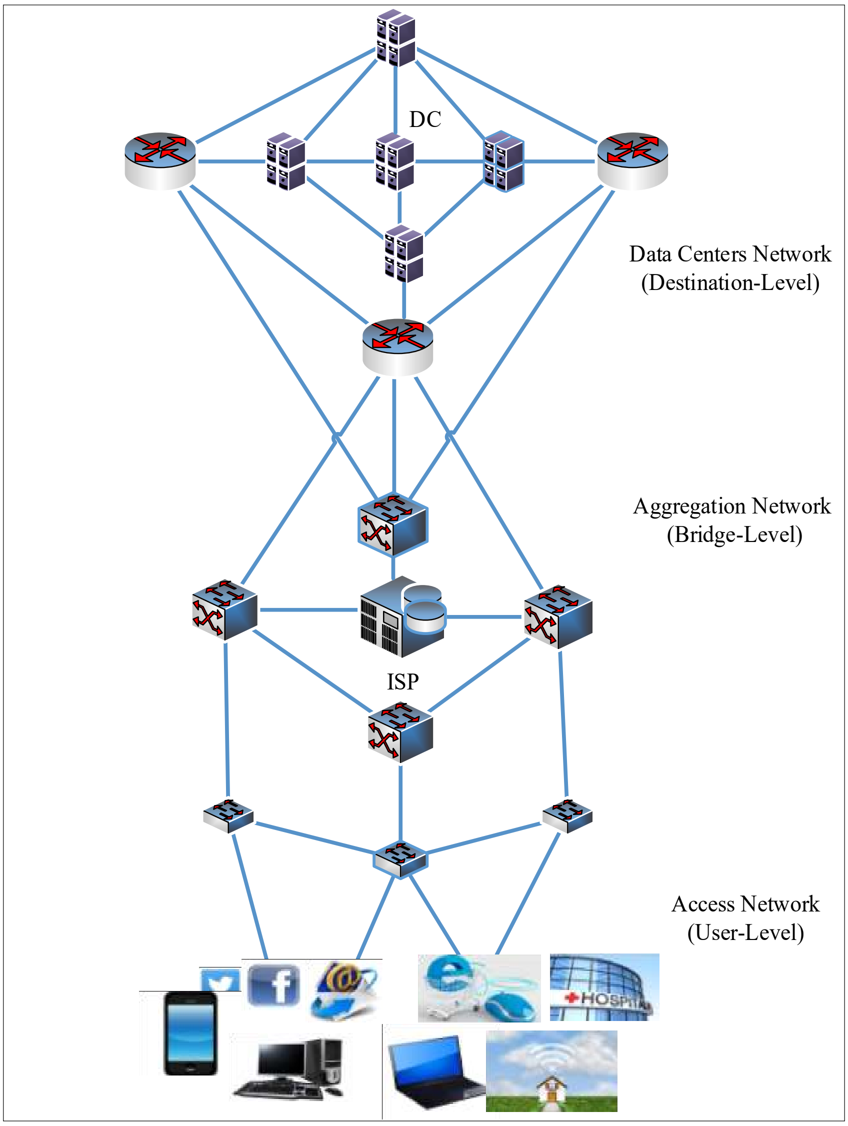


Figure 3-5: The network architecture utilised by big data.

3.3.1 Challenges of big data processing

Big data is currently generated at such high rate that a significant part of the data is not analysed at all or processed to extract insight. Recall Figure 3-3 that showed a decrease in the ratio of processed data to the overall huge volume of big data being created [65], and this results in:

- Massive networking power wastage due to transmitting unprocessed data when in many cases there is no interest in the raw data, but only in the knowledge extracted from such data. In some cases, such knowledge is easy to extract, for example, events.
- High financial cost due to resource wastage.
- Wasting storage and other network resources such as bandwidth and physical equipment to store and deliver minimally analysed data.

Managing massive data volumes calls for new processing and communications approaches. In addition, gathering and transmitting big data is exposing new challenges in how to efficiently and economically transport big data over the network with acceptable service quality while providing adequate processing and storage resources. For instance, medical sensor data must be transferred and processed in a very tight timeframe and the results have to be sent back to the hospital or patient wearable device as soon as possible before a health risk materialises. Such application-level constraints impose even more challenges and hard trade-offs on the energy efficiency that can be attained from optimising big data processing and networking.

The main challenge in terms of the volume of data is that the real interest is typically not in the data, but in the knowledge, that can be extracted from the data. Therefore, the challenge in terms of processing and networking lies in designing network architectures and algorithms that enable the data to be processed as close to the source as possible to reduce its volume and transmit the lower volume “knowledge”, which we refer to as “Info” resulting in lower network resource requirements and lower power consumption. Consider for example heart rate monitoring. The resulting waveform can have a large volume when measured over a long period of time. The waveform hopefully indicates for years that the person is fine and therefore its transmission to emergency services or to data centres is redundant data. If the data is processed near the source, then a simple message made up of few bits of information (knowledge) can be sent either to indicate that the person is fine or that emergency services should be directed to the person’s location. To capture the range of possible applications, we considered different data reduction factors that relate the volume of the original data to the volume of the knowledge extracted, all measured in bits.

3.4 Related Work: Networks and Big Data Processing Solutions

The challenges highlighted in the previous section led researchers to think critically about feasible solutions to reduce big data communication costs, such as the cost of energy needed for data transfer and processing. For instance, possible questions related to the potential of processing a large part of the data locally at the User-Level, versus forwarding big data to the data centres for storing and processing.

Alternatively, big data may be transferred and gradually processed in different intermediate nodes of the big data network. In the following sections, we review some of the recent approaches used to handle big data in access networks, optical core networks, and data centre networks.

3.4.1 Big data in social networks

Extracting knowledge from big data in social networks requires intricate processing. Researchers are working extensively to develop efficient approaches that address such challenges.

In [80], Z. Yun et al. proposed a comprehensive multi-dimension approach to extract knowledge from user behaviour data in social networks and two main big data characteristics were considered as their primary concern: Variety and Veracity. They covered comprehensively three main knowledge dimensions of data: Transactional dimensions such as sender, receiver, length, etc., Data Quality Dimensions such as timeliness, completeness, objective, reputation, readability, accessibility, etc., and Data effect dimensions such as emotions, valence, motive, intentions, incitement, etc. Also, they developed the hypergraph model which has already been applied in social tagging system and used it in social networks analytics.

T. Wei et al. in [81] used personal ad-hoc clouds which involved individuals' activities and behaviour in social networks to analyse social network-sourced Big Data. In [82], X. Han et al. developed a Big Data model than can be used in recommendation systems. The input to this model is the output of analysing social networks data using MapReduce. In [23], Y. Chen et al analysed and compared two MapReduce traces obtained from Facebook and Yahoo!. They evaluated the cluster performance of these realistic workloads to present a useful tool for designing and managing MapReduce systems. This work was extended by Y. Chen et al in [83], to develop the Berkeley Energy Efficient MapReduce (BEEMR) workload manager which is motivated by the significant analysis of real-life large dataset traces at Facebook.

3.4.2 Big data in I/O environments

In [84], P. DeMar et al. described the network research activities for Big Data at Fermilab. Also, they described the current R&D works and challenges of processing and movement across 100 GE wide-area networks of exascale datasets obtained from the Large Hadron Collider (LHC) Compact Muon Solenoid (CMS) experiments. Fermilab R&D experiments involve crucial work such as optimising system performance in the network I/O environments, improving application performance in the 100GE network, and monitoring and reconfiguring network path. S. U. Zai et al. developed in [85] a Multipath Transmission Control Protocol (MPTCP) that provides better throughput when dealing with Big Data application compared to the single path TCP. Also, they considered the Couple Congestion Control (CCC) with MPTCP to improve throughput. In [86], J. Liu et al. proposed an algorithm to reduce Big Data movement by segmenting, analysing, and reusing the same results which were obtained from different segments running similar tasks. A. Rajendran et al. presented results in [87], to maximise throughput in large-scale data movement through optimising and careful tuning of scientific applications and middleware. They discussed in detail the performance of data transfer at 100 Gbps speed and 53 ms latency. Furthermore, they measured the performance of both applications of High Energy Physics (HEP) and data transfer middleware (GridFTP, Globus Online, Storage Resource Management, XrootD and Squid).

3.4.3 Big data storage and segmentation

In [88], P. Bajcsy et al. presented a methodology of image computation and processing on a Hadoop cluster. They designed a benchmark and evaluated it on many image sets; each set sized over one half of Terabyte. The aim of this method is to improve image processing on Hadoop clusters. K. Lee et al. presented in [89], an optimised scalable framework called SPA for distributed processing of big Resource Description Framework (RDF) graph data in the cloud. They evaluated this data partitioning framework extensively through many experiment and obtained an efficient result for partitioning, distributing, and processing big RDF datasets. In [90], M. Shtern et al. introduced a novel approach called *DaasPatcher ecosystem* to share segmented Big Data on the cloud. This ecosystem has the ability to refine and improve enhanced data-as-a-service (eDaaS) in the clouds by determining the data that providers are willing to share with different types of clients. In [91], R. Zhou et al examined the overall effect of deduplicating Big Data workloads. Also, they studied the advantages and disadvantages of various deduplication layers, locations, and granularities. Furthermore, they investigated the relation between energy expense and redundancy degree.

3.4.4 Local gathering and processing of big data in access networks (user-level)

In this section, we review effective techniques to help reduce the big data transmission cost by processing big data locally in the access networks (User-Level).

3.4.4.1 MapReduce on IoT

Generally, IoT devices generate small volumes of data per device but owing to the large number of IoT devices expected, the overall data volume can be large. The immense amount of data generated by the IoT environment is transferred from IoT source data nodes to high-performance server clusters in clouds and data centres that perform extensive parallel processing tasks such as MapReduce to extract useful information to be sent to users. However, this is not a cost-effective approach in terms of transmission because moving such huge amounts of data from the users to the remote data centres requires significant communication resources and causes network congestions problems which have a negative impact on QoS, especially in the case of delay sensitive data that must be processed in near real time. Therefore, moving processing elements closer to the big data sources to execute MapReduce jobs is essential to overcome such problems. Ichiro [92], presented a MapReduce framework to locally process as much data as possible on multiple nodes of the IoT environment rather than transmitting the data to data centres.

3.4.5 Big data in aggregation networks (bridge-level)

The aggregation network compromises Internet Service Providers (ISPs) and Content Distribution Networks (CDNs). It is located beyond the access networks and is responsible for collecting and delivering the large data volumes to the data centres to be processed. In addition, it is responsible for pushing high volume applications' data, such as IPTV, online social networks, photos, and video streaming, from the data centres into user devices. Such huge and costly traffic creates significant challenges to the aggregation network. Hence, it is of high importance to investigate how to process big data before the aggregation network to reduce traffic that is forwarded to the data centres through the aggregation network.

3.4.6 Processing and transporting big data in geo-distributed networks (datacentres level)

In the following section, we discuss big data queries, scheduling and transporting big data in bulk, running MapReduce, and minimising the cost of big data processing. The techniques developed here can be directly or indirectly harnessed to minimise power consumption in geo-distributed data centres.

3.4.6.1 Query evaluation of big data

Query evaluation refers to the amount of computing, storage, and communication resources allocated to a query, a search or a process run on big data. To process big data queries across multiple data centres, it is important to efficiently evaluate the resources required to meet the demands of such queries. For instance, if the source data of a specific query is located at two data centres, then a number of options are considered [22]: The First option that is to be considered is to examine whether it is a cost-effective approach in terms of network resources to replicate millions of gigabytes of source data from one data centre to another and process the query at one of them, especially, if the data centres are at locations that are geographically far apart. Second, if there are enough computation and storage resources for such a query to be served at only one data centre, or if it is more efficient to be served at a third data centre that has enough resources and is not far from both original data centres. Alternatively, the communication cost can be reduced while responding to big data queries by employing multiple data centres. A substantial question for query evaluation of big data processing in geo-distributed clouds is how to satisfy as many queries as possible while keeping the communication cost to a minimum. Since the available bandwidth between different data centres varies over time and the communication cost is substantial, due to the large amounts of data transferred, there is a need to develop methods to minimise the communication cost when responding to big data queries. An effective approach to resolve big data queries in a distributed system is described in [22]. This approach aims to satisfy as many queries as possible over several time slots while keeping the communication cost at a minimum level. At the beginning of each time slot, this approach checks whether the available resources of the data centres are enough to meet the query demand, and if yes, the query is processed, otherwise it is excluded.

3.4.6.2 Scheduling and transferring big data bulk

Scheduling solutions for big data bulk transfer across geo-distributed data centres are required to satisfy transmission tasks that have different urgency levels. Such tasks can be optimally and dynamically arranged to fully exploit the available bandwidth at any time. Full bandwidth utilisation is essential to reduce the number of links over time. The flexibility of transmission scheduling provided by Software Defined Networks (SDN) enables dynamic optimal routing of individual big data *Chunk*s within each transfer by temporarily storing them at intermediate data centres and transmitting them at distinct scheduled times to reduce bandwidth requirements [93].

3.4.6.3 Sequence execution of MapReduce jobs

This deals with finding the best ways of processing big data jobs according to either the delay cost or another monetary cost, e.g. electricity cost. Sequencing techniques, such as Geo-Distributed MapReduce (G-MR) [19], are crucial in geo-distributed data centres to guarantee efficient big data processing. The process of determining the optimum execution path for a large data set is complex because every possible data motion should be considered. Thus, G-MR uses a data transformation graph (DTG) algorithm which helps find the best execution path for performing a sequence of MapReduce jobs in geo-distributed data sets.

3.4.6.4 Joint optimisation of job distribution and routing

Optimally distributing the processing of big data jobs among different nodes, taking into account routing paths, is more efficient for the network in terms of computation and communication requirements. A good example is proposed in [20] which takes into account the joint optimisation of four main aspects in geo-distributed data centre so that the overall computation and transmission cost is minimised:

- Task assignment: Here the task that should be served first is determined together with the tasks service order.
- Data placement: This deals with the best processing locations for big data *Chunk*s.
- Data Centre Resizing (DCR): This establishes the minimum amount of storage and processing elements required to serve big data *Chunk*s.
- Routing: Here the best transmission path for transferring the *Chunk*s is determined.

3.4.6.5 In-Network processing, network function virtualisation (NFV), and grid computing

In-network processing is proposed in [94] to achieve network-awareness to reduce bandwidth usage by custom routing, redundancy elimination, and on-path data reduction. In [95], Global Network Operators outlined the benefits, enablers, and challenges for Network Functions Virtualisation (NFV) such as reducing equipment cost and power consumption through equipment consolidation and by exploiting the economies of scale in the IT industry.

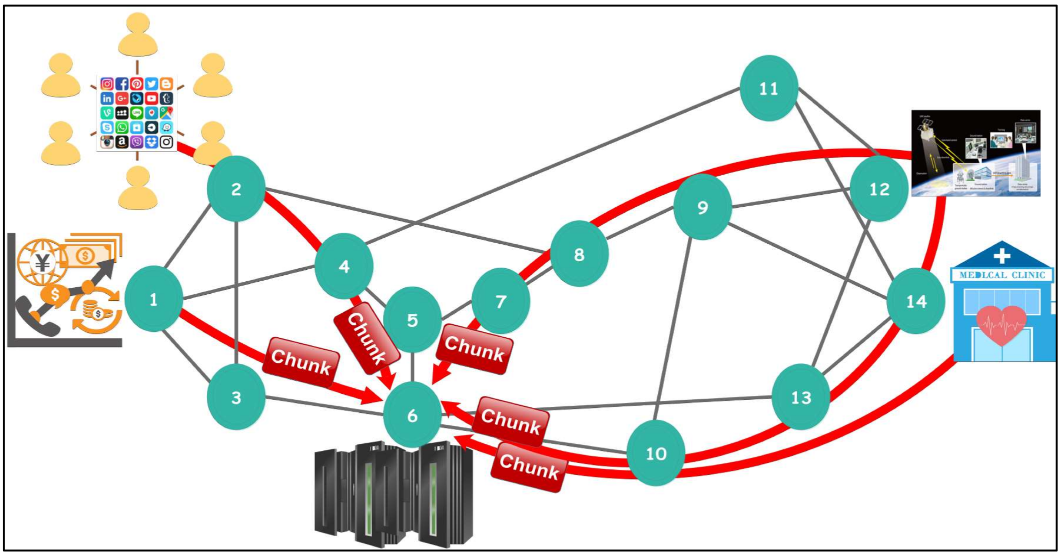
The authors in [96] presented a phased solution approach to dimension the computing resources deployed in Grid infrastructure. The dimensioning problems that usually arise in Grid computing include the number of servers to be provided, where to place them, and which network to install to interconnect server sites and users generating Grid jobs. In [16], the authors developed an energy efficient cloud computing framework in IP over WDM core networks.

3.4.7 Energy efficient cloud computing services in optical networks and big data transfer in elastic optical networks (EON)

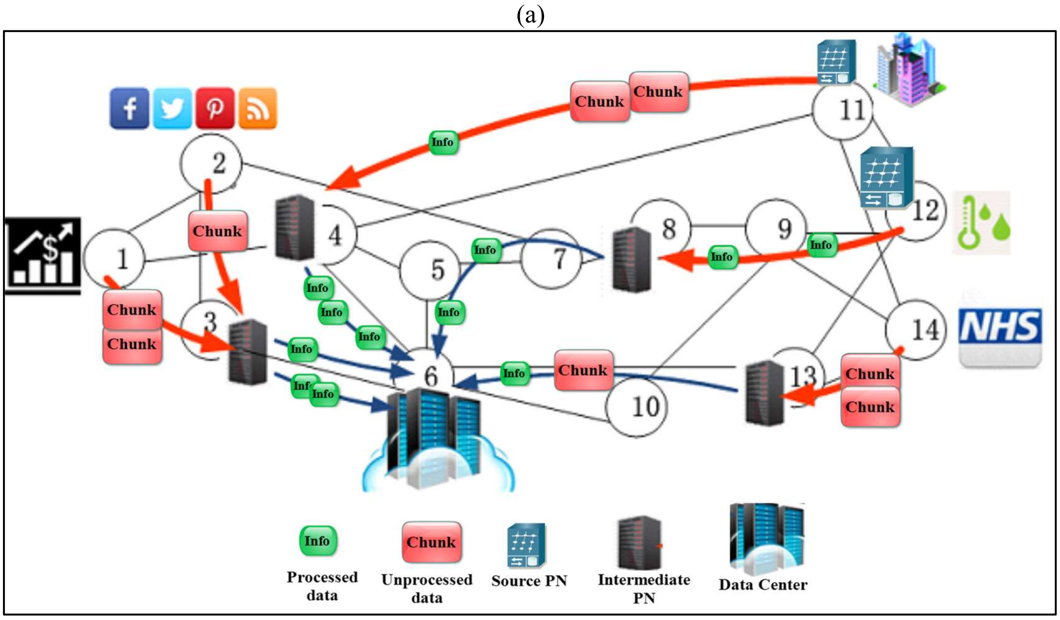
In [16], the authors presented a framework for energy efficient cloud computing services in IP over WDM core networks. In [97] mixed integer linear programming (MILP) models and heuristics are developed to minimise delay and power consumption of clouds over IP/WDM networks. The authors of [98], exploited anycast routing by intelligently selecting destinations and routes for users traffic served by clouds over optical networks, as opposed to unicast traffic, while switching off unused network elements. A unified, online, and weighted routing and scheduling algorithm is presented in [99] for a typical optical cloud infrastructure considering the energy consumption of the network and IT resources. In [100], the authors provided an optimisation-based framework, where the objective functions range from minimising the energy and bandwidth cost to minimising the total carbon footprint subject to QoS constraints. Their model decides where to build a data centre, how many servers are needed in each datacentre and how to route requests. In [101], the authors described the drivers, building blocks, architecture, and enabling technologies for elastic optical networks. The authors in [102] suggested that efficient bulk-data transfer in elastic optical networks (EONs) can be achieved with Malleable Reservation (MR). The authors in [18] discussed the technologies needed for realising highly efficient data migration and backup for big data applications in elastic optical inter-data-centre (inter-DC) networks. In [103], the authors investigated offline and online Routing and Spectrum Assignment (RSA) problems for anycast requests in elastic optical inter-DC networks by formulating an Integer Linear Programming (ILP) model and proposed several heuristics based on single-DC destination selection. In [104], we presented preliminary results to demonstrate the impact on network power consumption of processing and transferring big data in bypass IP over WDM networks. We considered one big data type from the MapReduce platform that was obtained from the log files of MapReduce clusters from Facebook [23]. In this type, the volume of the output of the reduced process is very small compared to the input of the mapping process. We investigated improving the energy efficiency of big data networks by processing this data type progressively in processing nodes (PNs) of limited processing and storage capacity along the data journey through the IP over WDM core network to the DCs. The amount of data transported over the core networks was significantly reduced each time the data was processed; therefore, we referred to such a network as an Energy Efficient Tapered Data Network.

3.5 The Proposed Energy Efficient Big Data Networks (EEBDN)

The concept of the EEBDN is illustrated in Figure 3-6. Figure 3-6-a displays a Classical Big Data Networks (CBDN) where the processing of big data Chunks (the raw big data before processing) is achieved inside DCs after being generated and forwarded by the source nodes (an example of a source node in Figure 3-6-a is National Health Service (NHS) node #14). In the EEBDN, shown in Figure 3-6-b, IP over WDM core nodes are attached to PNs (e.g., node #12) that can process Chunks and extract useful knowledge such as transportation and weather trends. We refer to the extracted knowledge after processing the raw big data Chunks in this work as info. These Info pieces are optimally transferred through energy efficient routes from the PNs to the DCs. The structure of a PN is similar to the cloud structure presented in [16]. It consists of a limited number of servers, storage (to store Chunks) and internal switches and routers. A PN is capable of edge processing the locally generated data and the data generated by other nodes and forwarding the results (Infos) to the DCs. The capacity of a PN is limited by the available space to build the PN inside the network centre. Note in Figure 3-6-b that the data generated by the source nodes can either be Chunks, Infos or both. The latter is the case if the source has processing elements. This type of source core node is referred to as a Source PN (SPN). On the other hand, the processing capability located at intermediate core nodes is referred to as Intermediate PN (IPN). IPNs perform the progressive processing of Chunks generated by other SPNs that did not perform local processing due to insufficient processing resources.



(a)



(b)

Figure 3-6. (a) The CBDN. (b) The EEBDN [104].

We assumed, for realistic considerations, that each PN's processing and storage capacity varies from one PN to another. The DCs' capacities are, however, large enough for the central storing and processing of the remotely forwarded *Chunks* from the PNs. When *Chunks* are processed inside DCs, the corresponding results (i.e., *Infos*) are stored there. It is essential to mention here that the amount of computing resources required to process the *Chunks* in both approaches (i.e., in the EEBDN and in the CBDN) remains the same. In addition, in both approaches, the processing resources are utilised in an energy efficient manner where we calculated the processing power consumption assuming that the minimum number of servers are utilised and also that slicing techniques are employed [105].

However, the energy savings obtained from the EEBDN approach comes from the optimal distribution of the processing resources among the network core nodes. Note in Figure 3-6-b *Chunks* can be processed either in SPNs or IPNs. Once the PNs' servers are fully utilised, no more edge or intermediate processing is performed. The centralised processing inside the DCs dominates the processing of big data since DCs' capacities are large enough for the central storing and processing of the remotely forwarded *Chunks* from the PNs.

The PNs are located close to the user while data centres are typically in central locations in the network and may be far from the user. Therefore, PNs enable edge processing of big data, hence saving power. PNs are different when compared to data centres in additional ways. PNs may have a limited set of software packages; they are small and hence may be less energy efficient. These are among the constraints considered in our progressive processing approach. In big data analytics, because of the variety in big data applications, the ratio of the size of *Infos* to the size of *Chunks* (i.e., output/input) is diverse [23]. For example, *Infos* size << *Chunks* size (i.e., the ratio can be << 1) in many big data applications, such as video monitoring in surveillance cameras that capture points of interest. While on the other hand, *Chunks* size ≈ *Infos* size (i.e., the ratio ≈ 1) in other big data job types, such as Call Detail Record (CDR) data produced by a telephone exchange or other telecommunications equipment. Further, there are several mixtures of jobs performed in big data analytics where the ratio is in between. The *Info* extracted can be the presence or absence of a person or an object in the video. We refer to this ratio as the Processing Reduction Ratio (PRR).

PRR is the ratio of the final reduced data (the information or knowledge of interest) to the original data size. For example, a 1 MB video clip may be the data. The presence or absence of a person in the clip may be the knowledge of interest. If the video is processed and a 1 kB packet is sent instead of the video clip, then PRR is 0.001. Therefore, we introduced Equation (3-1) where the volume of the *Chunks* is multiplied by different PRRs to produce the *Infos* carried by the corresponding *Chunks.* For instance, in MapReduce jobs, a *Chunk* of 1000 Gigabit (Gb) and PRR of 0.001 results in *Info* of 1 Gb [23]. Accordingly, significant network power saving is achieved if such *Chunks* are processed locally in the edge (SPNs) and progressively in the IPNs. Thus

$$Volume\ of\ Info = PRR \times Volume\ of\ Chunk \tag{3-1}$$

To provide a clear picture of the relation between the output data size and the input data size for different big data jobs, we summarised the table that appeared in [23], which is obtained from a Facebook cluster of a MapReduce trace file in a two week period as shown in Table 3-1.

Job counter	***Chunks* (input) size**	***Infos* (output) size**	**PRR**
1145663	6.9 MB	60 KB	0.0086

7911	50 GB	61 GB	1.19
11491	1.5 TB	2.2 GB	0.0014
670	2.1 TB	2.7 GB	0.0012
1876	711 GB	860 GB	1.21
169	2.7 TB	260 GB	0.096

Table 3-1. MapReduce Facebook Cluster Summary [23].

Note that we did not consider big data job types where PRRs > 1 (i.e., *Info* size > *Chunk* size) since our main objective is to reduce the network power consumption and it is axiomatic that such a type of job is directly forwarded to the DCs skipping our PNs. Forwarding such *Chunks* directly to the DCs confines the extra traffic generated by the *Infos* to the inside of the DCs only. This leads to upscale the big data traffic in the network. Therefore, reducing the big data traffic overhead in big data processing using control node with intelligent algorithm is important. In this thesis, we present new proposals and scenarios to illustrate the impact of the 4Vs of big data on the network power consumption and to optimise the DCs' locations by exclusively dealing with each V.

3.5.1 EEBDN: An illustrative example

To illustrate the concepts of the proposed EEBDN, consider the example network shown in Figure 3-7. There are four zones in Figure 3-7, with each connected to a certain PN, where each PN receives a different number of Chunks depending on its zone user population. For instance, zone 2 generates more Chunks compared to zone 4 that has a lower user population. The PN connected to a certain zone is referred to as a source PN (SPN) as it is the first PN in which *Chunk*s are received from its corresponding zone and locally or centrally processed. Each SPN can locally process a different maximum number of *Chunk*s depending on its processing, storage and internal switches and routers capacity. The remaining *Chunk*s that cannot be processed locally in an SPN are forwarded either to another optimally selected PN or a DC.

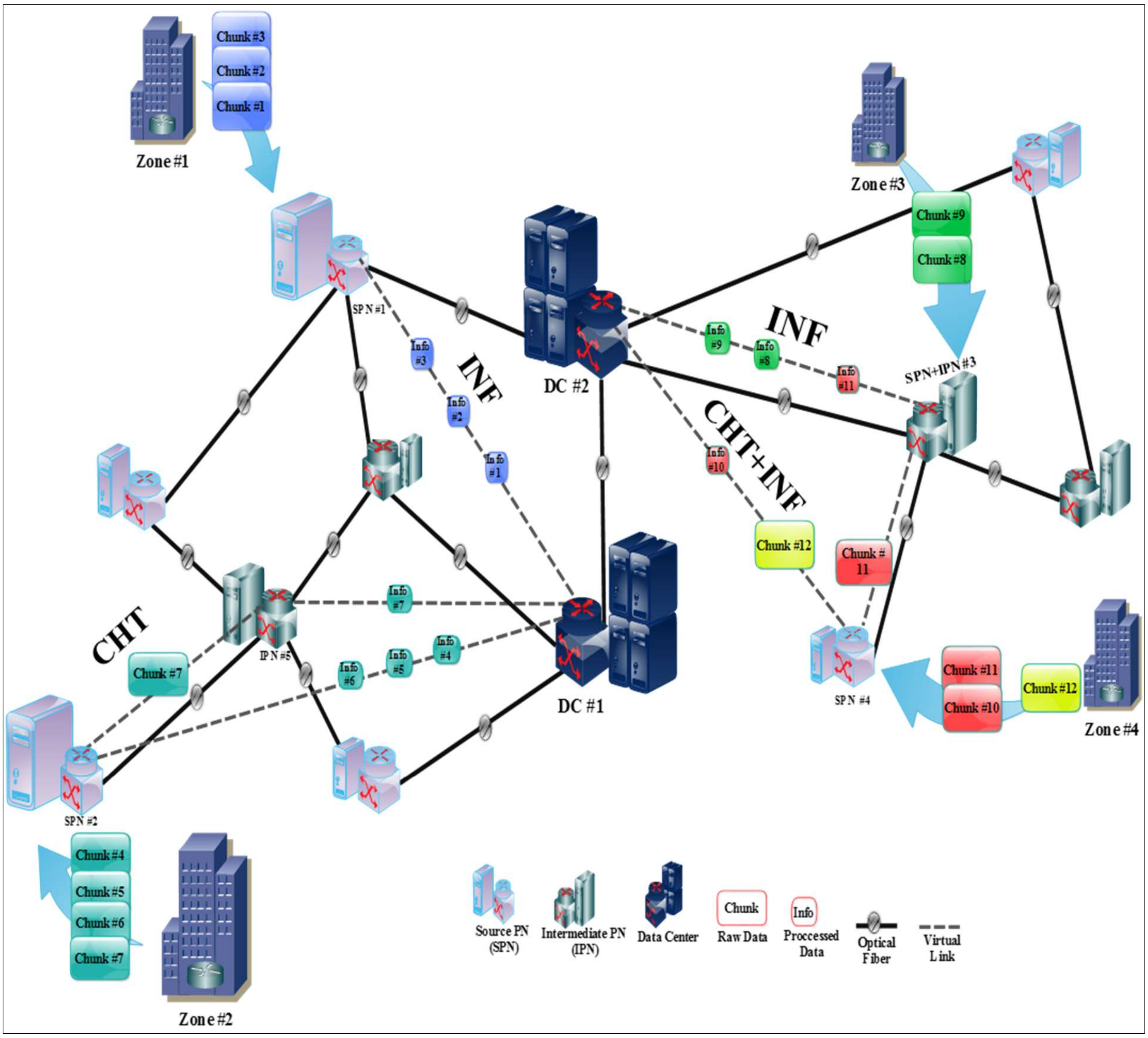


Figure 3-7: EEBDN: An illustrative example.

The PN connected to a certain zone is referred to as a source PN (SPN) as it is the first PN in which *Chunk*s are received from its corresponding zone and locally or centrally processed. Those PNs that receive *Chunk*s from other SPNs are called intermediate PNs (IPNs). An IPN, with respect to a given SPN, might itself be an SPN that implements local processing for its corresponding zone.

This means that a PN can perform both the roles of SPN and IPN if needed. The unprocessed *Chunk* traffic from SPNs to IPNs or to DCs is called *Chunk* Big Data Traffic (CHT).

After processing the *Chunk*s either in SPNs, IPNs or in the DCs, knowledge is extracted in the form of smaller rate traffic that we called the *Info* Big Data Traffic (INF). INF propagates from PNs (SPN or IPNs) toward DCs through the core network. Note that DCs have the special property that both the locally generated INF and the remotely received INF from other PNs do not flow outside the DC. As mentioned before, a PN is built at a certain core node; therefore, the PN ID is the same as the core node ID at which it is installed. This also applies to the DC-ID. Each zone in Figure 3-7 represents a probable scenario that our approach can optimise as follows: Zone 1: The SPN #1 of zone 1 is capable of processing all incoming Chunks (Chunks #1, #2, #3) and all the output (Infos #1, #2, #3) are optimally aggregated to DC #1. This scenario generates only INF in the network from SPNs to DCs.

Zone 2: The SPN #2 of zone 2 can process *Chunks* #4, #5 and #6. *Chunk* #7 is, however, transported as a **CHT** to an optimal IPN (IPN #5) as one or more of the resources (CPU, storage, internal switches and routers) of SPN #2 are fully utilised. After *Chunk* #7 is forwarded to IPN #5, it is processed there and the output (*Info* #7) is aggregated as an INF through an energy efficient route to DC #1.

Zone 3: The SPN #3 of zone 3 processes its own data (*Chunks* #8 and #9) and also acts as an IPN to process other incoming *Chunks* (*Chunk* #11 from SPN #4) when it is not being fully utilised. The movement of *Infos* from this PN represents the INF.

Zone 4: The SPN #4 of zone 4 has the smallest processing and storage space, thus it processes the smallest number of *Chunk*s (*Chunk* #10) and forwards any extra *Chunk*s to the next optimal PN or

DC. For instance, *Chunk* #11 is forwarded to IPN #3. However, when all other PNs deplete their processing resources, then any extra unprocessed *Chunk*s by SPN #4 (i.e., *Chunk* #12) are uploaded directly from SPN #4 to be processed by an optimally selected DC (DC #2 in Figure 3-7). For such an event, CHT starts to dominate the network traffic from SPNs to DCs.

3.6 Summary

This chapter presented the main sources and characteristics of big data as well as an introduction to Hadoop-MapReduce, which is the main storage and processing framework of big data. It also showed the main challenges facing the classical big data networks in terms of processing and networking. Then, it gave a comprehensive literature review that summarised the work done previously in the literature on big data processing and networking with focusing on minimising big data communication cost and power consumption. Finally, it gave a brief description of the classical big data networks with its drawback followed by the proposed energy efficient big data networks and its main benefits and promises with an illustrative example.

Chapter 4 Energy Efficient Big Data Networks: The Impact of Volume and Variety

In this chapter, we studied the impact of big data's volume and variety dimensions on the EEBDN by developing a Mixed Integer Linear Programming (MILP) model to encapsulate the distinctive features of these two dimensions. Firstly, a progressive energy efficient edge, intermediate, and central processing technique is proposed to process big data's raw traffic by building processing nodes (PNs) in the network along the way from the sources to datacentres. Secondly, we validated the MILP operation by developing a heuristic that mimics, in real time, the behaviour of the MILP for the volume dimension. Thirdly, we determined the energy efficiency limits of the EEBDN approach under several conditions where PNs are less energy efficient in terms of processing and communication compared to data centres. Fourthly, we evaluated the performance limits in the EEBDN approach by studying a "software-matching" problem where different software packages are required to process big data.

The results are then compared to the Classical Big Data Networks (CBDN) approach where big data is only processed inside centralised data centres. The results also identify the limits of the progressive processing approach and in particular the conditions under which the CBDN centralised approach is more appropriate given certain PNs energy efficiency and software availability levels.

4.1 Energy Efficient Big Data Networks: Impact of Volume

The remarkable evolution of Internet-enabled technologies is driving the world to be inundated by a colossal amount of data generated from various domains, such as bioinformatics, health informatics, social media, text, log files, sensors data, video streaming, purchase transaction records and more. The term big data has been devised to describe the handling of the enormous number of data types generated by numerous data sources. As mentioned earlier, the first challenge facing the DCs is the enormous volume of data fluxing to them. Although it is currently extremely hard to determine the volume of data generated by a significant number of Internet-enabled devices, the situation is going to be more complicated in the near future, as the projected number of Internet-connected devices is anticipated to reach 100 billion devices by 2020 [106]. Based on the International Data Corporation (IDC) report [107], the overall envisaged data volume will reach 40,000 Exabytes in 2020. Facebook and Twitter create more than 18 Terabytes (TB) every single day [8]. More than 210 billion emails are sent every day [108]. The size of the climate change data repositories is projected to grow to nearly 350 Petabytes (PBs) by 2030 [109]. Five PBs is equivalent to the total number of letters delivered by the US Postal Service in one year [110]. This exponential increase in the speed of generating data, in the volumes of data and in the variety of big data sources comes in parallel with drops in the percentage of data processed inside datacentres (DC) because of insufficient and inefficient analysis tools [8], recall Figure 3-3. Accordingly, a large amount of the data to be processed is neglected, deleted or delayed. Thus, immense networking power is consumed due to transferring unprocessed data from its sources to DCs, while the only interest is in the small volume of knowledge it carries. Furthermore, extra wastage in storage and bandwidth can result from transferring raw data, which leads to a magnification of the financial costs. High-speed processing of such immense data volumes as produced by plentiful data sources calls for new processing and communications methodologies in the big data era.

4.1.1 Problem statement

In this section, we focused on the big data large volume and variety and address the problem of power minimisation in networks that support big data. Despite the large volume associated with big data, the real interest in most cases is in the knowledge derived after processing the data and not in the data itself as discussed earlier. Therefore, in this work, we addressed four main problems. Firstly, where to process big data to minimise power consumption given limited processing capacity at the source and intermediate nodes but large processing capability at central data centres. Secondly, we addressed the problem of how to optimally, from an energy efficiency point of view, deal with big data *Chunk*s that have variable processing requirements. Thirdly, we considered the problem of jointly optimising communication and processing power consumption in big data networks when the processing equipment power consumption increases for the same task. This increase can happen when variable size and sophistication equipment is used. Here we evaluated the impact on the choice of optimal location to process content, given that the optimal location of where to process is dictated by the interplay between communications and processing power consumption. Fourthly, we focused on the problem of how to optimally deal with a software matching problem where some nodes may not have the full software library needed to process different big data applications.

4.1.2 Volume MILP model

In this section, we introduce a MILP model for the EEBDN by taking into consideration the bypass approach in an IP over WDM network. For details of energy efficient MILP optimisation in DCs and IP over WDM network architectures see [4, 17, 42, 43, 47, 111-113]. We placed capacitated PNs at each core node of the NSFNET, as depicted in Figure 4-1, with DCs with large enough capacities. The NSFNET network consists of 14 nodes connected by 21 bidirectional links [42]. The DCs are used to process all extra big data *Chunks* originated by other PNs and to receive the results of the processed *Chunks* (i.e., *Infos*) produced by the PNs to store them for further use.

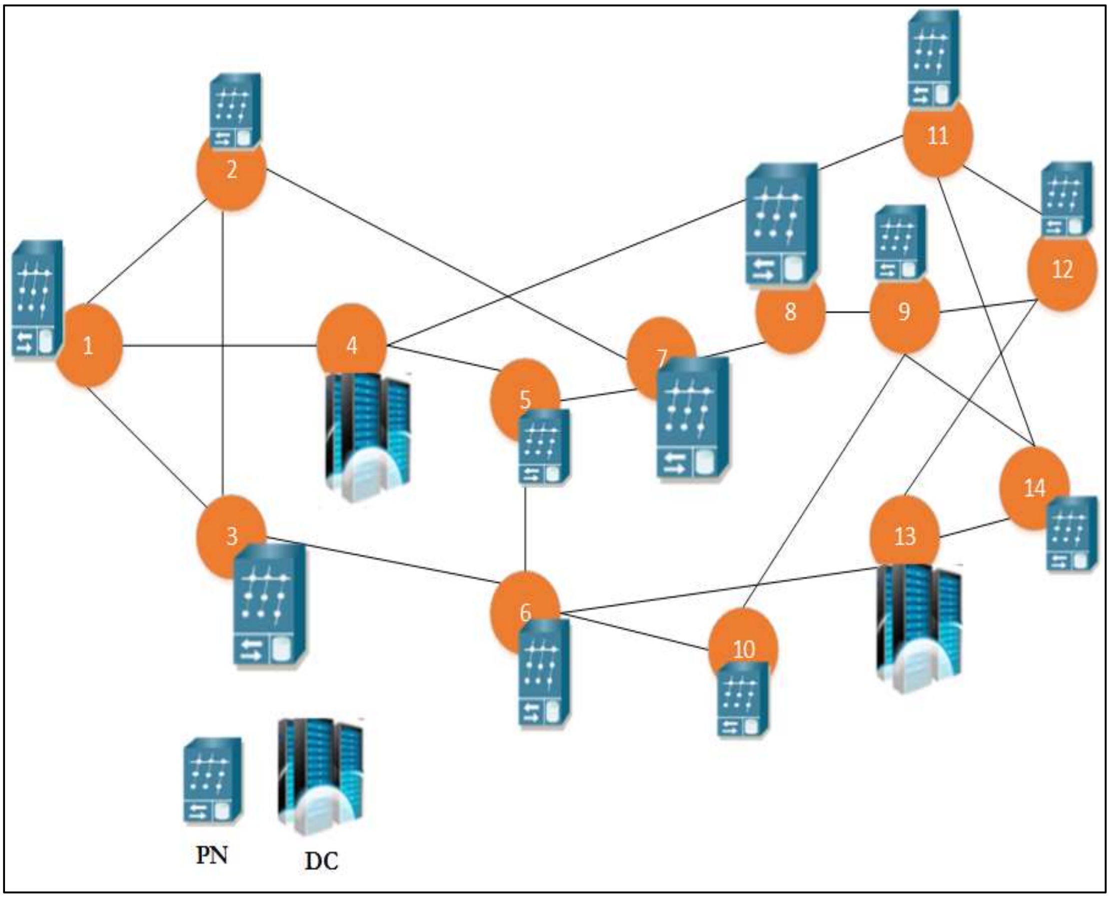


Figure 4-1. NSFNET network with PNs

The IP over WDM power consumption comprises the power consumption of the router ports, transponders, EDFAs, regenerators and optical switches. On the other hand, the power consumptions of the PNs and the DCs are composed of the power consumption of the servers, storage, and the internal LAN switches and routers. We assume that the power consumption of routers and switches is proportional to the offered load. Note that, in addition to the existence of these big data *Chunks* and *Infos* in the network, we assumed, for realistic considerations, that there is additional traffic between core nodes, which is referred to as regular traffic. This traffic represents any data that is not intended for big data analytics [47].

Main Features of the MILP Model:

From the introduction and the example in Chapter 3, Section 3.5.1, we summarise the main features of our model as follows:

- The model optimises the location where the processing of each *Chunk* is carried out, in stages by starting the processing locally at the SPNs, moving through the IPNs, and finally stepping out towards the data centres.
- The model optimises the destination of *Infos* so that each *Info* goes to one DC that is optimally located.
- The model performs a consolidation process to serve as many *Chunks* as possible in the same server, by ensuring that the total CPU utilisation in one server allocated to process one or more *Chunk*s does not exceed the server CPU capacity. Accordingly, the number of *Chunk*s per server is optimised.
- The model ensures that all servers usage of a given PN does not exceed the processing capacity of that node. For blocking avoidance, the internal LAN switches and routers capacity are assumed to be large enough. Note that the traffic is routed over the network since the model ensures that the flow conservation requirements of big data and regular traffic are satisfied at all networking levels (i.e. IP and optical layers of the IP over WDM network). In the heuristic, we employed a minimum hop routing algorithm. Scheduling, capacity consideration, and routing the intra data centres traffic are left for future work.
- The model optimises DCs locations which are used for processing the incoming *Chunks* when all PNs are fully utilised and for receiving the *Infos* from PNs to store them for further analysis.

- Assessing the energy efficiency limits of PNs in the EEBDN when less energy efficient (compared to DCs) networking and computing equipment is used inside PNs .
- Analysing the software matching problem and its effect on EEBDN performance where *Chunk*s are associated with the correct PNs hosting the appropriate software package that can process that *Chunk*.
- Table 4-1 defines the parameters and variables used in the EEBDN MILP model:

Notation	**Description**
$s\ and\ d$	Denote source and destination points of regular traffic demand between a node pair.
$i\ and\ j$	Denote end points of a virtual link in the IP layer.
$m\ and\ n$	Denote end points of a physical fibre link in the optical layer.
R_{sd}	The NSFNET regular traffic demand from node s to node d (Gbps).
N	Set of IP over WDM nodes.
N_i	The set of neighbour nodes of node i in the optical layer.
NS_p	Number of servers at the PN p.
SW_{sc}	The CPU workload of the server required to process *Chunk c* generated at source node s (GHz).
MSW	Maximum server workload (GHz).
MP_p	Maximum workload node p. $MP_p = NS_p \cdot MSW$ (GHz).
MSR_p	Maximum internal switches and router capacity of the PN p (Gbps).
MS_p	Maximum storage of node p (Gb).
NCH	Total number of *Chunks* in one node per second.
CH_s	Set of *Chunks* in a source node s.

CHV_{sc}	The volume of *Chunk c* generated at source node *s* (Gb).
PRR_{sc}	Processing reduction ratio for *Chunk c* generated by node *s* (unitless).
WL	Number of wavelengths in a fibre.
B	Wavelength bit rate (Gbps).
S	Maximum distance between neighbouring EDFAs (km).
PR	Power consumption of a router port (W).
PTR	Power consumption of a transponder (W).
PO_i	Power consumption of optical switch installed at node $i \in N$ (W).
PE	Power consumption of EDFA (W).
PRG	Power consumption of a regenerator (W).
D_{mn}	Distance between node pair (m, n) (km).
A_{mn}	Number of EDFAs on physical link (m, n). Typically, $A_{mn} = \left\lfloor \frac{D_{mn}}{S} - 1 \right\rfloor + 2$ [42].
RG_{mn}	Number of regenerators on physical link (m, n).
PUN	Power usage effectiveness of IP over WDM networks (unitless). PUN is defined as the ratio of the power drawn from the electric source to the power used by the equipment (networking in this case). PUN accounts for cooling, lighting and related power consumption.
PU	Power usage effectiveness of the PNs and DCs (unitless).
SMP	Server maximum power consumption (W).
SEB	PNs' and DCs' switch energy per bit (W/Gbps).
REB	PNs' and DCs' router energy per bit (W/Gbps).

RS	Internal PNs' and DCs' switches redundancy.
RR	Internal PNs' and DCs' routers redundancy.
RSG	PNs and DCs storage redundancy.
PSG	PNs' and DCs' storage power per Gigabit (W/Gb).
δ	Server power per GHz, $\delta = (SMP - PIDLE) / MSW$ (W/GHz). GHz is used to specify the capability of a processor and the number of processors a job needs.
DCN	Number of location optimised DCs.

Table 4-1: List of parameters and their definitions.

Table 4-2 defines the variables used in the EEBDN MILP model:

Notation	Description
CHT_{sp}	Big data *Chunks* traffic generated at SPN s and directed to destination node p (p could be SPN, IPN or DC) (Gbps).
INF_{pd}	Aggregated big data *Info* traffic from PN p to DC d. Node p could be SPN or IPN only (Gbps).
C_{ij}	Number of wavelength channels in the virtual link (i, j).
R_{ij}^{sd}	Traffic flow of the regular traffic R_{sd} between node pair (s, d) traversing virtual link (i, j).
W_{mn}^{ij}	Number of wavelength channels in the virtual link (i, j) traversing physical link (m, n).
W_{mn}	Number of wavelength channels in the physical link (m,n).
CHT_{ij}^{sp}	Traffic flow of the big data *Chunks* traffic CHT_{sp} between node pair (s, p) traversing virtual link (i, j).

INF_{ij}^{pd}	Traffic flow of the big data *Info* traffic INF_{pd} between node pair (*p, d*) traversing virtual link (*i, j*).
AR_i	Number of aggregation ports in router *i* utilised by regular traffic R_{sd}
ACH_i	Number of aggregation ports in router *i* used in big data *Chunks* traffic CHT_{sp}.
AI_i	Number of aggregation ports in router *i* utilised by big data *Info* traffic INF_{pd}.
F_{mn}	Number of fibers in physical link (*m,n*).
PNW_p	Total PN *p* workload (GHz).
Y_{spc}	Y_{spc} = 1 if *Chunk c* is generated at SPN *s* and processed in PN *p*, else Y_{spc} = 0.
SCH_p	Amount of big data *Chunks* stored in PN *p* (Gb).
DC_d	DC_d = 1 if a DC is built at core node *d*, else DC_d = 0.

Table 4-2: List of variables and their definitions.

Under the bypass approach, the total IP over WDM network power consumption is composed of the following components

1) The power consumption of router ports

$$\sum_{i \in N} PR \cdot (AR_i + ACH_i + AI_i) + PR \cdot \sum_{j \in N:\, i \neq j} (C_{ij}). \tag{4-1}$$

2) The power consumption of transponders

$$\sum_{m \in N} \sum_{n \in N_m} PTR \cdot W_{mn}. \tag{4-2}$$

3) The power consumption of regenerators is

$$\sum_{m \in N} \sum_{n \in N_m} PRG \cdot W_{mn} \cdot RG_{mn}. \tag{4-3}$$

4) The power consumption of EDFAs

$$\sum_{m\in N}\sum_{n\in N_m} PE\cdot A_{mn}\cdot F_{mn}. \tag{4-4}$$

5) The power consumption of optical switches

$$\sum_{i\in N} PO_i. \tag{4-5}$$

Equation (4-1) evaluates the total power consumption of the router ports for all the types of traffic, which are the regular traffic R_{sd}, big data *Chunks* traffic CHT_{sp}, and big data *Info* traffic INF_{pd}. It computes the total power consumption of the ports aggregating data traffic and the ports connected to optical nodes. Equations (4-2) and (4-3) evaluate the power consumption of all the transponders and regenerators in the optical layer. Equation (4-4) evaluates the total power consumption of the EDFAs in the optical layer. Equation (4-5)) evaluates the total power consumption of the optical switches. The power consumption of the PNs and DCs is composed of the following sections:

1) The power consumption of internal PNs and DCs switches and routers

$$\begin{aligned} PSR = & \sum_{p\in N}\sum_{s\in N} CHT_{sp}\cdot\ (RS\cdot SEB + RR\cdot REB) \\ & + \sum_{p\in N}\sum_{d\in N}\left(CHT_{pd} + INF_{pd}\right)\cdot(RS\cdot SEB + RR\cdot REB) \\ & + \sum_{p\in N}\sum_{d\in N} INF_{pd}\cdot(RS\cdot SEB + RR\cdot REB) \end{aligned} \tag{4-6}$$

Equation (4-6) evaluates the total power consumption of the internal switches and routers in the PNs and DCs. This is done by multiplying the incoming and outgoing big data traffic by the switches' and routers' energy per bit. We performed the analysis by considering a network architecture where $RS = RR = 1$.

2) The power consumption of the storage

$$\sum_{p \in N} SCH_p \cdot RSG \cdot PSG. \tag{4-7}$$

3) The power consumption of all servers inside PNs and DCs

$$\sum_{p \in N} \delta \cdot PNW_p + NS_p \cdot PIDLE. \tag{4-8}$$

Equation (4-7) represents the storage power consumption of node p. We performed the analysis by considering a network architecture where $RSG = 1$. Equation (4-8) represents the PNs and DCs power consumption. Note that server power consumption is a function of the idle power, maximum power and CPU utilisation [114]. Therefore, the power consumption of all servers inside the PNs and DCs is calculated using equation (4-8). The model is defined as follows:

Objective: Minimise

$$\begin{aligned}
PUN \cdot \Bigg(& \sum_{i \in N} PR \cdot (AR_i + ACH_i + AI_i) + PR \cdot \sum_{j \in N: i \neq j} (C_{ij}) \\
& + \sum_{m \in N} \sum_{n \in N_m} PTR \cdot W_{mn} + \sum_{m \in N} \sum_{n \in N_m} PRG \cdot W_{mn} \cdot RG_{mn} + \sum_{m \in N} \sum_{n \in N_m} PE \\
& \cdot A_{mn} \cdot F_{mn} + \sum_{i \in N} EO_i \Bigg) \\
+ PU \cdot \Bigg(& \sum_{p \in N} \delta \cdot PNW_p. + NS_p \cdot PIDLE + \sum_{p \in N} \sum_{s \in N} CHT_{sp} \cdot (RS \cdot SEB + RR \cdot REB) \\
& + \sum_{p \in N} \sum_{d \in N} \left(CHT_{pd} + INT_{pd}\right) \cdot (RS \cdot SEB + RR \cdot REB) \\
& + \sum_{p \in N} \sum_{d \in N} INF_{pd} \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} SCH_p \cdot RSG \cdot PSG \Bigg).
\end{aligned} \tag{4-9}$$

Equation (4-9) gives the model objective, which is to minimise the IP over WDM network power consumption as well as the PNs' and DCs' power consumption.

Subject to:

PNs and DCs Constraints:

1) Processing counter of big data *Chunks* constraint

$$\sum_{p \in N} Y_{spc} = 1, \qquad (4\text{-}10)$$

$$\forall s \in N, \forall c \in CH_s.$$

Constraint (4-10) ensures that a *Chunk c* generated by PN *s* is processed by no more than one PN *p*. However, our model performs slicing, i.e., multiple servers could process a given *Chunk* in a PN as long as these servers belong to that PN.

2) Big data *Chunks* traffic constraint

$$CHT_{sp} = \sum_{c \in CH_s} CHV_{sc} \cdot Y_{spc}, \qquad (4\text{-}11)$$

$$\forall s, p \in N.$$

Constraint (4-11) calculates the big data *Chunks* traffic generated at source node *s* and directed to node *p*. This traffic is generated by transmitting CHV_{sc} from node *s* to node *p* in one second.

3) Aggregated processed big data traffic constraint

$$\sum_{d \in N} INF_{pd} = \sum_{s \in N} \sum_{c \in CH_s} CHV_{sc} \cdot Y_{spc} \cdot PRR_{sc}, \qquad (4\text{-}12)$$

$$\forall p \in N.$$

Constraint (4-12) represents the aggregated big data *Info* traffic INF_{pd} generated by PN *p* and destined to DC *d*. The big data *Info* traffic is obtained by multiplying the *Chunks* (CHV_{sc}) allocated to the PN *p* by the PRR_{sc}.

4) Number and locations of DCs constraints

$$\sum_{p\in N} INF_{pd} \geq DC_d, \tag{4-13}$$

$$\forall d \in N,$$

$$\sum_{p\in N} INF_{pd} \leq Z \cdot DC_d, \tag{4-14}$$

$$\forall d \in N,$$

$$DCN = \sum_{d\in N} DC_d. \tag{4-15}$$

Constraints (4-13) and (4-14) build a DC in location *d* if that location is selected to store the results of the processed big data traffic (i.e., *Infos*) or selected to process the incoming big data *Chunks* from PNs, where *Z* is a large enough unitless number to ensure that $DC_d = 1$ when $\sum_{p\in N} INF_{pd}$ is greater than zero. Constraint (4-15) limits the total number of built DCs to DCN.

5) PNs and DCs workload and processing capacity constraints

$$PNW_p = \sum_{s\in N} \sum_{c\in CH_s} SW_{sc} \cdot Y_{spc}, \tag{4-16}$$

$$\forall p \in N \text{ and}$$

$$PNW_p \leq NS_p \cdot MSW + \left(M \cdot DC_p\right), \tag{4-17}$$

$$\forall p \in N.$$

Constraint (4-16) represents the total workload at PN *p*, which is the summation of the CPU workload of all the servers in that PN. Constraint (4-17) ensures that the total workload of PN *p* does not exceed the maximum workload assigned to this PN, *M* is a large enough unitless number. However, the workload capacity is large enough if a DC is built at core node *p*. Note that, the model implements a consolidation process by processing as many *Chunks* as possible within the same server to minimise the network power consumption and number of active servers.

6) PNs and DCs storage constraints

$$SCH_p = \sum_{s \in N} \sum_{c \in CH_s} CHV_{sc} \cdot Y_{spc} \,, \tag{4-18}$$

$\forall p \in N$ and

$$SCH_p \leq MS_p + \left(H \cdot DC_p\right), \tag{4-19}$$

$\forall p \in N.$

Constraint (4-18) represents the size of *Chunks* in Gb stored in PN p. Constraint (4-19) ensures that the total data stored in PN p does not exceed the storage capacity of that PN. *H* is a large enough unitless number to guarantee that there is no storage capacity limitation at the DCs.

7) PNs and DCs internal switches and routers constraints

$$\sum_{s \in N} CHT_{sp} \leq MSR_p + \left(A \cdot DC_p\right), \tag{4-20}$$

$\forall p \in N.$

Constraint (4-20) ensures that the total amount of big data traffic between the PNs does not exceed the maximum switching and routing capacity of the internal switches and routers in those PNs. On the other hand, the capacity of the DCs' switches and routers is unlimited, where *A* is a large enough unitless number to guarantee that there is no capacity limitation at the DCs. To avoid blocking of big data *Chunks*, we assumed that the internal switches and routers capacity of the PNs is also large enough.

The IP over WDM Network Constraints:

1) Flow conservation constraints for the regular traffic

$$\sum_{j\in N:\, i\neq j} R_{ij}^{sd} - \sum_{j\in N:\, i\neq j} R_{ji}^{sd} = \begin{cases} R_{sd} & i = s \\ -R_{sd} & i = d \\ 0 & otherwise \end{cases} \tag{4-21}$$

$$\forall s, d, i \in N: s \neq d.$$

2) Flow conservation constraints for the big data *Chunks* traffic

$$\sum_{j\in N: i\neq j} CHT_{ij}^{sp} - \sum_{j\in N:\, i\neq j} CHT_{ji}^{sp} = \begin{cases} CHT_{sp} & i = s \\ -CHT_{sp} & i = p \\ 0 & otherwise \end{cases} \tag{4-22}$$

$$\forall s, p, i \in N: s \neq p.$$

3) Flow conservation constraints for the big data *Info* traffic

$$\sum_{j\in N: i\neq j} INF_{ij}^{pd} - \sum_{j\in N: i\neq j} INF_{ji}^{pd} = \begin{cases} INF_{pd} & m = p \\ -INF_{pd} & m = d \\ 0 & otherwise \end{cases} \tag{4-23}$$

$$\forall p, i \in N, \forall d \in N: p \neq d.$$

Constraints (4-21)-(4-23) represent the flow conservation constraints for the regular traffic R_{sd}, big data *Chunks* traffic CHT_{sp} and big data *Info* traffic INF_{pd}, in the IP layer. These constraints ensure that the total outgoing traffic should be equal to the total incoming traffic, except for the source and destination nodes. It can also ensure that the flow can be divided into multiple flow paths in the IP layer.

4) Virtual link capacity constraint

$$\left(\sum_{s\in N} \sum_{d\in N:\, s\neq d} R_{ij}^{sd} + \sum_{s\in N} \sum_{p\in N:\, s\neq p} CHT\,_{ij}^{sp} + \sum_{p\in N} \sum_{d\in N:\, p\neq d} INF_{ij}^{pd} \right) \leq C_{ij}.B \tag{4-24}$$

$$\forall i, j \in N: i \neq j.$$

Constraint (4-24) ensures that the summation of all traffic flows through a virtual link does not exceed its capacity.

5) Optical layer flow conservation constraints:

$$\sum_{n \in N_m} W_{mn}^{ij} - \sum_{n \in N_m} W_{mn}^{ij} = \begin{cases} C_{ij} & m = i \\ -C_{ij} & m = j \\ 0 & otherwise \end{cases} \tag{4-25}$$

$$\forall i, j, m \in N: i \neq j.$$

Constraint (4-25) represents the flow conservation constraints in the optical layer. It ensures that the total outgoing wavelengths in a virtual link should be equal the total incoming wavelengths, except for the source and the destination nodes of the virtual link. It is assumed that wavelength conversion is available in the model to enable better utilisation of bandwidth and reduce blocking probabilities.

6) Physical link capacity constraints

$$\sum_{i \in N} \sum_{j \in N: i \neq j} W_{mn}^{ij} \leq WL \cdot F_{mn}. \tag{4-26}$$

$$\forall m \in N, n \in N_m.$$

Constraint (4-26) ensures that the summation of the wavelengths in a virtual link traversing a physical link does not exceed the capacity of the fibres in the physical link.

7) Wavelengths capacity constraint

$$\sum_{i \in N} \sum_{j \in N: i \neq j} W_{mn}^{ij} = W_{mn} \tag{4-27}$$

$$\forall m \in N, n \in N_m.$$

Constraint (4-27) ensures that the summation of the wavelengths traversing a physical link do not exceed the total number of wavelengths in that link.

8) Number of aggregation ports utilised by regular traffic constraint

$$AR_i = \frac{1}{B} \cdot \sum_{d \in N: i \neq d} R_{id} \tag{4-28}$$

$$\forall i \in N.$$

9) Number of aggregation ports utilised by **CHT** traffic constraint

$$ACH_i = \frac{1}{B} \cdot \sum_{p \in N:\, i \neq p} CHT_{ip} \qquad (4\text{-}29)$$

$$\forall i \in N.$$

10) Number of aggregation ports utilised by **INF** traffic constraint

$$AI_i = \frac{1}{B} \cdot \sum_{d \in N:\, i \neq p} INF_{id} \qquad (4\text{-}30)$$

$$\forall i \in N.$$

Constraints (4-28)-(4-30) calculate the number of aggregation ports for each router that serves the regular traffic R_{sd}, big data *Chunks* traffic CHT_{sp} and big data *Info* traffic INF_{pd}.

The MILP in Section 4.1.2 is used to evaluate the proposed EEBDN. In addition, the same model can be used to evaluate the CBDN approach by introducing a constraint that prevents the processing of big data outside the DCs. The classical model is characterised by:

- Each node of the NSFNET generates a similar number of *Chunks* as in the scenarios of the EEBDN model.
- The classical model optimally selects DC locations to host, store and process *Chunks* that are forwarded by all the nodes through energy efficient routes.
- The extracted knowledge from the *Chunks* (i.e. *Infos*) is stored in the same DC that processed the corresponding *Chunks* for further analysis.

- The DCs process the incoming *Chunks* with minimum power consumption by utilising the lowest number of servers that can handle those *Chunks*. This is emphasised by calculating the processing power consumption required by the minimum number of servers.
- The processing of big data is carried out inside the DCs only when there are no PNs in the network. The objective of the model is to minimise the Network Power Consumption (NPC) and the DC power consumption.

4.1.3 Volume heuristic

In this section, we validated the MILP operation by developing a heuristic that mimics, in real time, the behaviour of the MILP. Having obtained results from the MILP we developed insight into what minimises power consumption in our proposed progressive processing big data networks. We observed from the results that the MILP attempts to process all the data in the source node if the source node has enough capacity, which reduces the communication transmission power consumption needed otherwise to reach remote processing nodes. If the source processing node does not have enough capacity then the *Chunk*s are transmitted to the processing node at minimum hop distance and when such intermediate nodes are depleted of processing capability, the minimum hop data centre is used. Routing in the network was observed to follow minimum hop routing. None of these rules were written in the MILP. The MILP was only required to minimise the total power consumption (network and processing). We therefore used these insights to construct our heuristic, which therefore mimics the MILP behaviour. The heuristic uses simple rules as described above and hence can run fast unlike the MILP. Therefore, the heuristic can be used to provide real time control and routing in the network.

The heuristic is used for two main purposes. Firstly, as a verification of the MILP results. Secondly, since the heuristic uses simple rules, it runs fast unlike the MILP. Therefore, it can enable network control (which *Chunk* to process where for example) and routing which can both be performed in real time through the use of the heuristic. The second objective (real time control of the network) is fully achieved by our heuristic. The first objective (verification of MILP) is partially achieved. The heuristic uses the optimum data centre node locations (nodes 4 and 13) obtained from the MILP. The heuristic is otherwise independent of the MILP. The flowchart of Figure 4-2 shows the heuristic, which aims to process the incoming *Chunks* by utilising the minimum number of resources so that minimal power is consumed. The heuristic is initialised by defining the physical network topology, in this case the NSFNET, with 14 nodes and 21 links. 12 PNs are distributed in the network and 2 DCs are located at nodes 4 and 13. Note that between each node pair there exists a regular traffic demand R_{sd} in the network.

Each node receives a number of *Chunks* (β) from its corresponding zone. Each *Chunk* is characterised by a volume and CPU workload requirements. The heuristic starts at the edge processing stage by selecting an SPN, then picks a *Chunk* from this SPN to read its CPU requirement. The heuristic checks the processing capacity of that SPN and the *Chunk* is processed locally in the current SPN in case there are enough processing resources. This approach guarantees the implementation of as much edge processing as possible. The heuristic repeats this process for all SPNs. Note that changing the order of SPN selection does not change the results as each SPN can be totally packed with processing jobs and in this case all processing tasks have the same CPU requirement. Once a *Chunk* is processed locally in an SPN, a corresponding INF demand is calculated between that SPN and the nearest DC following a minimum hop path. In case all *Chunks* are processed locally by SPNs, the only demands in the network are therefore the INF and regular demands. Those demands are routed and the NPC is calculated using the algorithm developed in [42].

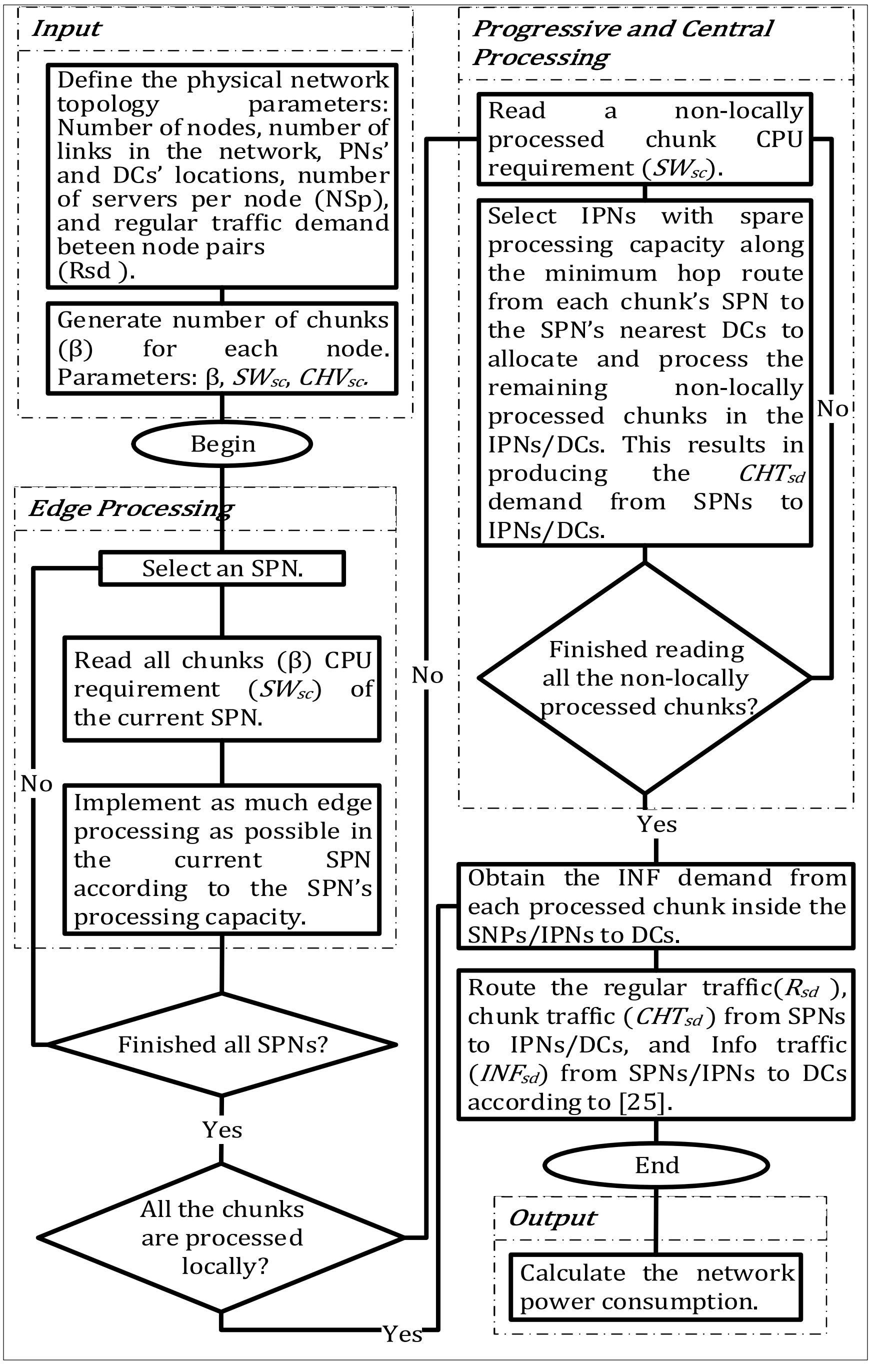


Figure 4-2: EEBDN: volume heuristic.

However, the progressive processing stage inside the IPNs and the central processing stage inside the DCs are started when the SPNs are not capable of implementing full edge processing, (i.e. not all *Chunks* are processed locally in the SPNs). This is done by forwarding the remaining non-locally processed *Chunks* from all SPNs along the minimum hop path to the nearest IPNs/DCs. An IPN is selected if there is spare processing capacity. This results in the CHT traffic demands between the SPNs and IPNs/DCs. The heuristic then obtains the INF demands resulting from processing the non-locally processed *Chunks* in IPNs. Therefore, in this case the network has three traffic demands: INF_{sd} from partial local processing in SPNs, INF_{sd} from progressive processing in IPN and the CHT_{sd} demands. Again, these types of traffic demands are routed over the network as well as the regular traffic R_{sd} according to the heuristic in [42] and the total NPC is calculated.

4.1.4 Complexity analysis

The proposed EEBDN heuristic aims to work around the NP-hard complexity [115] of the MILP model solved using CPLEX. There are two main processes in the heuristic. Firstly, the bin packing problem where objects (where a number of *Chunk*s per node (β)) of different volumes must be packed into a finite number of bins (servers) each of capacity C in a way that minimises the number of bins used. This is a greedy approximation algorithm where for each *Chunk*, it attempts to place it in the first server that can accommodate this *Chunk*. Thus, it requires $O(\beta \, log \, \beta)$ time [116]. Secondly, the generation of initial set of paths is based on minimum hop routing algorithm, which has a complexity of the order $O(N)$ [117], where *N* is the number of nodes in the network. Thus, the overall complexity is $O(\beta \, log \, \beta) + O(N)$ for the processes of the proposed heuristic.

4.2 Results of Volume Scenarios

Our MILP model and the heuristic were evaluated using the NSFNET network depicted in Figure 4-1. The storage capacity of the PNs was assigned to be large enough. Note that we used processor cycles in GHz as a measure of the total processing capability of a node [118]. Table 4-3 summarises the input parameters. We performed the MILP optimisation using the AMPL/CPLEX software running on a PC with 8 GB RAM and an i5 CPU. The heuristic is implemented using MATLAB on the same PC. A single run for the MILP took around 10 s to finish, while the heuristic took less than 1 s. Note that the computational complexity of the MILP grows fast with network size.

Server CPU capacity *in* GHz (*MSW*)	4 GHz
Max server power consumption (*MSP*)	300 W [16]
Idle server power consumption (*PIDLE*)	200 W
PNs and DCs switch power consumption (*PS*)	3.8 kW [16, 119]
PNs and DCs switch capacity (*CS*)	320 Gbps [16, 119]
Energy per bit of the PNs and DCs switch $(SEB) = PS/CS$	11.875 W/Gbps
PNs and DCs router power consumption (*PR*)	5.1 kW [16, 119]
PNs and DCs router capacity (*CR*)	660 Gbps [16, 119]
Energy per bit of the PNs and DCs router $(REB) = PR/CR$	7.727 W/Gbps
PNs' and DCs' storage power per Gigabit (PSG)	0.008 W/Gb [16]
Router power consumption (*PR*)	825 W [54]
IP over WDM regenerator power consumption (*PRG*)	334 W [54]
IP over WDM transponder power consumption (*PTR*)	167 W [54]
IP over WDM optical switch power consumption (PO_i) $\forall i \in N$	85 W [54]

IP over WDM EDFA power consumption (*PE*)	55 W [54]
Wavelength bit rate (*B*)	40 Gbps
Distance between EDFAs (*S*)	80 km
Number of wavelengths per fiber (*WL*)	32
Number of location optimised DCs (*DCN*)	2
IP over WDM power usage effectiveness (*PUN*)	1.5 [16]
PNs and DCs power usage effectiveness (*PU*)	2.5 [16]

Table 4-3: Input data for volume MILP model.

To provide a holistic assessment of the impact of the volume dimension on the EEBDN, we evaluated the proposed progressive processing approach in two volume scenarios.

4.2.1 Scenario #1: Deterministic *Chunk*s volume, PRR = 0.001, number of servers per PN = 5-15 servers

In this scenario, we considered the number of *Chunks* generated per node (β) which vary between 5 and 30. There are two different units used in conjunction with each *Chunk*. Firstly, the size of the *Chunk* which is quoted in Gb and we considered the transmission of each *Chunk* in one second. Therefore, for example, the data rate associated with the transmission of an 80 Gb *Chunk* is 80Gb/s. Secondly, each *Chunk* has a GHz number associated with it which indicates the processing requirement of the *Chunk*. For example, a processor may be able to handle 4 GHz and the *Chunk* may require 1 GHz. Thus, if $\beta = 5$, this means that the total number of *Chunks* to be processed in the network is 70 (since there are 14 nodes in the NSFNET), and it takes one second for the transmission of the given *Chunks* and the corresponding *Infos*. This is a reasonable assumption as we considered the network resources capacity to be enough to handle the *Chunks*. We leave the impact of the capacitated resources on the EEBDN for future work. Note that there is no transmission of *Chunks* and *Infos* in the network when they are handled by the DCs.

We considered the following scenario. The processing capacity of each PN is different and varies between 5 and 15 servers per PN. Each *Chunk* demands 3 GHz of the CPU for processing. The volume of each *Chunk* is 80 Gb and the PRR_{sc} is assumed to be 0.001 for all *Chunks* (i.e., 99.9% reduction). An example of such case is Electrocardiography (ECG) used to detect abnormality during each heartbeat of millions of patients. Table 4-4 summarises the input values needed in this scenario. The results in Figure 4-3-a are based on our MILP optimisation and heuristic and compare our EEBDN power consumption with the NPC of the CBDN approach where big data *Chunks* are sent directly to the DCs for processing.

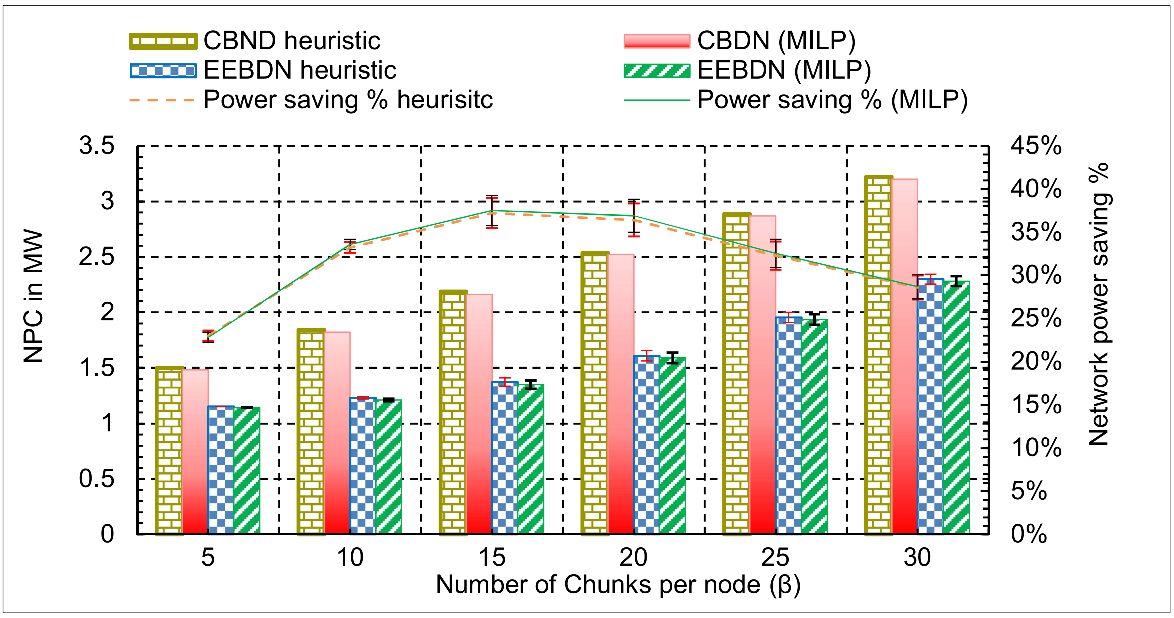
Number of *Chunks* per node (β)	Number of servers per PN (NS_p)	CPU workload per *Chunk* in GHz (SW_{sc})	*Chunk* volume in Gb (CHV_{sc})	PRR_{sc}
5-30	5-15	3 [75]	80 [23]	0.001 [23]

Table 4-4: Volume Scenario #1 parameters.

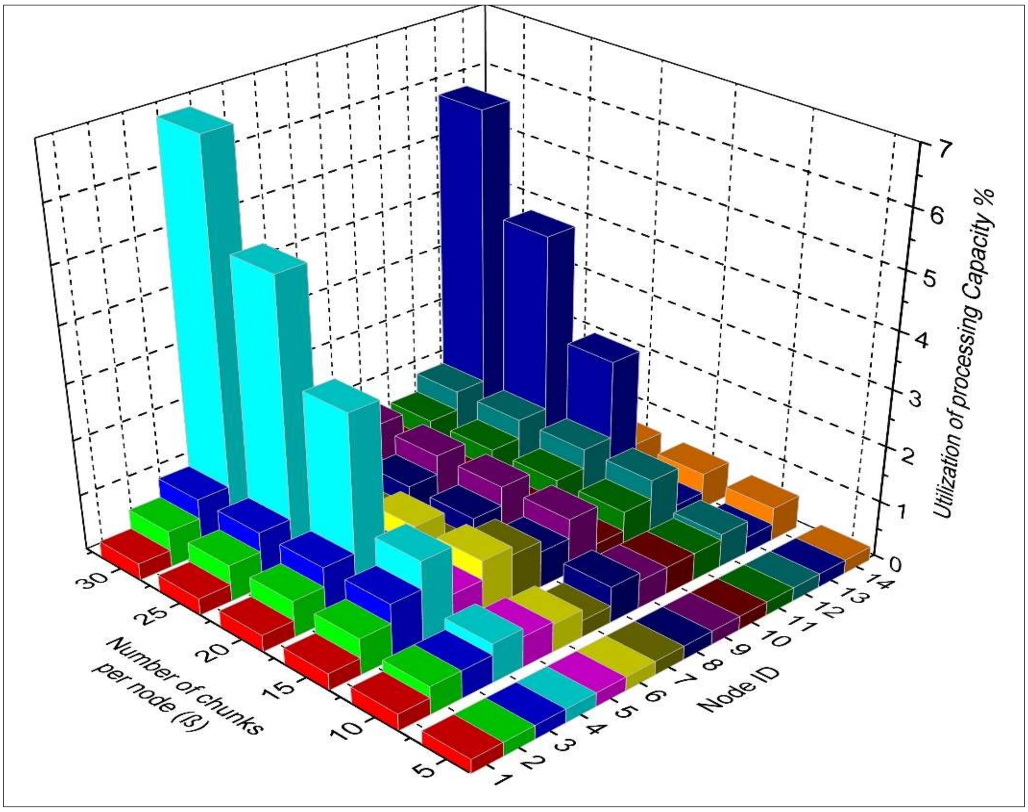
For the MILP, and for all cases, the NPC increased when β increased as more *Chunks* are delivered to the network. Introducing the PNs has, however, greatly bounded the growth in power consumption when the number of *Chunks* increased which leads to network power savings compared to the classical approach in all the cases of the considered values of β due to processing near the source. At $\beta = 5$, the network power saving is smaller than that at $\beta = 15$ since the big data traffic is a small portion of the overall network traffic at these low number of *Chunks* per node ($Network\ traffic = Regular\ traffic + Big\ data\ traffic$). At $\beta = 15$, big data traffic becomes larger due to the large number of *Chunks* generated per node and therefore saving power by processing big data leads to best network power saving at these intermediate levels of big data value. At $\beta = 30$, the big data volume has become so large and dominant that full edge processing (i.e. in the SPNs and IPNs) is not possible given the servers numbers in SPNs and IPNs and therefore the network carries more *Chunks* (unprocessed big data) compared to the case where $\beta = 15$ which has more *Infos*. Note that a maximum network power saving of 38% is achieved at $\beta = 15$, and an average network power saving of 32% is computed considering all β values, compared to the classical approach where no PNs exist in the network for the range of parameters considered. For the heuristic, the same inputs in Table 4-4 are used. The performance of the EEBDN heuristic was compared to the MILP performance in Figure 4-3-a and the heuristic and MILP are in close agreement.

From Figure 4-3-a, the heuristic power savings approach those of the MILP (The MILP power saving is only slightly (i.e. 1.15%) higher than the heuristic's). Moreover, the heuristic can help extend the evaluation by increasing the number of incoming *Chunks* and resources beyond the MILP computational limits. Note that the heuristic for the classical approach is implemented using the same heuristic with an additional condition that prevents processing big data outside the DCs. The results for the EEBDN were repeated 11 times and the graphs showed the average values. The 95% confidence interval [120] is shown as error-bars.

Figure 4-3-b shows the utilisation of the processing capacity as a percentage of the PN processing capacity. At $\beta = 5$ all the SPNs can perform edge processing. When β is between 10 and 15, some PNs with large capacities perform edge processing, as well as processing received *Chunks* from other SPNs that have less processing space, hence PNs here perform progressive processing.



(a)



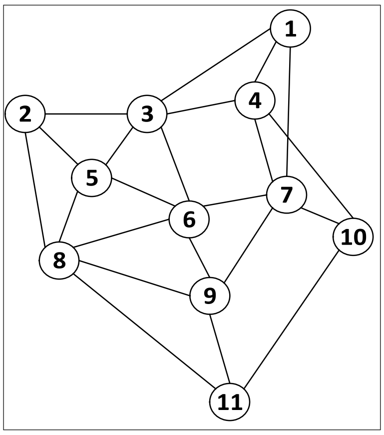
(b)

Figure 4-3 (a) CBDN power consumption vs EEBDN power consumption (MILP and heuristic) for volume Scenario #1. (b) Utilisation of processing capacity % in the EEBDN with different values of β for volume Scenario #1.

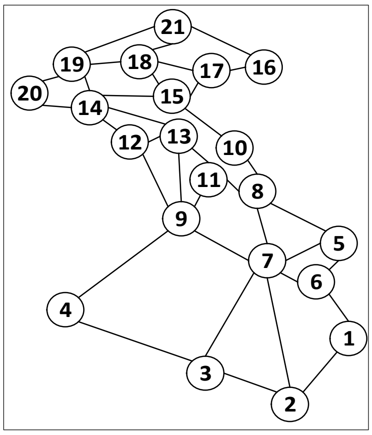
This results in a CHT between SPNs and IPNs, and very small amount of CHT between SPNs and DCs, thereby minimising the DCs processing utilisation. Note that PN #12, which has the capacity to process up to 20 Chunks, is 100% utilised at $\beta = 15$. This is because this node processed its own 15 Chunks and handled an extra five progressed Chunks from other SPNs. At $\beta > 20$, no edge processing inside SPNs and progressive processing inside IPNs is possible since all PNs processing space is depleted and all Chunks are centrally processed inside the DCs. Therefore, the DCs processing utilisation increases dramatically. Note that nodes 4 and 13 have high utilisation as they are the two data centre nodes.

The main goal in this chapter is to show the effectiveness of our progressive processing approach compared to the classical centralised processing approach. We carried out a comparison with the classical (centralised) case, which is the case that is known in the literature and can act as a benchmark. Furthermore, we evaluated the complexity of our heuristic in Section 4.1.3 and therefore provide details relating to complexity / efficiency of our heuristic. The effectiveness of our heuristic was evaluated and it produced results close to the optimum MILP results, for example Figure **4-3**-a.

We re-evaluated the volume scenario of MILP model and heuristic on two more different networks. The COST239 network [121], (see Figure **4-4**-a), which is smaller than the NSFNET and consists of 11 nodes and 25 bidirectional links, and the Italian network [51, 122], (see Figure **4-4**-b) which is bigger than the NSFNET and consists of 21 nodes and 36 bidirectional links. The results in Figure **4-5**-a and Figure **4-5**-b show that the average power savings of the MILP model and the heuristic obtained in the COST239 network are 58% and 56%, respectively for the volume scenario.

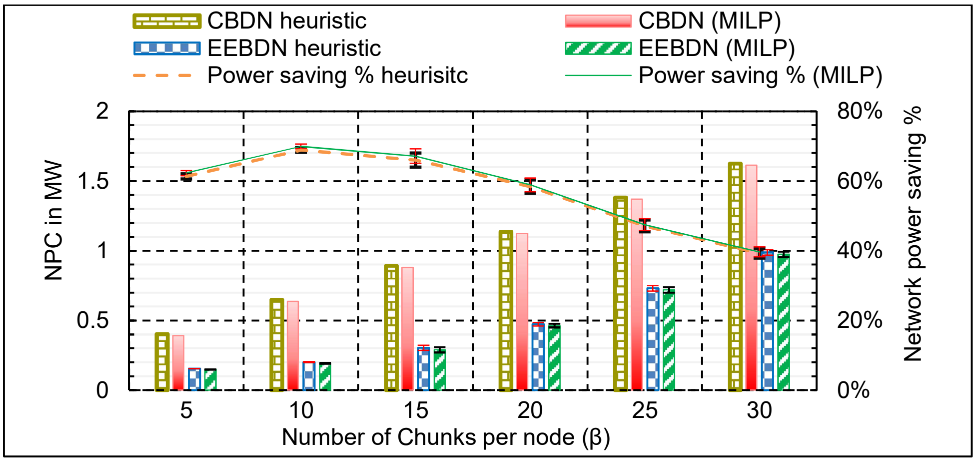


(a)

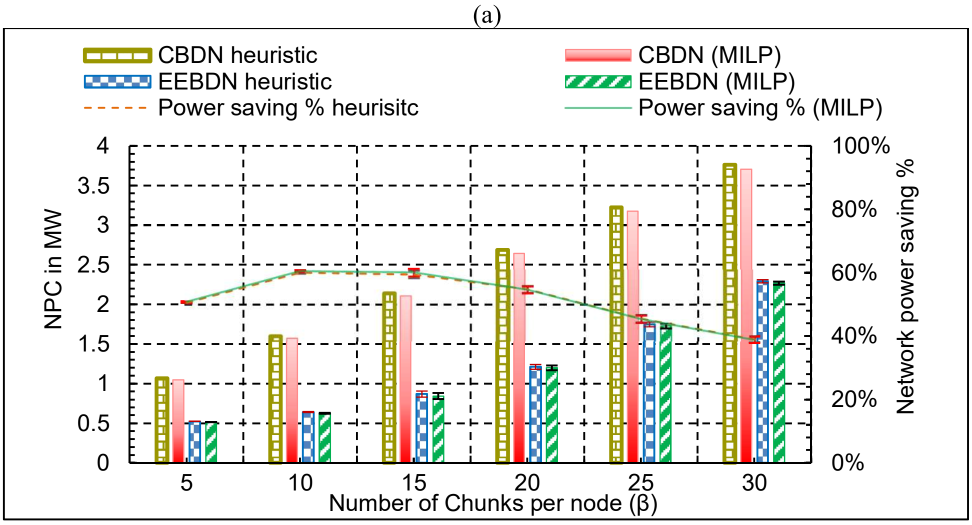


(b)

Figure 4-4: The COST239 network, and (b) the Italian network.



(a)



(b)

Figure 4-5: CBDN power consumption vs EEBDN power consumption (a) COST239 network, (b) Italian network.

On the other hand, the power savings of the MILP and heuristic obtained in the Italian network are 52% and 51%, respectively for the volume scenario. Note that nodes #4 and #8 are selected as the best DCs locations in COST239 network, and nodes #7 and #9 are selected in the Italian network. These node selections were made based on a MILP optimisation similar to that in [43]. The heuristic and MILP results are in close agreement in the cases described above.

4.2.2 Scenario #2: Deterministic *Chunk*s volume, PRR = 0.001, number of servers per PN = 10-30 servers

In this scenario, we considered a variation of scenario A where the average processing capability per node is increased but the processing capacity per node remains random between 10 and 30 servers instead of 5 to 15 servers. This is to study the effect of increasing the processing capacity on the progressive processing of larger big data volume, which eventually influences the energy efficiency of the network. See Table 4-5.

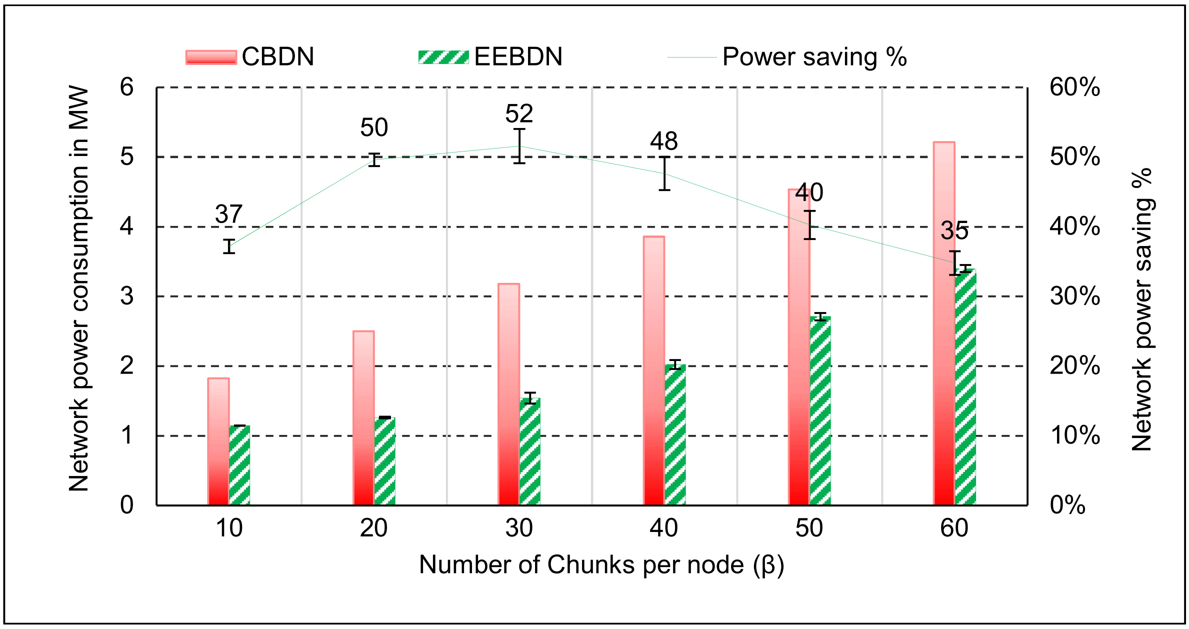
Number of *Chunks* per node (β)	Number of servers per PN (NS_p)	CPU workload per *Chunk* in GHz (SW_{sc})	*Chunk* volume in Gb (CHV_{sc})	PRR_{sc}
10-60	10-30	3 [75]	80 [23]	0.001[23]

Table 4-5: Volume Scenario #2 parameters.

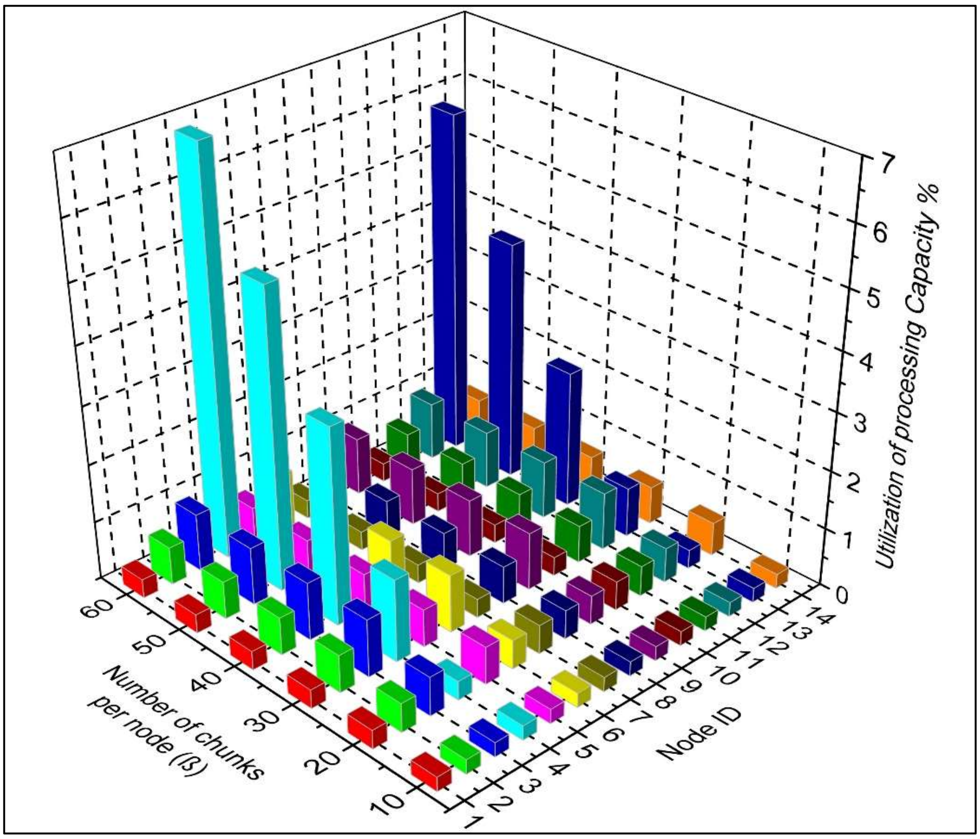
Figure 4-6-a shows the NPC of the classical networks and EEBDN. The power saving increased at $10 \leq \beta \leq 30$ and reached a maximum value of 52% at $\beta = 30$ (compared to the maximum power saving of 38% at $\beta = 15$ in Scenario #1). This is because most of the big data traffic in the network is the INF when $10 \leq \beta \leq 30$ with a small amount of CHT as most of the *Chunks* are processed locally and in the intermediate nodes. After that point (i.e., $\beta > 30$), the CHT between the PNs and DCs dominates the network where the computing resources of all PNs are depleted, which leads to reduced power savings. However, the average power saving increases to 44% for $10 \geq \beta \geq 60$ (higher than the average power saving of 32% in Scenario #1) as more *Chunks* are processed in SPNs and IPNs. Thus, increasing the PNs processing capacity has a positive impact on both the average network power saving and the total number of served *Chunks* in the system.

It should be noted that a full treatment of the internal design of processing nodes requires consideration of their internal architecture. For example, a fat-tree, spine-and-leaf, D-Cell or some other data centre architecture. This is however beyond the scope of the current work. It also introduces high complexity that is hard to handle in the MILP. We calculated the number of switches and routers needed by considering the amount of traffic arriving to a processing node and the data rate that can be handled by a switch or a router. This approach is appropriate for the ingress/egress router, which must handle the entire PN traffic. The approach however replaces the many small switches in the fat-tree or spine-and-leaf by a single large switch, or few large switches. This is not a typical implementation; however, it may be considered in our small processing nodes that have 5 to 15 servers or 10 to 30 servers (maximum 60 servers). It is an approach, which simplifies the models used. Typically, in current data centres, about 90% of the power consumption of IT is attributed to servers and 10% to communication equipment [123]. Therefore, having considered the power consumption of a large switch (or few large switches) instead of multiple smaller switches (and their architecture) results in changes in power consumption bounded typically by less than the 10% figure.

Figure 4-6-b shows that the SPNs now can locally process all the *Chunks* when $\beta = 10$ since their processing capacity has increased. At $20 \leq \beta \leq 40$, PNs start to reach their maximum processing capacity, such as PNs #2 and #3 at $\beta = 20$ and 30, respectively.



(a)



(b)

Figure 4-6: (a) CBDN power consumption vs EEBDN power consumption for volume Scenario 2. (b) Utilisation of processing capacity % in EEBDN with different values of *β* for volume Scenario 2.

Note that only at $\beta = 20$ is the processing utilisation of nodes #7 and #10 > 100% because they are selected as DCs, while at all other values of β, nodes #4 and #13 dominate the selectivity of DCs locations, as in Scenario #1. We also note that the processing utilisation of the DCs of the present scenario is smaller than that of Scenario #1 at $\beta = 30$, at which the DC utilisation reaches the maximum value for in Scenario #1. This is due to the growth in the PNs' processing capability in the current scenario, which helps to reduce the DCs' processing utilisation.

In summary, increasing the PNs' processing capacity has a noteworthy impact on network power saving as the volume of the processed big data inside SPNs and IPNs increases, which results in serving a larger number of *Chunk*s as close to the edge as possible. In both scenarios, these are very general results as they contain all the cases, which are full edge processing inside SPNs when big data traffic is small, progressive processing inside IPNs for intermediate levels of big data traffic, and full central processing inside the DCs when the volume of big data traffic is very high.

4.3 Assessing the Energy Efficiency Limits of PNs in the EEBDN

Progressive processing is an appealing approach to reduce the power consumption associated with big data traffic as illustrated in the previous sections. In reality, however, PNs might be equipped with components that have lower energy efficiency compared to those hosted in the centralised DCs. This might be due to technology, economic and/or space limitations in PN sites. In this section, we analysed the impact on the power consumption of EEBDN of utilising less energy efficient equipment (servers, LAN switches and routers) in PNs compared to the large DCs. Two cases were studied: (i) PRR=0.001 and (ii) PRR=0.6, with β =30 (*Chunk*s per PN) in both cases.

4.3.1 Results

Results are shown in Figure 4-7 where the y-axis is NPC. Note that total power consumption follows similar trends. The x-axis represents PNs power consumption as a percentage of the data centres equipment power consumption. For instance, 10% means that the PNs equipment consume 10% more power compared to the corresponding equipment in the DCs. Therefore, if the DC server consumes a maximum of 300 W, the PNs server consumes a maximum of 330 W. Since the equipment in the classical approach is regarded as the basis of this comparison, the total power consumption in the classical approach is not affected by this analysis as shown in Figure 4-7 (red bars). In addition, and to reduce the complexity of the analysis, the DCs are fixed in the optimal locations obtained in Section 4.2.1and 4.2.2 as their location is not the critical element that we assessed.

When PRR=0.001 (green bars) and the PNs equipment power consumption is 0% to 60% greater than the DCs equipment's power consumption, the power saving is at its maximum. After this critical stage, the energy efficiency of our approach declines gradually, approaching the energy efficiency of the classical central processing approach (i.e. 80% case). Comparing this to the case where the PRR=0.6, we noticed that our approach is useful only when the PNs equipment power consumption is between 0% and 20% greater than the DCs equipment power consumption. Beyond this range, the optimal solution is processing the majority of the *Chunk*s in the centralised DCs rather than in the PNs. Therefore, our approach is the better approach at a wider range of energy inefficiency values at PNs when the type of big data applications allows for higher reductions (i.e. lower PRRs). This is because lower PRR is associated with higher network power savings, and to lose this high saving, the PNs need to be implemented using equipment with lower energy efficiency (70% to 90% less energy efficient than the DCs). Our goal here is to show the impact of processing locally versus processing totally in the central data centres. Total processing in the central data centres becomes more attractive at the point when PNs are 90% less energy efficient than data centres and here the long journey to central data centres just becomes viable, comparing for example PRR=0.001 and the 80% and 90% cases, for the set of power consumption parameter we used. In practice, such a point may not be reached with current equipment trends and therefore edge processing may remain viable for big data even when the edge equipment is not as energy efficient as the central data centre equipment.

An extreme potential scenario may be a situation where the central data centre power usage effectiveness (cooling, lighting) becomes a factor of 2 better than the edge PN power usage effectiveness and PNs are made of conventional processors that are four to five times less energy efficient than the best recent processors that have 64 cores which may be used in data centres in future [124, 125]. This situation is represented by the 90% case in Figure 4-7.

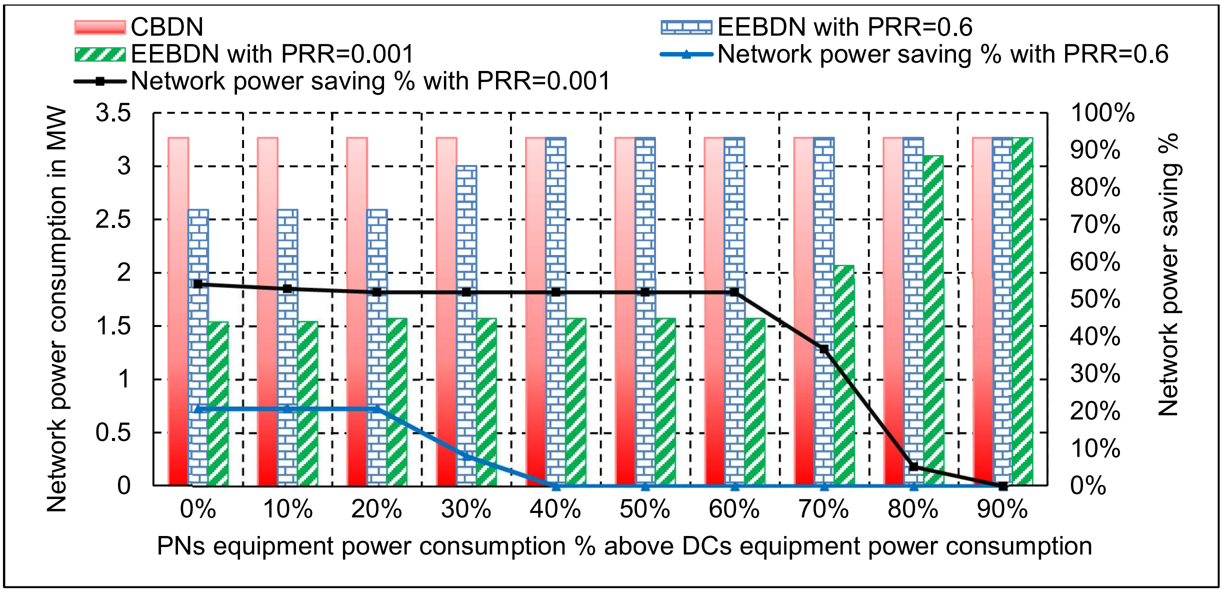


Figure 4-7: CBDN power consumption vs EEBDN power consumption when PNs equipment consume more power than DCs equipment at PRR=0.001 and PRR=0.6 with β=30.

We conclude that at lower PRR, the EEBDN can host energy inefficient equipment in PNs and yet gain considerable network power saving. However, at higher PRR, i.e. higher INF traffic, the network power saving is already small; therefore, the network can only sustain PNs equipment with energy efficiency values very close to those in data centres, e.g. within 20%.

4.4 Software Matching Problem and Its Effect on EEBDN Performance

Another idealistic assumption made in the previous sections is that all PNs can process all types of big data *Chunk*s, i.e. they are provided with all the necessary software packages that correspond to all the possible types of *Chunk*s. This is obviously not possible in small sized PNs due to processing and storage limitations. Therefore, in this section, we assessed the impact of software shortage in PNs on the performance of the overall EEBDN approach in terms of number of processed *Chunk*s at the edge. This analysis is carried out by extending our model to include a software matching dimension where *Chunk*s are associated with the correct PNs hosting the appropriate software package that can process that *Chunk*. Note that DCs are assumed to host all the software packages needed. Therefore, if the software required by the arrived *Chunk* is not available in the receiving SPN, the SPN forwards (i.e. matches) that *Chunk* to the nearest IPN/DC that host the required package. In the software matching problem, it is worth noting that big data applications may be numerous covering for example healthcare, vehicular, smart city, manufacturing, agriculture, financial and other applications. Therefore, a single PN may not hold a full suite of software packages to support all the applications, due to size (storage for example) limitations, or due to security, isolation and resilience requirements where some high value (e.g. financial) or life critical (e.g. healthcare) applications must be segregated.

4.4.1 MILP model extension description

In addition to the parameters mentioned in Section 4.1.2, we define the following parameters in Table 4-6:

Notation	Definition
S	Set of all software packages
$PK_{p,g}$	$PK_{p,g} = 1$ if software package g ($g \in S$) is available at node *p*; otherwise, $PK_{p,g} = 0$.
$CSF_{s,c,g}$	$CSF_{s,c,g}$=1 if *Chunk*s *c* generated at node *s* needs software package *g*; otherwise, $CSF_{s,c,g} = 0$.

Table 4-6: List of parameters and their definitions.

In addition to the constraints mentioned in Section 4.1.2, we defined the following constraint:

$$Y_{spc} \leq \sum_{g \in S} CSF_{s,c,g} \cdot PK_{p,g} \qquad (32)$$

$$\forall s, p \in N, \forall c \in CH_s$$

Constraint (32) ensures that *Chunk c* generated at node *s*, which requires software package *g*, can be processed at node *p* if *p* contains the required package *g*.

4.4.2 Results

We assume that each PN receives β (=10) *Chunks* from its corresponding zone, and each *Chunk* needs a unique software package. This models the case where *Chunks*' population spans a wide spectrum of types. In addition, we analysed a different number of packages per PN. In each case corresponding to a certain number of packages per PN, all PNs host the same types of packages.

Hosting different types of packages can only be an informed decision when the packages are optimally placed at PNs, which we left for future work. Also, recall that DCs contain the set of all software packages. Upon the arrival of the *Chunks*, the SPN decides whether to process the *Chunks* locally based on software availability, otherwise, the *Chunk* is forwarded to the nearest DC.

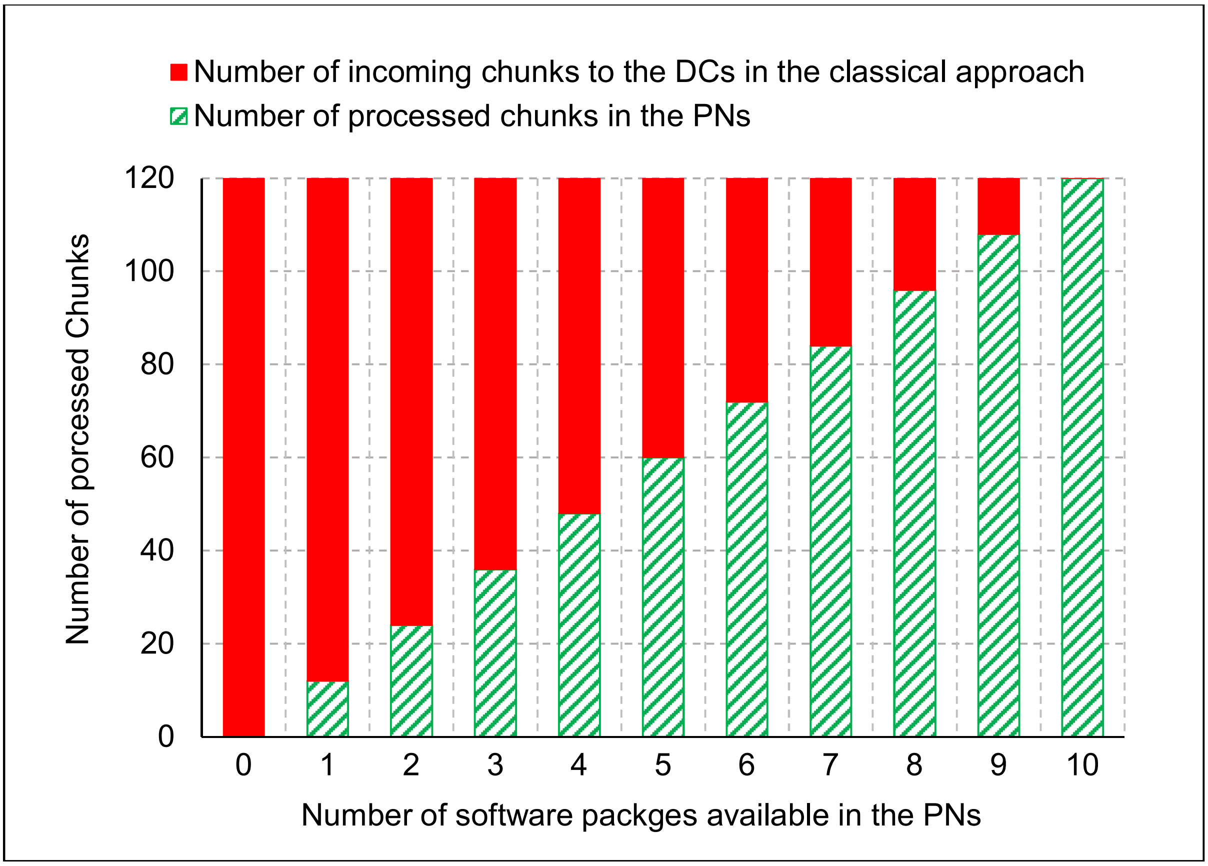


Figure 4-8: Software packages availability and its impact on EEBDN performance at β=10.

Figure 4-8 shows the effect of software package availably inside the PNs on the network performance at β=10. The x-axis represents the number of packages per PN, while the y-axis quantifies the number of edge processed *Chunks* (i.e. in SPNs).

Since we assumed that the packages are homogeneously distributed among the PNs (i.e. all PNs host similar packages), when a *Chunk* is not matched to its SPN due to lack of the required package, this *Chunk* cannot be matched to any other IPN and it is processed in the central DCs. The extreme example for this case is when all PNs lack all packages as shown in Figure 4-8 at 0 number of packages per PN.

The performance of our approach almost linearly improves with the availability of more software packages in PNs as more *Chunks* are processed in the edge network while the rest are forwarded to the DCs. When PNs host the full package set, the maximum performance can be reached as all *Chunks* are processed locally in the edge SPNs. Note that in this case, there are 120 software packages running in the network. The number of running packages in the network can be reduced by optimally allocating packages to PNs according to the incoming *Chunks*-SPN-packages distribution. This can guarantee processing all *Chunks* with a smaller number of software instances in the network.

The proposed edge processing (with progressive processing) approach, may increase the number of software packages installed, however this may not have a direct cost implication. Typically, site software licenses can be offered which cover all the sites of the user. If, however, a given software package does not offer this facility then the extra cost may be offset by the financial savings as a result of energy savings, however techno-economic studies are beyond the scope of this work. It is also worth highlighting the fact that the non-availability of a software package in a close-by PN may lead to longer journeys in the network and increased power consumption. Figure 4-8 showed the split between edge and central processing as nodes have more of the software packages, up to the point where every node has all the software packages.

4.5 Energy Efficient Big Data Networks: Impact of Variety

Variety means that there are different types of big data such as CPU intensive, memory intensive, Input/output (IO) intensive, CPU-Memory intensive, CPU/IO intensive, and memory-IO intensive applications. Each requires difference amount of processing, memory, storage, and networking resources.

The different types come from the diversity of big data sources, such as sensors, smart devices, and social networks, etc. Therefore, big data has a complexity feature as it comes from not only traditional structured data (e.g., customer data, sales data) but also unstructured (e.g., social media, photos, PDF) and/or semi-structured, which is a combination of both (e.g., email, XML). Such complexity can cause traditional database systems to struggle to store, process and analyse big data to obtain useful information since they are not related to the relational database technologies. Successful organisations that rely on big data to enrich their decision-making should be able to handle the variety of data [8].

4.5.1 Variety MILP model

The MILP model presented in Section 4.1.2 is also used to evaluate the impact of variety on EEBDN. However, the input data to the model is modified to satisfy the distinct features of the variety domain.

4.5.2 Results of variety scenarios

We presented in this section the following two scenarios:

4.5.2.1 Scenario #1: Deterministic CPU workload per *Chunk* with different PRR per *Chunk*

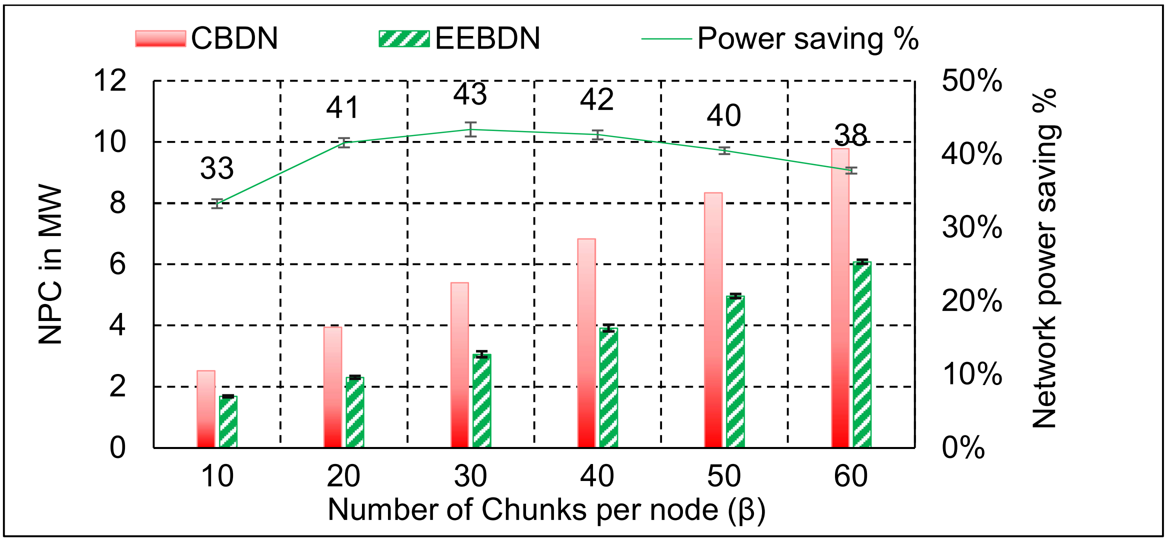
In this scenario, all *Chunks* have similar CPU requirements while they exhibit different PRRs. Each PRR could represent a particular application that encodes information differently. Table 4-7 shows the input parameters used in this section. All nodes generate 10 to 60 *Chunks* and the volume of *Chunks* varies between 10 and 330 Gb using a random uniform distribution. The values of the PRR_{sc} of *Chunks* range between 0.001 and 1 per *Chunk*, i.e., some *Infos* volume would be equal to its corresponding *Chunk* volume, with PRR_{sc} being generated using a random uniform distribution. Each *Chunk* demands CPU workload of 3 GHz.

Number of *Chunks* per PN	10-60 (random uniform)
Number of *Chunks* per PN	10-30 (random uniform)
CPU workload per *Chunk* in GHz (SW_{sc})	3 [75]
Chunk volume in Gb (CHV_{sc})	10-330 (random uniform) [23]
PRR_{sc}	0.001-1 (random uniform) [23]

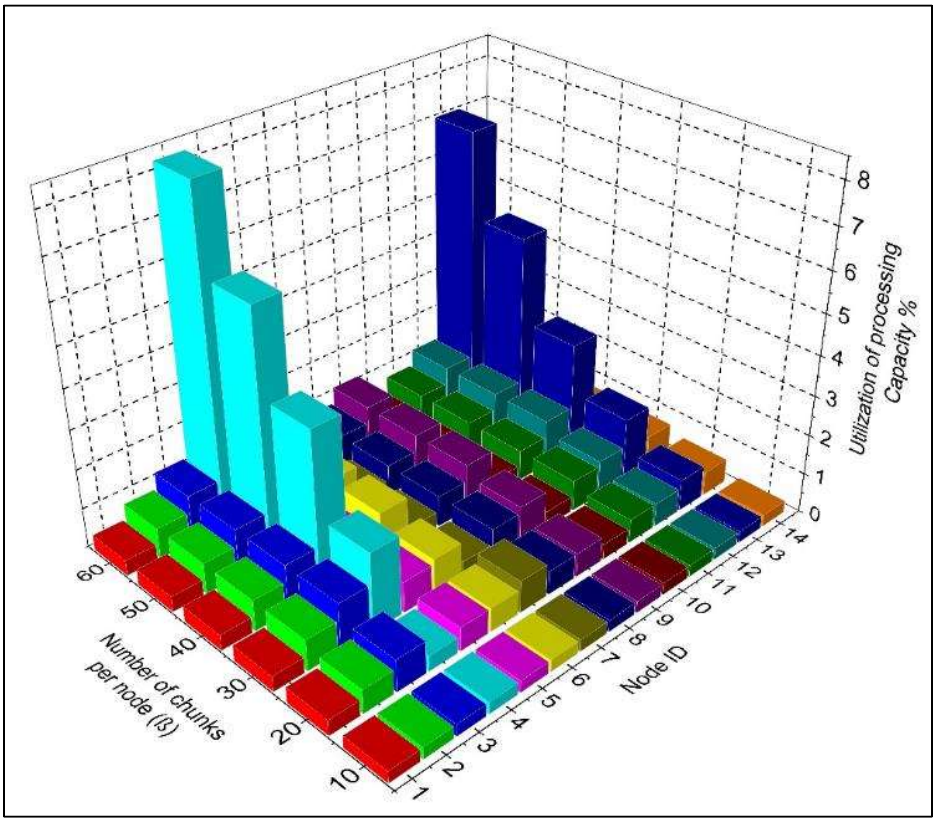
Table 4-7: Variety Scenario #1 parameters.

Figure 4-9-a shows that the maximum power saving is 43% at $\beta = 30$ *Chunks* and the average power saving is 40%. An interesting feature in this figure is the effect of “variety of big data applications” on the network power saving compared to the previous section, i.e., volume Scenario #2, where we obtained an averaged maximum power saving of 52% with a single value for PRR = 0.001. PRR in this scenario covers values with small reduction percentages, i.e., INF volume is larger here compared to the volume Scenario #2, which reduces the power savings achieved. From the results in Figure 4-9-b, it is apparent that the system shows similar performance to volume Scenario

#2, in which some PNs exhaust their processing capacity earlier than other nodes, such as PNs #1, #7 and #11 at β = 10, while other PNs are fully utilised later as is the case for PN #9 at



(a)



(b)

Figure 4-9: (a) The CBDN power consumption vs the EEBDN power consumption for variety Scenario #1. (b) Utilisation of processing capacity % in EEBDN with different values of β for variety Scenario #1.

β = 40. Note that the selected DCs in all cases here are nodes #4 and #13 when applying the different number of *Chunks* per node. The similar performance of this scenario and volume Scenario #1 is due to the assumed insensitivity of CPU utilisation to different values of PRR_{sc} as shown in Table 4-7.

4.5.2.2 Scenario #2: Different CPU workloads and PRR per *Chunk*

This scenario further investigates the effects of various data types on the overall network and PN performance. We considered different volumes of big data *Chunks* generated by PNs with different PRR_{sc} per *Chunk*, such that each PRR_{sc} represents a specific type of data and each *Chunk* acquires a distinct CPU portion to reflect a more realistic picture for the network. Table **4-8** shows that the CPU workload per *Chunk*, *Chunk* volume and PRR_{sc} per *Chunk* which follow a random uniform distribution between 1 and 4 GHz, 10 and 330 Gb and 0.001 and 1, respectively.

Number of *Chunks* per PN	10-60 (random uniform)
Number of *Chunks* per PN	10-30 (random uniform)
CPU workload per *Chunk* in GHz (SW_{sc})	1-4 [75]
Chunk volume in Gb (CHV_{sc})	10-330 (random uniform) [23]
PRR_{sc}	0.001-1 (random uniform) [23]

Table 4-8: Variety Scenario #2 parameters.

Figure 4-10 shows a sample of the input data for this scenario considering node #8 at β = 10. Note the variation among different *Chunk*s in terms of volume, processing requirements and reductions ratios. This is because big data applications and forms are growing at an incredible rate, therefore we explored the conceivable space in this scenario. For instance, *Chunk* #1 has a large volume and requires high processing workload and produces information with very small volume (i.e. high reduction ratio). This can represent a WordCount program [126], which is both CPU-intensive and network intensive as an application. This program reads text input files to search and count the number of occurrences of a specific word to produce a very small volume *Info* that is only an integer number indicating the count value. *Chunk* #4 comes with large volume, needs large processing resources and produces large volume *Info*.

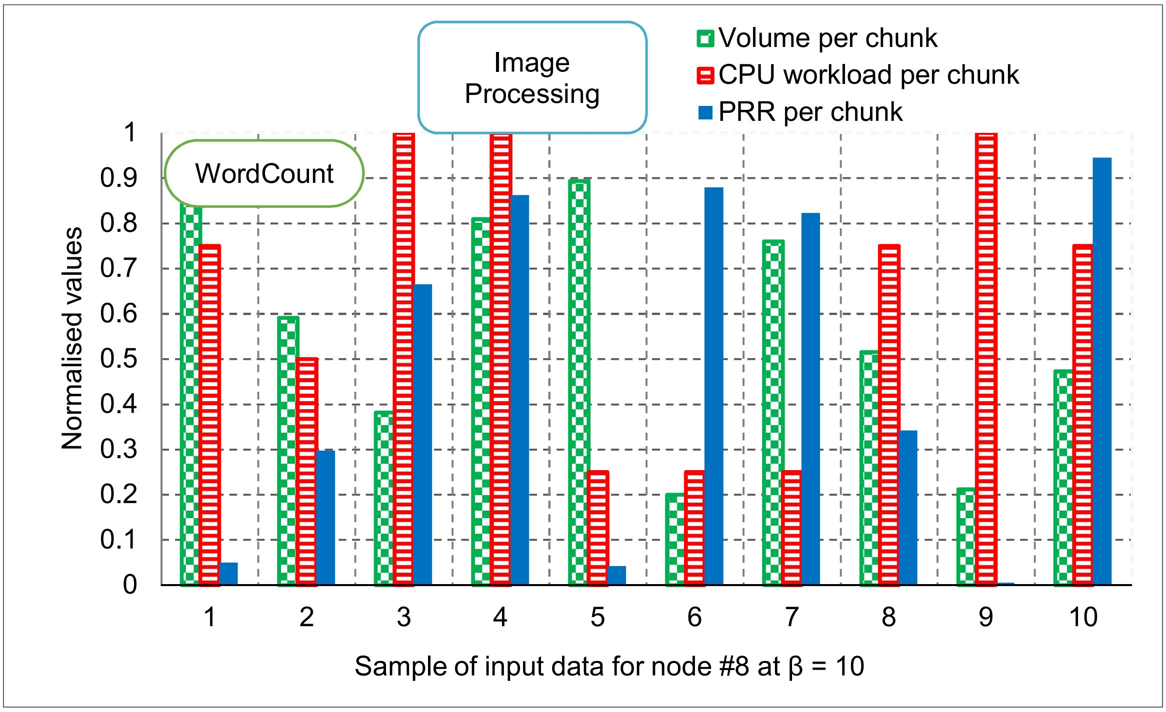


Figure 4-10: Sample of input data for variety Scenario #2 for node #8 at β = 10.

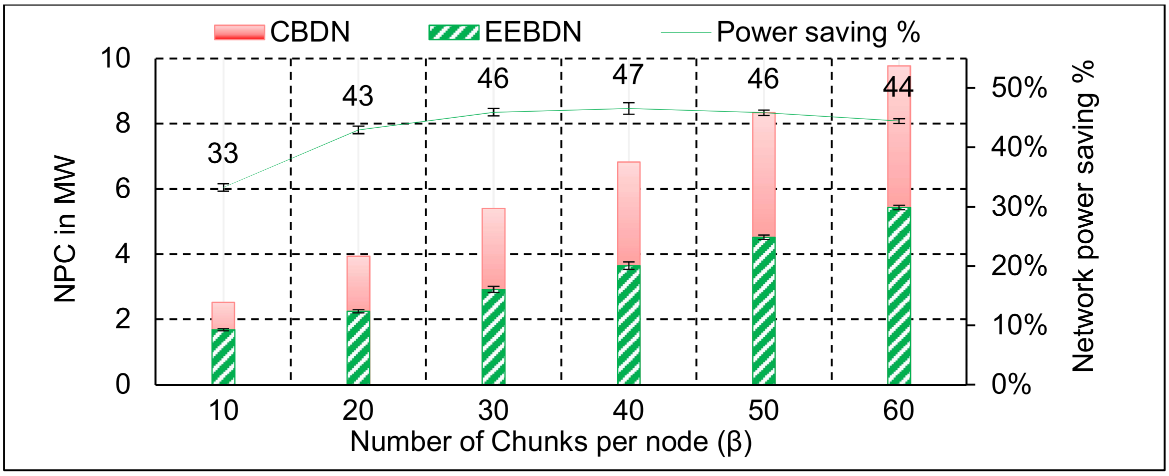
This can represent an image processing application that modifies certain properties of an image, such as brightness level, which does not result in a huge reduction in image size. Figure 4-10 shows those two points in the explored space and displays their corresponding applications.

Figure 4-11-a displays the main findings and differences with the previous variety Scenario #1, where the CPU workload per *Chunk* was fixed at 3 GHz. The main observations are as follows: first, the maximum power saving (47%) exceeds the one obtained in the variety Scenario #1 (43%).

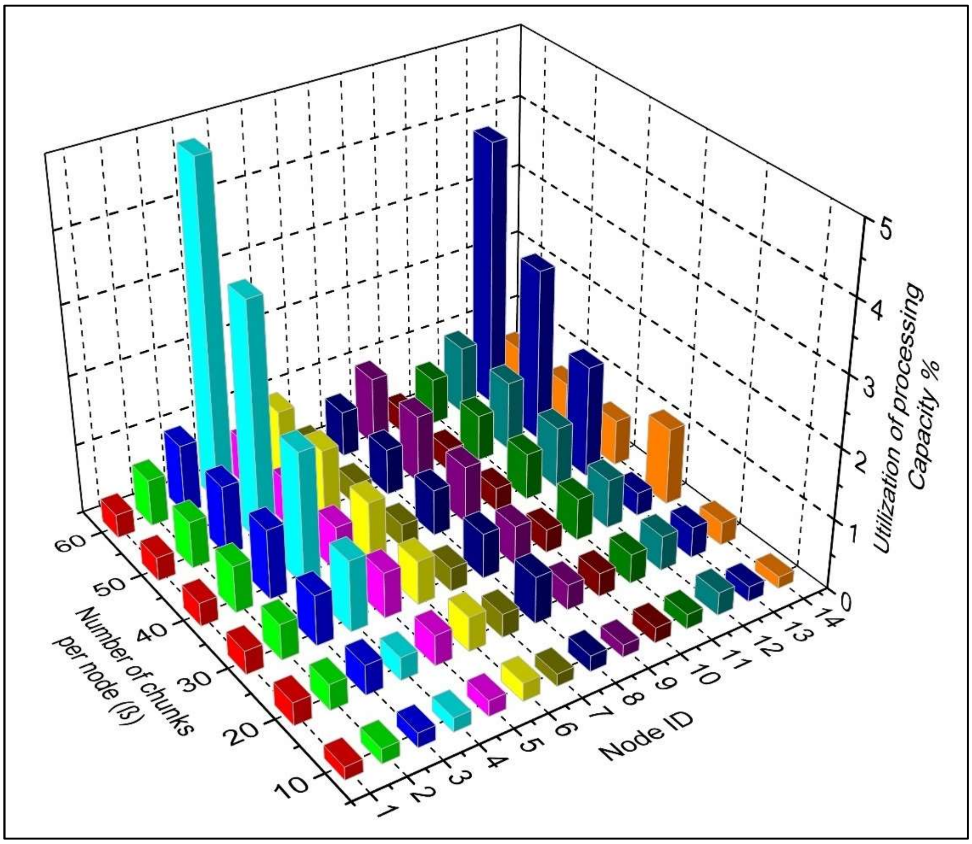
This is due to the ability to consolidate the CPU processing for more *Chunks* by PNs as some *Chunks* arrive with lower processing requirements compared to the variety Scenario #1. The same observation is also true for the average power savings (43% for the current scenario, 40% for variety Scenario #1). Second, the maximum power saving occurred at β = 40 and not β = 30 as observed in the variety Scenario #1. This is also due to the extra available processing space at the IPNs due to processing *Chunks* with low processing requirements.

This section tries to capture the distinct features of variety by allowing the modelled big data network to handle *Chunks* associated with different reduction ratios, CPU processing requirements, and volumes. There are a number of key take away messages. Figure 4-11-a shows results when the network has regular traffic and big data traffic. Here the larger the big data traffic, the more is the traffic reduction that can be achieved by processing big data and hence the larger the power saving. As shown in Figure 4-11-a, however, beyond a certain big data traffic volume, the processing capability of PNs at the edge gets depleted and the power savings drop (in Figure 4-11-a, the savings drop from 47% to 44%). In addition, big data applications that have small PRR (i.e. large reduction after processing) are critical in terms of network power saving and hence should be given priority. Figure 4-10 shows example applications and their PRR values.
Figure 4-11-b shows the processing utilisation for the different PNs and DCs. Firstly, we noted that some PNs are capable now of serving more *Chunks*. For instance, PNs #1 and #7 can process locally up to 20 *Chunks*, while other PNs can serve up to a maximum of 40 *Chunks,* such as nodes #3 and #6. This is higher than what some PNs in the variety Scenario #1 could serve. This shows the impact of having various CPU workloads per *Chunk* which extend the PNs ability to server

more *Chunks* with lower CPU requirements. Secondly, the model, in most stages, selected optimally the prevailing DC locations at nodes #4 and #13.



(a)



(b)

Figure 4-11: (a) CBDN power consumption vs EEBDN power consumption for variety Scenario #2. (b) Utilisation of processing capacity % in the EEBDN with different values of β for variety Scenario #2.

Note that we re-evaluated our results where we optimised the locations of 5 DCs rather than 2 DCs locations in the NSFNET. Nodes #1, #4, #7, #9 and #13 are the optimum DCs locations for all scenarios including the classical approach. Under the 2 DCs scenario the EEBDN approach resulted in up to 52% and 47% power saving compared to CBDN approach under the volume and variety scenarios. With 5 DCs the savings increased to 54% and 48% under the volume and variety scenarios due to the availability of more nearby destinations for the data.

Furthermore, it should be noted that energy efficiency in core networks is essential due to the high energy density in core nodes and the increasing power consumption of large data centres which are placed in the core network, a view shared by GreenTouch where the GreenTouch effort resulted in the development of methods to improve the energy efficiency of core networks by 316x compared to their 2010 levels[43], [54], [127]. Our work here considers big data traffic as well as regular traffic. For regular traffic see equations (4-31), (4-32), (4-33), and (4-34), and for example the explanation of Figure 4-11-a. The interest in big data is attributed to its large volume and the ability to reduce this volume through processing, hence saving power. Table 4-9 summarises the results obtained in the two scenarios.

Parameters	Scenario	
	#1	#2
Volume per *Chunks* in Gb ($Chunk_{sc}$)	10-330 (random uniform)	10-330 (random uniform)
PRR_{sc}	0.001-1 (random uniform)	0.001-1 (random uniform)

CPU workload per *Chunk* in GHz (W_{sc})	3	1-4 (random uniform)
Number of servers per PN (NS_p)	10-30	10-30
Results	Scenario	
	#1	#2
Maximum network power saving	43%	47%

Table 4-9. Variety Scenarios Summary.

4.6 Assessing the EEBDN by Considering Different Power Profiles

In the previous sections we implemented the EEBDN using the power profile of 2010 [54], where the IP over WDM network equipment consumes high power. In this section, however, we assessed the efficiency of our EEBDN using two different power profiles compare to the 2010 power profile. These two power profiles are based on the projections of the 2020 equipment power consumption. The first one is the Business as Usual (BAU) power profile, and the second one is the GreenTouch Business as Usual (BAU+GT) power profile [128]. Table 4-10 shows the power consumption values of various components that have been used for the MILP model for a 2010 network and a 2020 BAU and BAU+GT network worked out according to the methods aforementioned.

Device	2010 power consumption	2020 power consumption	
		BAU	BAU+GT
Router Port 40 Gb/s	825 w	178.2 w	21.3 w
Transponder 40 Gb/s	167 w	35.7 w	27.6 w

Regenerator 40 Gb/s	334 w	71.4 w	55.2 w
EDFA	55 w	15.3 w	15.3 w
Optical Switch	85 w	85 w	8.5 w

Table 4-10 Power Consumption Values.

We re-evaluated the volume scenario #1 for the MILP model but with using the power profiles BAU and BAU+GT and compared it to 2010 power profile. Results shows that our EEBDN performs efficiently using power profiles 2010 and BAU as the power savings reached up to 38% and 37%, respectively. Using the BAU+GT power profile, had, however, slightly reduced the power saving and reached up to 33%. This means that our approach performs efficiently even if the energy efficiency of the network equipment is improvement by 315x [128].

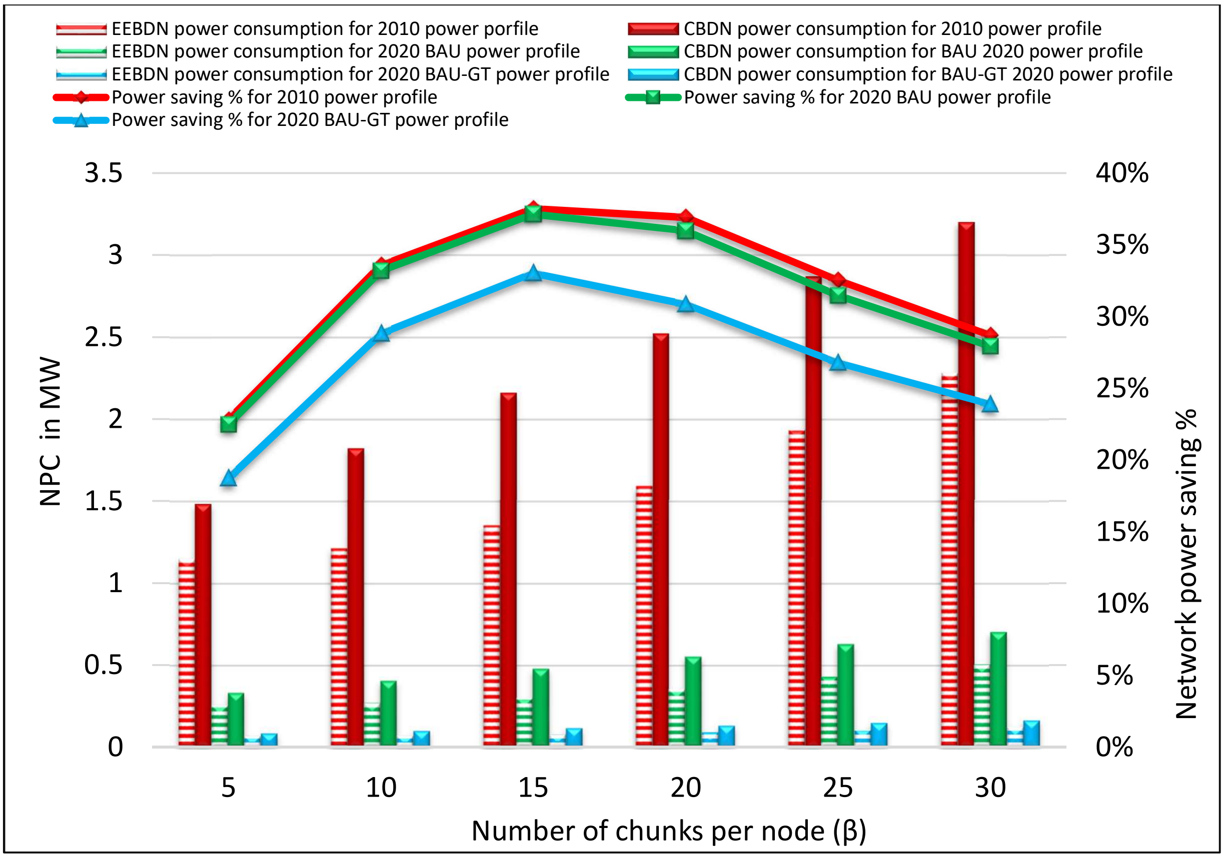


Figure 4-12: CBDN power consumption vs EEBDN power consumption of 2010, 2020 BAU, and 2020 BAU+GT power profiles.

4.7 Summary

This chapter presented a Mixed Integer Linear Programming (MILP) model to study the impact of big data's volume and variety on network power saving when such networks carry big data traffic. We employed our progressive processing technique to process big data raw traffic in the edge stage, intermediate stage, and the central processing stage. This is done by building Processing Nodes (PNs) in the ISP network centres that host the IP over WDM nodes. A PN is a mini datacentre (DC) with a limited processing and storage capacity depending on the available building space inside the core centre. The volume scenarios captured generic results that showed how the processing capability of the PNs dictates the big data volume that exists in SPNs, IPNs and data centres. We obtained up to 52% and 34% of network power saving in two different volume scenarios, compared to the power consumption of the classical processing approach where the *Chunks* are directly forwarded from the source node to the DCs.

The results of the MILP model for the volume dimension are validated by developing a heuristic that mimics the MILP model behaviour. We further assessed the energy efficiency limits of PNs in the EEBDN and the results showed that employing PNs equipment with lower energy efficiency compared to the DCs equipment led to lower utilisation in our approach. Furthermore, we analysed a software matching problem and its impact on EEBDN performance. The results revealed that the performance of our approach improves with the availability of more software packages in PNs as more *Chunks* are processed in the edge of the network and the approach reached maximum performance when PNs host the full software package set. The variety scenarios revealed the impact of serving *Chunks* with different CPU workloads, volumes and PRRs on the power saving. In view of that, *Chunks* that utilise small portions of the CPU help the nodes process as many

Chunks as possible inside the local servers, hence, reducing the number of unprocessed *Chunks* in the network. We obtained up to 47% and 43 % of network power savings in two different variety scenarios.

Chapter 5 Energy Efficient Big Data Networks: Impact of Velocity

Velocity is a property of data in motion. It is the speed at which data is fluxing in to be processed [8]. The flux rate can grow larger for applications collecting information from wide spatial or temporal domains. For instance, the Square Kilometre Array [109] telescope combines signals with a flow speed of 700 TB/second of data received from thousands of small antennas spread over a distance of more than 3000 km. In another example, five million trade events created each day are scrutinised to identify potential fraud [129]. Five hundred million daily call detail records are analysed in real-time to predict customer churn faster [129].

An effective tactic to deal with the velocity of big data is to perform analysis while flowing and not only after being stored because there is an immense amount of data that has very short life cycles [129]. This requires extracting the useful knowledge from big data in near real time. Therefore, the intention of the present study in this chapter is to analyse both time dependent types of big data applications: expedited-data processing and relaxed-data processing.

This chapter, makes a number of new contributions beyond the previous chapter as follows: Firstly, we developed a MILP model to examine the impact of the velocity of big data on network power consumption in bypass IP over WDM core networks. We consider an expedited-data processing mode and a relaxed-data processing mode. In the relaxed-data mode, the execution time needed to process an application is relatively long as it can tolerate some delay. In the expedited-data mode, the execution time required to process delay sensitive applications is optimised to be as short as possible. Secondly, we extended the objective of the MILP model so that it minimises the network power consumption as well as minimising the execution time of big data applications. The addition of the time dimension is essential when considering big data applications where velocity (time sensitivity) is an important attribute. Thirdly, we used our progressive processing technique to process big data *Chunk*s and compared the results to the classical approach where progressive processing is not allowed. In our approach, the processing locations are optimally selected at Source PNs (SPNs), at the Intermediate PNs (IPNs) or inside the centralised data centres (DCs). As a result, a significant reduction in the network power consumption is achieved each time the data is processed along the journey from the source to the DCs. Note that the main similarities in all the MILP models are due to the comparable goals of optimising the processing locations of *Chunk*s, optimising the locations of the DCs, ensuring the flow conservation of big data traffic, and minimising the power consumption of PNs, DCs, and IP over WDM network. In summary, the differences between the different MILP models we developed reflect the different requirements and features of big data forms/applications where a particular big data V may be important.

The intention of this chapter is to analyse both time dependent types of big data applications: expedited-data processing and relaxed-data processing. Relaxed-data processing can tolerate some delay and can be processed in a batch processing mode after being stored inside DCs, such as digital image processing and automated transaction processing. Several benefits can be gained in batch processing jobs, such as avoiding the idle status of computing resources by shifting the time of job processing to less busy hours, hence, gaining a higher overall rate of utilisation. Further, batch processing reduces the system overhead by running a program one time to achieve multiple tasks for the same job rather than running that program many times to perform those different tasks. On the other hand, in expedited-data processing, it is essential to analyse data as fast as possible to maximise its value while fluxing into the DCs. For instance, sometimes two minutes delay is too much to catch fraud or it could lead to a disaster, such as the situation of remote patient monitoring that requires the analysis of the abnormality in their sensed organ readings almost immediately. An effective method to quickly process data is to provide sufficient and efficient computational resources to decrease the processing latency of such CPU intensive applications. This can be done by optimally allocating processing workloads according to the data type. If it is expedited-data, then the allocated CPU should be a large portion of the CPU processing capacity, so as to serve the expedited-data quickly. Therefore, increasing the CPU frequency has a positive impact on decreasing the execution time of CPU intensive applications [9]. Equation (5-1) represents the CPU performance relationship [130].

$$\frac{CPU\ Execution\ Time\ (CET)}{Instruction\ Count\ (IC)} = \frac{Cycles\ per\ Instruction\ (CPI)}{Processing\ Workload\ (PW)}. \tag{5-1}$$

The term CET represents the total duration a CPU requires to execute a program with a certain number of instructions (IC). Note that the program is used to extract useful knowledge from a given *Chunk*. The term CPI is the average number of clock cycles needed to execute each instruction of that program. PW represents the CPU cycles per second in GHz used to process a given *Chunk*. Note that in [130] PW is referred to as Clock Rate.

In our model, *Chunk*s initially request a certain CET, called $RCET$. Based on the $RCET$, CPI and IC, the initially requested PW for *Chunk*s is deduced, called RPW. However, to expedite the process, the model can allocate processing workloads for *Chunk*s that exceeds the RPW, called Allocated PW (APW). Based on APW, CPI and IC, the model can deduce the optimal CPU execution time to process a *Chunk*, referred to as Allocated CET ($ACET$). Note that $ACET$ can be shorter than $RCET$ for expedited-data processing.

5.1 Velocity MILP Model

In this section, and for the completeness of our work in the previous chapter, we introduced a MILP model for the EEBDN in the bypass approach of the IP over WDM network. We attached capacitated PNs at each core node of the NSFNET, as shown in Figure 4-1, with DCs with large enough capacities. The DCs are employed to process all incoming big data *Chunk*s from PNs. Further, the DCs receive *Infos* produced by the PNs.

We performed the MILP optimisation using the AMPL/CPLEX software running on a PC with 8 GB RAM and an i5 CPU. The model execution takes few minutes to around two hours to solve the problem in the scenarios studied in the work. However, for faster results and larger networks, the current MILP can be applied using a High-Performance Computer (HPC). For example, we used a Polaris machine with 16 cores (processors) and 256 GB of RAM. Furthermore, a heuristic can be implemented to achieve two main purposes. Firstly, as a verification of the MILP results and secondly, since the heuristic uses simple rules, it runs fast unlike the MILP. Therefore, a heuristic can enable network control (which *Chunk* to process where for example) and routing, which can both be performed in real time using the heuristic. To demonstrate the potential of serving both types of time dependent data (expedited-data and relaxed-data) and the suitability of this approach for the EEBDN, we presented a CET dimension, i.e., the CPU Execution Time required to process big data *Chunk*s. The assigned processing resources per *Chunk* are optimally allocated for PNs in a manner that satisfies the *Chunk* minimum processing duration requirements.

In addition to the parameters defined in Table 4-1: List of parameters and their definitions.
in Chapter 4, we defined the following parameters in Table 5-1:

Notation	Description
$RCET_{sc}$	Requested CPU execution time of *Chunk c* generated by node *s*.
RPW_{sc}	Requested processing workload for *Chunk c* generated by node *s*.
Φ	Required processing weight for *Chunk*s (W/GHz).

Table 5-1: List of parameters and their definitions.

In addition to the variables defined in Table 4-2 in Chapter 4, we defined the following variables in Table 5-2:

Notation	Description
APW_{spc}	Allocated processing workload of *Chunk c* that is generated by node *s* and processed at node *p*.
$ACET_{spc}$	Allocated CPU execution time of *Chunk c* that is generated by node *s* and processed at node *p*.
PW	Total processing workload consumed by all the *Chunks* in the network including DCs.
T_p	Maximum CPU execution time allocated to process *Chunks* at processing node *p*, $T_p = Max(ACET_{spc})$.
$MAXT$	Maximum CPU execution time needed to process all the *Chunks* in the network. $T = Max(T_p)$.
$MINT$	Minimum CPU execution time needed to process all the *Chunks* in the network. $T = Min(T_p)$.

Table 5-2: List of variables and their definitions.

The model is now updated, defined as follows:

Objective: Minimise

$$PUN \cdot \left(\sum_{i \in N} PR \cdot \ (AR_i + ACH_i + AI_i) + PR \cdot \sum_{j \in N: i \neq j} (C_{ij}) + \sum_{m \in N} \sum_{n \in N_m} PTR \cdot W_{mn} + \sum_{m \in N} \sum_{n \in N_m} PRG \cdot W_{mn} \cdot RG_{mn} + \sum_{m \in N} \sum_{n \in N_m} PE \cdot A_{mn} \cdot F_{mn} + \sum_{i \in N} EO_i \right)$$

$$+ \ PU \cdot \left(\sum_{p \in N} \delta \cdot PNW_p + \sum_{p \in N} \sum_{s \in N} CHT_{sp} \cdot \ (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{d \in N} \left(CHT_{pd} + INT_{pd} \right) \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{d \in N} INF_{pd} \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} SCH_p \cdot RSG \cdot PSG \right)$$

$$-\Phi \cdot \sum_{p \in N} PNW_p. \tag{5-2}$$

Equation (5-2) gives the model objective which maximises the CPU workload per node p and minimises the IP over WDM network, PNs and DCs power consumptions. Φ is a weight that controls the model emphasis on the *Chunks'* allocated CPU workload in the nodes within the fixed nodes' processing capacity. The objective function (equation (5-2)) minimises the network power consumption, minimises the processing power consumption and to different extents, through the parameter Φ, the objective function maximises the amount of processing used such that expedited data can be served quickly when present. For example, if 100% of the data requires expedited-processing, then a high value of Φ is used. Conversely, when all the data requires relaxed-processing, the value of Φ that should be used is low and approaches zero. In this case, the objective function, equation (5-2) simply minimises the overall power consumption made up of network and processing power consumptions. Therefore, there is a trade-off between power saving and the proportion of big data that requires expedited processing.

Subject to:

In addition to constraints (4-10)-(4-30) defined in Chapter 4, the model is subject to the following constraints

$$PNW_p = \sum_{s \in N} \sum_{c \in CH_s} APW_{spc} \tag{5-3}$$

$\forall p \in N,$

$$PW = \sum_{p \in N} PNW_p, \tag{5-4}$$

$$APW_{spc} \geq Y_{spc} \tag{5-5}$$

$\forall s, p \in N, \forall c \in CH_s,$

$$APW_{spc} \leq M.Y_{spc} \tag{5-6}$$

$\forall s, p \in N, \forall c \in CH_s,$

$$APW_{spc} \leq MSW \tag{5-7}$$

$\forall s, p \in N, \forall c \in CH_s$ and

$$\sum_{p \in N} APW_{spc} \geq RPW_{sc} \tag{5-8}$$

$$\forall s \in N, \forall c \in CH_s.$$

Constraint (5-3) replaces constraint (4-16). It calculates each PN's workload (i.e. APW_{spc}) by summing the CPU workload allocated to each individual *Chunk* processed at that PN. Constraint (5-4) calculates the total *Chunks* allocated to the processing workload per node. Constraints (5-5) and (5-6) specify the processing location of *Chunk c* generated by node *s* and processed at node p, where *M* is a large enough unitless number to ensure that $Y_{spc} = 1$ when APW_{spc} is greater than zero. Constraint (5-7) ensures that the processing workload allocated for each *Chunk* does not exceed the maximum processing threshold MSW. Constraint (5-8) ensures that the allocated processing workload for each *Chunk* satisfies the minimum processing workload requested for that *Chunk*. Note that we calculated the $ACET$ in equation (5-9) as follows:

$$ACET_{spc} = \frac{CPI \cdot IC \cdot Y_{spc}}{(APW_{spc}.Y_{spc} + e)} \tag{5-9}$$

where e is a very small number to ensure that $ACET_{spc}$ equals zero when Y_{spc} is zero. Note that equation (5-9) is calculated offline after running the model and obtaining Y_{spc} and APW_{spc}.

5.2 Velocity Model Results

The NSFNET network is also considered for evaluating the impact of velocity on the EEBDN. We consider two velocity scenarios as follows:

5.2.1 Scenario #1: Deterministic volume and RCET per *Chunk*

In this scenario, we assumed that there is only relaxed-data *Chunks* in the network with a PRR per *Chunk* of 0.001 and a volume per *Chunk* of 80 Gb. Each node generates 100 *Chunks* per second and can process locally a different number of *Chunks* depending on the PN's resources capacity.

The Instruction count (IC) per *Chunk* is assumed to be 1 $billion$ instructions. CPUs are used with $CPI = 1$ so that each instruction needs only one clock cycle to be executed, these values approach the values in [131]. $RCET$ is assumed to be one second (i.e., $RPW = 1\ GHz$). However, *Chunks* can have $ACET < RCET$, by optimally selecting $APW > RPW$. Note that each PN has been assigned a random uniformly distributed amount of storage ranging between 10 Pb to 70 Pb. Furthermore, the number of servers per PN is random (uniform distribution) and ranges between 10 and 30 servers. Table 5-3 shows the input values used in this scenario.

Number of *Chunks* per node per second (CH_s)	100
PNs storage capacity (MS_p) $\forall p \in N$	10 Pb - 70 Pb (random uniform)
Number of servers per PN (NS_p) $\forall p \in N$	10-30 (random uniform)
Volume per *Chunk* in Gb ($Chunk_{sc}$)	80
PRR_{sc} per *Chunk*	0.001
Instruction Count (IC)	10^9
CPI	1
Requested CPU execution time $RCET_{sc}$ in seconds	1
Maximum CPU workload allowed for each *Chunk* in GHz MSW	4
Required processing weight for *Chunks* (Φ)	0-1500

Table 5-3: Velocity Scenario #1 parameters.

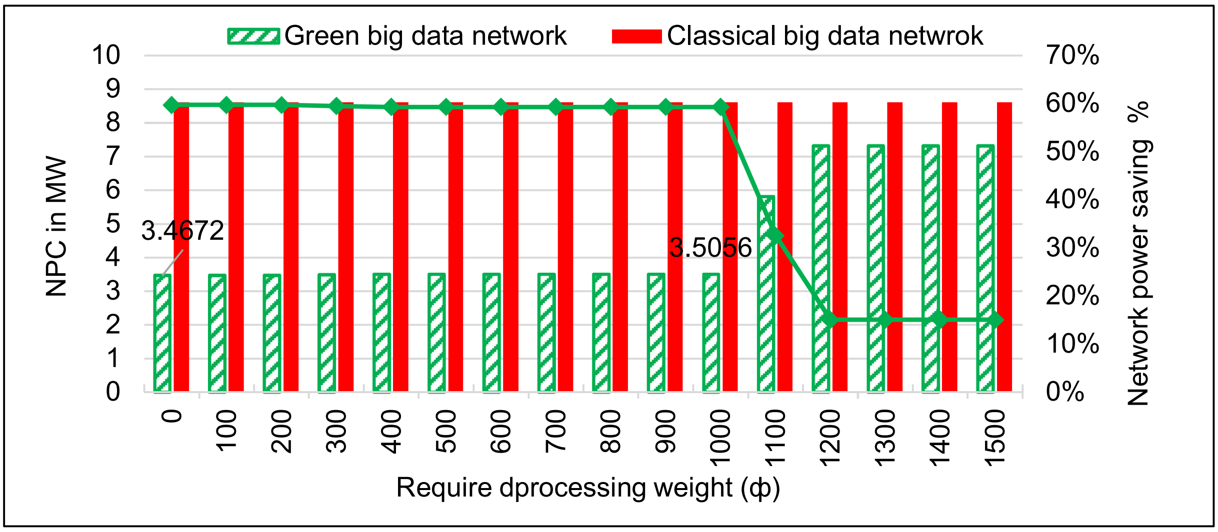
Figure 5-1-a illustrates the energy efficient part of velocity. It shows the relationship between increasing the required processing weight (Φ) and network power consumption. Recall that Φ represents a measure of the degree to which the processing of *Chunks* is expedited, where larger values of Φ correspond to *Chunks* preferring shorter $ACET$. In classical big data networks, the network power consumption remains steady when increasing Φ as all *Chunks* directly traverse to the DCs before processing. In the EEBDN and up to Φ = 1000, a slight increase in power consumption appears in the network (from 3.46 (relaxed-data which is not velocity sensitive) to 3.5 MW as indicated in Figure 5-1-a). This means that a considerable number of *Chunks* are processed locally inside the PNs and the type of network traffic at this point is mostly INF. This results in a 60% power saving compared to the classical approach. At the point where Φ = 1100, however, the effect of Φ becomes evident as the network power consumption increases dramatically and the power saving decreases to 33%. This means that PNs allocated higher CPU processing workloads per *Chunk*, which causes fewer *Chunks* to be locally processed, and more *Chunks* to be forwarded to the optimal DC, i.e., a larger amount of CHT traffic flows in the network. For the same reason, the effect of Φ becomes greater when Φ = 1200 and above, where it causes a maximum level of network power consumption and a minimum power saving of 15%. Therefore, the EEBDN are always better than the classical approach in terms of network power savings even when they serve computationally demanding requests.

Note that the reason of why the power saving remained steady and then started to decrease sharply at Φ=1000 is because the input parameters do not include a variety, i.e. serving only one big data application with deterministic volume, CPU, PRR per *Chunk*. We also evaluated the case where $IC = 2$ billion and kept $RCET = 1$ seconds and $CPI = 1$, hence requiring the system to finish the processing job at the same time interval. The results showed a reduction in network power saving to 32% at Φ = 0 & 800, and 15% at Φ = 1500. This is because CPU workload is proportional to the IC in this case. This leads to higher processing requirements that might exceed PNs processing capacity, hence increasing the central processing at DCs, thereby increasing network power consumption as more CHT flows in the network.

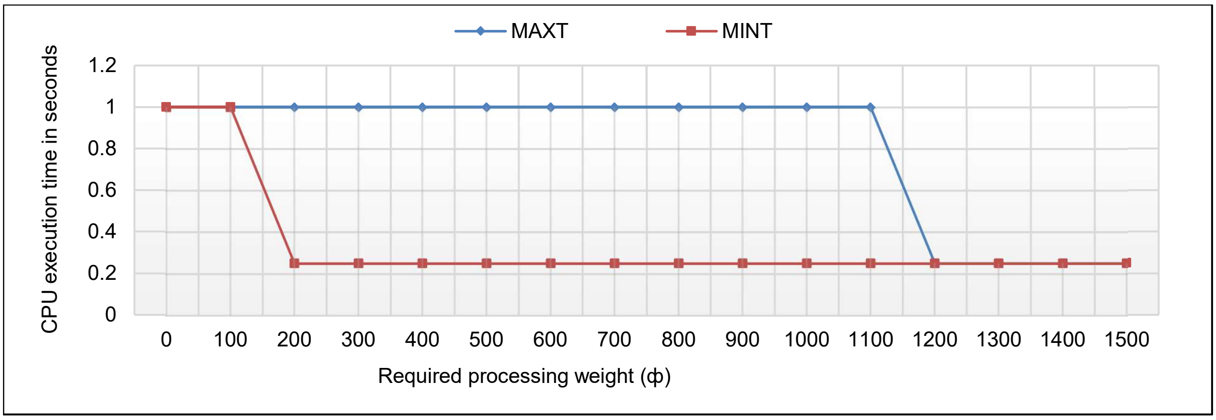
To conclude, Φ and the percentage of the amount of data that requires expedited-processing are proportionally-related. For example, if 100% of data requires expedited-processing, then the value of Φ is high, while this percentage decreases to almost 0% when the value of Φ is low, which means nearly all data requires relaxed-processing. On the other hand, if mixed modes are operated in the network, (i.e., around 50% of data requires relaxed-processing and the other 50% of data requires expedited-processing.), then the value of Φ is moderate.

Figure 5-1-b illustrates the expediting part of velocity. It displays the effect of increasing Φ on the CPU execution time needed to process all the *Chunks* in the network. When the value of Φ is between 0 and 1100, power saving is more important, therefore, allocating minimum number of servers, hence most of the *Chunks* are served in longer time at $MAXT$ of one second, which is the maximum allowed time. Conversely, if these *Chunks* need to be processed in near real time (i.e., large value of Φ), allocating high number of servers is important to have $MAXT$ equal to the minimum allowed CPU execution time of 0.25 second, hence less edge and progressive processing is achieved and more central processing, thereby, reducing the network power saving. This happened at the point where $\Phi \geq 1200$ in our analysis when all *Chunks* are allocated a shorter $ACET$ of 0.25 second ($i.e. APW_{spc} = MSW$). On the other hand, the request for the minimum CPU execution time appears earlier in the DCs at small values Φ, therefore $MINT = 0.25$ for the central processing since there is large enough number of servers inside the DCs. Figure 5-1-c shows the relationship between Φ and the total amount of computational resources (PW) allocated to all

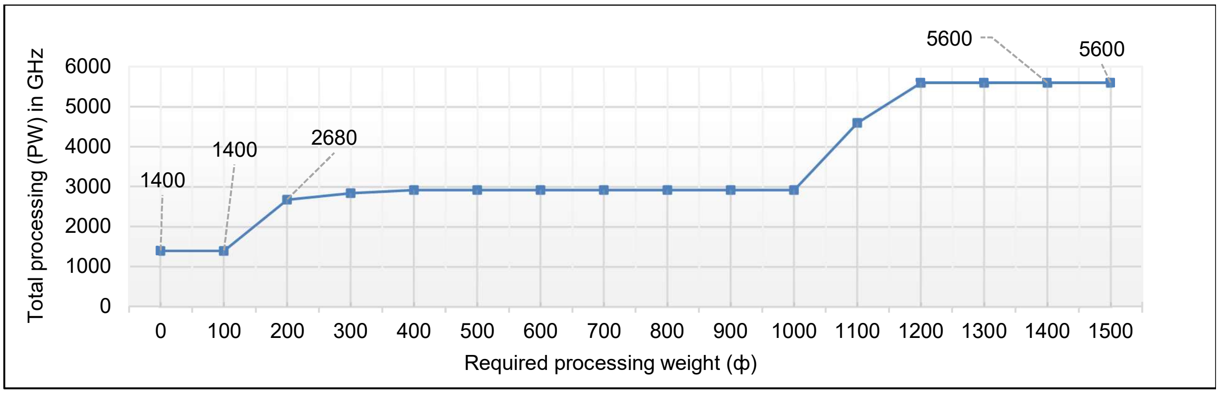
Chunks in the network. All *Chunks* are allocated the minimum CPU workload at $0 \leq \Phi \leq 10$ (i.e. $PW = APW_{spc} \cdot 100\ chunks\ per\ node\ per\ second\ (CH_s) \cdot\ 14\ nodes = 1400\ GHz$). PW increases gradually when increasing Φ until all *Chunks* demand the maximum allowable processing value (MSW) when $\Phi \geq 1200$. Figure 5-1-d shows that the processing resources of all PNs are fully utilised at all values of Φ. At low values of Φ the processing resources of the PNs are fully utilised to serve the largest possible number of *Chunks* to reduce the network power consumption. At high values of Φ, the PN processing resources are also fully utilised by serving a lower number of *Chunks* for a shorter $ACET$. As in previous results, different PNs have different processing capacities, as illustrated in Figure 5-1-d *Chunks* that require processing resources beyond the ability of the PNs are forwarded to the DCs. Therefore, the processing utilisation of the DCs (which are selected optimally at nodes 3 and 14) grows progressively as Φ increases. It is an indication that the DCs are receiving gradually more *Chunks* from the PNs. As a result, the larger the value of Φ, the smaller the number of locally processed *Chunks* inside the PNs, and the higher the number of forwarded *Chunks* to the DCs.



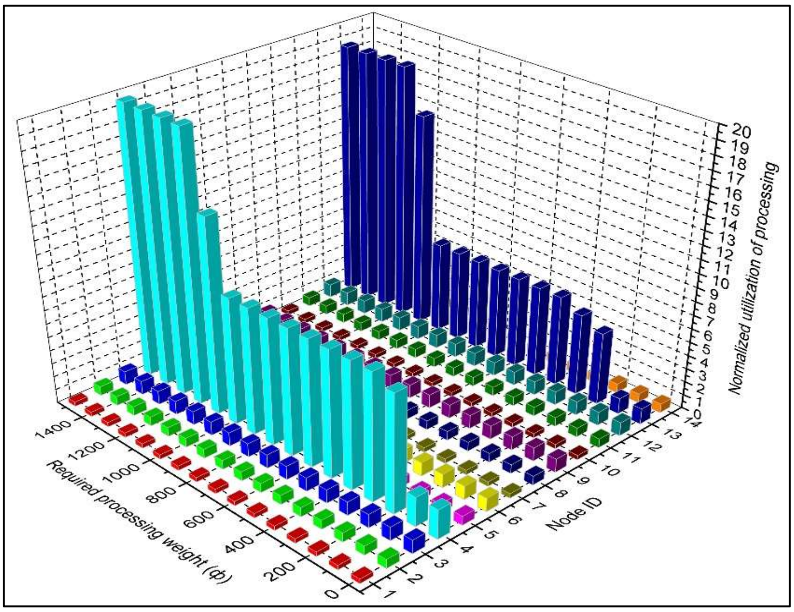
(a)



(b)



(c)



(d)

Figure 5-1: (a) Network power consumption for the CBDN and EEBDN vs ф when CHs=100, for velocity Scenario #1. (b) Max and Min CPU execution time needed to process the *Chunk*s in the network vs Φ when CHs=100, for velocity Scenario #1.

To conclude, using our EEBDN approach, serving *Chunks* in relaxed-data mode results in a 60% network power saving whereas serving *Chunks* in expedited-data mode results in a 15% network power saving.

5.2.2 Scenario #2: Different volume, PRR and RCET per *Chunk*

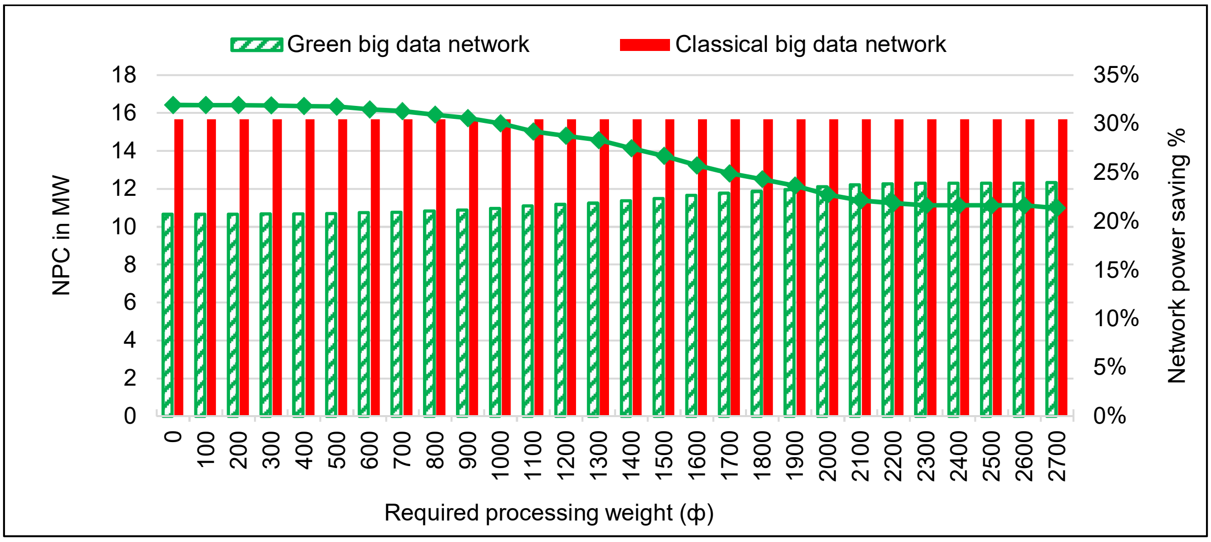
In this scenario, we assumed that each node generates 100 *Chunks* per second with random uniform distribution that ranges between 10 Gb and 330 Gb per *Chunk*. To evaluate a variety of big data applications in the network, the PRR of the *Chunks* is varied using a random uniform distribution between 0.001 and 1. Furthermore, different $RCET$ are assigned for the *Chunks* in a random uniform distribution that varies between the shortest CET of 0.25 seconds to the longest CET of 1 second. *Chunks* with a $RCET = 0.25$ seconds are already within the minimum $ACET$ allowed in our analysis; therefore, changing the value of Φ has no impact on the processing allocation for those *Chunks*. Table 5-4 displays the input values for this scenario.

Number of *Chunks* per node per second (CH_s)	100
PNs storage capacity (MS_p) $\forall p \in N$	10 Pb - 70 Pb (random uniform)
Number of servers per PN (NS_p) $\forall p \in N$	10-30 (random uniform)
Volume per *Chunk* in Gb ($Chunk_{sc}$)	10-330 (random uniform)
PRR_{sc}	0.001-1 (random uniform)
Instruction count (IC)	10^9
CPI	1
Requested CPU execution time ($RCET_{sc}$) in seconds	0.25-1 (random uniform)

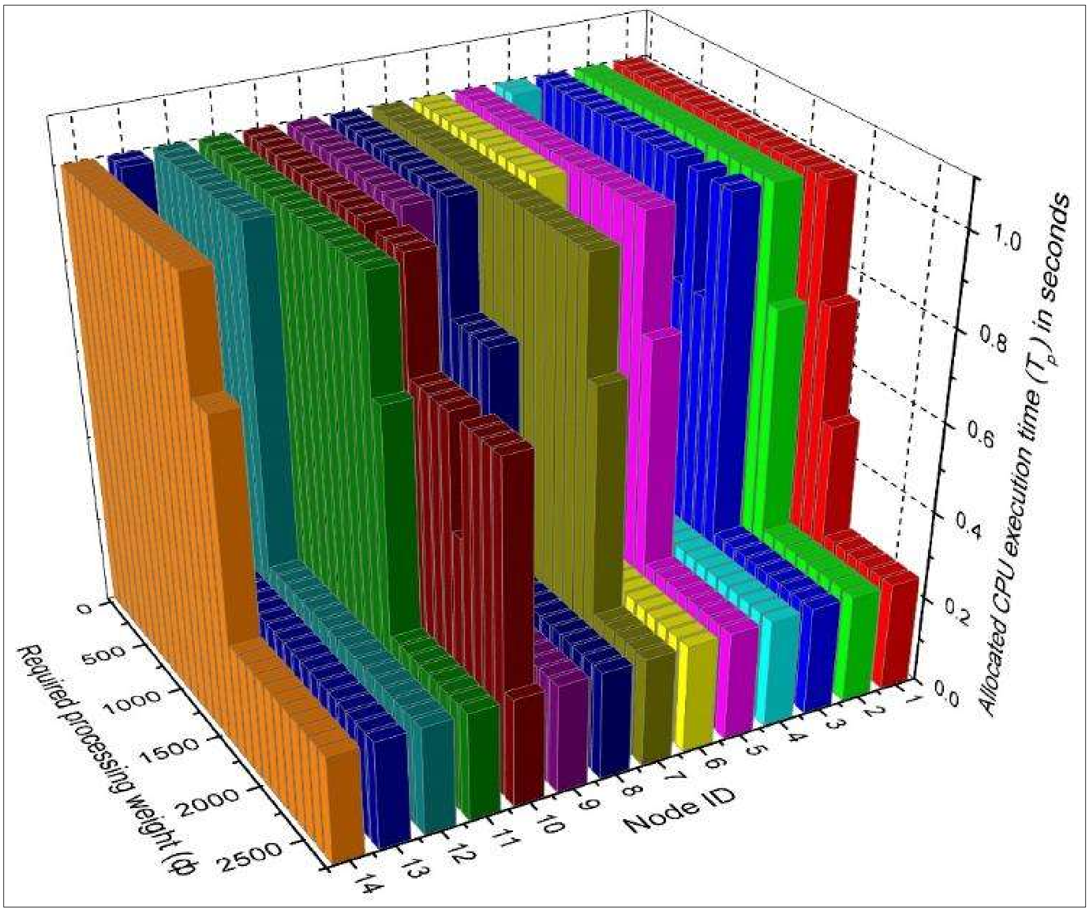
Number of *Chunks* with RCET = 0.25 seconds	450
Number of *Chunks* with RCET > 0.25 seconds	950
Maximum CPU usage allowed for a *Chunk* in GHz (MSW)	4
Required processing weight for *Chunks* (Φ)	0-2700

Table 5-4: Velocity Scenario #2 parameters.

Figure 5-2-a shows a gradual increase in the network power consumption for the EEBDN while it remains constant for the classical approach when applying all the given values of Φ. The interesting point for this trend when compared to the results in the velocity Scenario #1 is the large escalation in the network power consumption at all values of Φ even though the network is serving *Chunks* with larger volumes of up to 330 Gb compared to the 80 Gb *Chunk* size in the velocity Scenario #1. This is because the $RCET$ now is different from one *Chunk* to another, which reflects the diversity in the requested processing workloads (RPW). For instance, there are already 450 *Chunks* in the network that requested the shortest CET by consuming the maximum allowable CPU workload(MSW), whereas the $RCET$ was initially fixed at the longest time of one second for all *Chunks* in Scenario #1, and that initially consumed the lowest RPW values. Therefore, at $0 \leq \Phi \leq 1000$, the maximum power saving decreased to 32% compared to the velocity Scenario #1. The power saving begins to decline gradually until it reaches a minimum level of 21% at $\Phi \geq 2700$, where all the *Chunks* are processed now in expedited-data mode. At this point, the CHT traffic reaches the highest level since every *Chunk* requests the minimum allowed CET value, whereas the allocated CPU workload per *Chunk* (APW) reaches a maximum level. Figure 5-2-b explains the effect of Φ on the T_p for each PN and DC. Nodes 4 and 13 are selected as optimal DCs for handling big data *Chunks* and *Infos*. Moreover, the DCs take the longest T_p of one second for processing *Chunks* at the point where $0 \leq \Phi \leq 100$.



(a)



(b)

Figure 5-2: (a) Network power consumption for the CBDN and EEBDN vs ϕ when CHs=100, for velocity Scenario #2. (b) CPU execution time (Tp) allocated to process *Chunk*s at each PN and each DC vs ϕ when CHs=100, for Scenario #2.

After that point, T_p is at the shortest period of 0.25 seconds, which means that the allocated processing resources for all *Chunks* inside the DCs are at the maximum value. On the other hand, the longest T_p of one second for all PNs is allocated for most of the *Chunks* when $0 \leq \Phi < 1100$. After that point, the impact of Φ on CET is dictated based on the PN processing capacity. That is, the effect of increasing Φ appears and begins first within the PNs with higher processing capacities, such as PN #12 (30 servers) at Φ = 1100, while it affects later PN #11 (20 servers) at Φ = 1900 and PN #7 (with only 10 servers) at Φ = 2200. This is because the optimal approach to minimise network power consumption by decreasing the **CHT** flow is for the PNs with the largest capacity to allocate the shortest $ACET$ for as many *Chunks* as possible and for the PNs with lower processing capacity to allocate longer $ACETs$. The T_p of all the PNs is at the lowest value when Φ > 2600, which means that all the PNs' processing resources are fully utilised with the highest APW values. This leads to fewer locally processed *Chunks* inside the PNs and a greater number of processed *Chunks* inside the DCs.

5.3 Summary

This Chapter introduced a Mixed Integer Linear Programming (MILP) model to investigate the impact of the velocity of big data on the EEBDN in bypass IP over WDM core networks. We used the proposed EEBDN approach by introducing Processing Nodes (PNs) that are attached to the IP over WDM nodes to progressively process big data in the edge, intermediate, and central networks. We served big data in two modes: expedited-data mode and relaxed-data mode. In the first mode, the *Chunk*s are processed quickly by utilising a greater number of computational resources compared to the second mode. The average network power saving was 60% and 15% in the first and second mode, respectively. The reason for the reduction in power saving for the second mode is that more servers are employed to implement less edge and progressive processing, hence smaller number of *Chunk*s can be processed locally in the source PNs and along the route in the intermediate PNs due to the higher CPU workload per *Chunk*.

Chapter 6 Energy Efficient Big Data Networks: Impact of Veracity

The veracity of big data is a more serious challenge to data scientists since they need to distinguish between meaningful data and dirty data [132]. Significant effort is needed to keep dirty data out of organisations' databases. A good reason to motivate big data scientists to analyse the veracity is that, for example, low quality data causes the U.S. economy to waste $3.1 trillion each year [132].

Data cleansing [11] deals with detecting and removing errors and duplications from data to improve its quality. When dealing with multiple big data sources, the need for data cleansing becomes significant since the sources may contain dirty data due to overlaps, duplications or contradictory materials. Therefore, it is important to cleanse data so that it is readied for big data analytics, see Figure 6-1. Hence, providing easy access to accurate, consistent and consolidated data of different data forms is needed [133].

Figure 6-1 illustrates an architectural framework for big data analytics. Pooling data generated from multiple applications and locations is the first phase of big data analytics. In the second phase, the data is in a 'raw' state and needs to be cleansed and readied via several cleansing and transformation options, such as Extract, Transform, Load (ETL) steps [134]. Another approach, which works for the batch processing mode, is data warehousing, wherein data from diverse sources is cleansed, aggregated and made ready for processing [133]. Once the data is cleansed, it should replace the dirty data in the original sources to give legacy applications the improved data. Depending on whether the data is structured or unstructured, various data formats can be input to big data analytics platforms, such as Hadoop [8] and MapReduce [75].

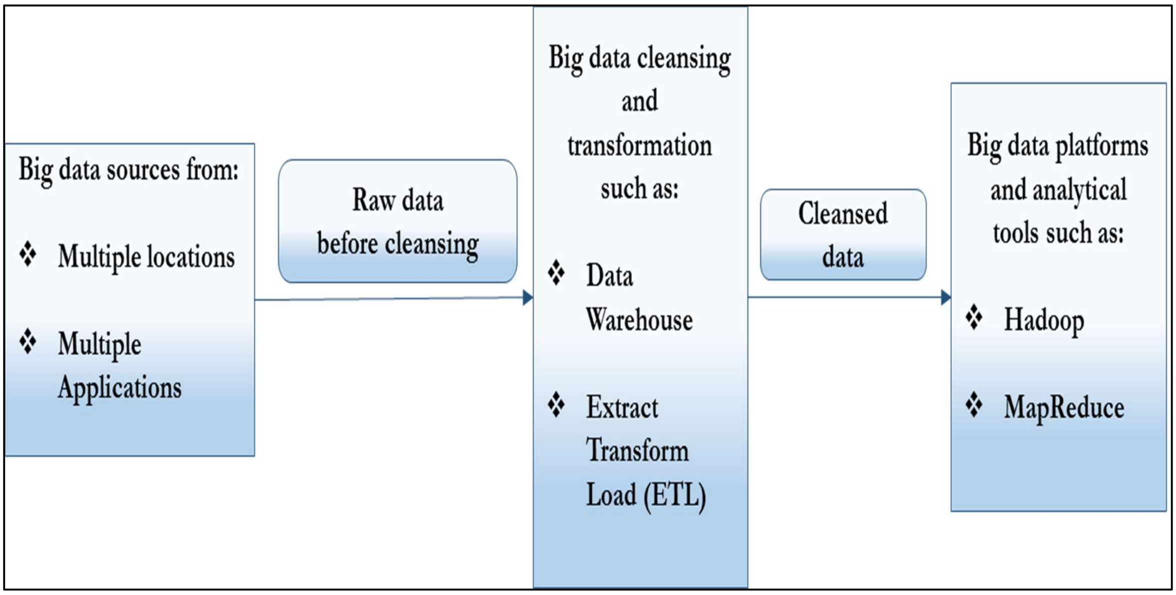


Figure 6-1 Architectural framework for big data analytics [133].

6.1 Veracity MILP Model Description

The objectives of the present section are as follows: (i) to optimise the storage location of cleansed big data *Chunks* before processing, subject to PN storage limitations, and (ii) to build 1 to N (N is the number of nodes in the network) DCs for backup of the cleansed big data *Chunks* to store one copy of each cleansed *Chunk* for the purposes of protection and to recall them for future analytics, and (iii) minimise the power consumption while meeting the first two objectives. The optimisation can find the location of 1 to N such DCs. In the results, we evaluated the case of one backup DC for all the cleansed data, we referred to this DC as a Backup Node (BN).

We extend the model in Chapter 4 to satisfy those objectives. The cleansing process is performed at each SPN when receiving raw data from multiple sources, and generates cleansed *Chunks* with smaller volumes. These cleansed *Chunks* are progressively processed in the EEBDN. Limited storage capacity for the PNs is considered to capture the distinct impact of storage limitations on the optimal location to store the cleansed *Chunks* in the network. We assume that the cleansing phase in the SPNs is implemented using temporary storage shared among the *Chunks*; however, the long-term storage of the cleansed *Chunks* is determined by the model. A cleansed *Chunk* replica is a backup *Chunk* created if the original *Chunk* is lost or destroyed. This backup *Chunk* is optimally stored in the BN. This BN could be either SPN or IPN. However, selecting the location of the BN in the same location of one of the two locations optimised DCs is not considered in our work. The number of employed BNs can be decided according to the level of resilience desired for the big data original *Chunks*. In addition to the parameters defined in Table 4-1 in Chapter 4, we defined the following parameter:

PSB BN storage power per Gigabit (W/Gb).

In addition to the variables defined in Table 4-2 in Chapter 4, we defined the following variables, see Table 6-1.

Notation	**Description**
BN_d	$BN_d = 1$ if node d is a backup node, else $BN_d = 0$.
BCH_{sd}	Backup *Chunks* traffic from source node s to backup node d.
BCH_{ij}^{sd}	Traffic flow of the backup *Chunk* traffic BCH_{sd} between node pair (s, d) traversing virtual link (i, j).
AB_i	Number of aggregation ports in router i utilised by backup *Chunk* traffic BCH_{sd}.
$SBCH_d$	Amount of backup *Chunks* stored in BN d in Gb.

Table 6-1: List of variables and their definitions.

The power consumption of the router ports is calculated as follows:

$$PRPORTS = \sum_{i \in N} PR \cdot (AB_i + AR_i + ACH_i + AI_i) + PR \cdot \sum_{j \in N : i \neq j} (C_{ij}). \tag{6-1}$$

Equation (6-1) replaces equation (4-1). It calculates the power of the router ports when backup big data traffic BCH_{sd} exists in the network.

The power consumption of the internal switches and routers is calculated as follows:

$$PSR = 2 \cdot \sum_{s \in N} \sum_{d \in N} BCH_{sd} \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{s \in N} CHT_{sp} \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{d \in N} \left(CHT_{pd} + INT_{pd}\right) \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{d \in N} INF_{pd} \cdot (RS \cdot SEB + RR \cdot REB). \quad (6\text{-}2)$$

Equation (6-2) replaces equation (4-6). It calculates the power consumption of the internal switches and routers in the SPNs, IPNs and DCs, as well as the extra internal switches' and routers' power consumption in the SPNs and BNs resulting from sending backup *Chunks* between them. As we assumed a homogeneous network and equipment, the total power consumption in the SPNs due to backup *Chunk* traffic is equal to the power consumption of the BN receiving that traffic, hence the factor of two in equation (6-2). The power consumption of the storage for the BN is calculated using equation (6-3).

$$BNSTORAGE = \sum_{d \in N} SBCH_d \cdot RSG \cdot PSB. \quad (6\text{-}3)$$

To assess the impact of the veracity on the EEBDN, we integrated the PNs' storage limitations and cleansed *Chunks* backup dimension with the objective function that optimises the variety. We chose the variety model as it encapsulates the volume analysis and considers a generic data input. The objective of the model is:

Objective: Minimise

Minimise:

$$
\begin{aligned}
PUN \cdot \Bigg(& \sum_{i \in N} PR \cdot \left(AB_i + AR_i + ACH_i + AI_i\right) + PR \cdot \sum_{j \in N: i \neq j} \left(C_{ij}\right) \\
& + \sum_{m \in N} \sum_{n \in N_m} PTR \cdot W_{mn} + \sum_{m \in N} \sum_{n \in N_m} PRG \cdot W_{mn} \cdot RG_{mn} + \sum_{m \in N} \sum_{n \in N_m} PE \\
& \cdot A_{mn} \cdot F_{mn} + \sum_{i \in N} EO_i \Bigg) \\
+ \; PU \cdot \Bigg(& \sum_{p \in N} \delta \cdot PNW_p + 2 \\
& \cdot \sum_{s \in N} \sum_{d \in N} BCH_{sd} \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{s \in N} CHT_{sp} \\
& \cdot \; (RS \cdot SEB + RR \cdot REB) \\
& + \sum_{p \in N} \sum_{d \in N} \left(CHT_{pd} + INT_{pd}\right) \cdot (RS \cdot SEB + RR \cdot REB) \\
& + \sum_{p \in N} \sum_{d \in N} INF_{pd} \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} SCH_p \cdot RSG \cdot PSG \\
& + \sum_{d \in N} SBCH_d \cdot RSG \cdot PSB \Bigg).
\end{aligned}
\tag{6-4}
$$

Equation (6-4) presents the model objective, which is to minimise the IP over WDM network power consumption, the PN power consumption and the BN power consumption. The model objectives are subject to the following constraints in addition to the constraints defined in Chapter 4:

Subject to:

1) BN for big data *Chunks* constraint which is

$$\sum_{d\in N} BCH_{sd} = \sum_{s\in N} \sum_{c\in CH_s} CHV_{sc}. \quad (6\text{-}5)$$

Constraint (6-5) calculates the backup *Chunk* traffic generated at source node *s* and stored at BN *d*. This is done by summing the individual cleansed original *Chunks* generated at source node *s* that are to be stored at node *d*. This constraint ensures that only a single copy of a *Chunk* is stored.

2) Number of BNs and location constraints which are

$$\sum_{s\in N} BCH_{sd} \geq BN_d \quad (6\text{-}6)$$

$$\forall d \in N,$$

$$\sum_{s\in N} BCH_{sd} \leq Z \cdot BN_d \quad (6\text{-}7)$$

$$\forall d \in N,$$

$$BN = \sum_{d\in N} BN_d = 1 \text{ and} \quad (6\text{-}8)$$

$$DC_d \leq 1 - BN_d \quad (6\text{-}9)$$

$$\forall d \in N.$$

Constraints (6-6) and (6-7) build a BN in location *d* if that location is selected to store the backup *Chunks*, where *Z* is a large enough unitless number to ensure that $BN_d = 1$ when $\sum_{s\epsilon N} \mathrm{BCH}_{sd}$ is greater than zero. Constraint (6-8) calculates the total number of backup nodes in the network (we displayed the results when only one backup node is optimally selected in the network). Constraint (6-9) ensures that selecting a node as a DC and BN is not allowed.

3) BN and PNs storage capacity constraint

$$SBCH_d \leq MS_d + H \cdot BN_d \tag{6-10}$$

$$\forall d \in N.$$

Constraint (6-10) ensures that if a PN *d* is chosen to be a BN, then that node has a large enough storage capacity, where *H* is a large enough unitless number, while the PN *d* has a limited storage otherwise.

4) Flow conservation constraints for big data backup *Chunks* traffic which is

$$\sum_{j \in N: i \neq j} BCH_{ij}^{sd} - \sum_{j \in N: i \neq j} BCH_{ij}^{sd} = \begin{cases} BCH_{sd} & m = s \\ -BCH_{sd} & m = d \\ 0 & otherwise \end{cases} \tag{6-11}$$

$$\forall s, i \in N, \forall d \in N: s \neq d.$$

Constraint (6-11) represents the flow conservation constraint for big data backup traffic in the IP layer. This constraint ensures the total outgoing traffic should be equal to the total incoming traffic, except for the source and destination nodes. It can also ensure that the flow can be divided into multiple flow paths in the IP layer.

5) Virtual link capacity constraint

$$\left(\sum_{s \in N} \sum_{d \in N: s \neq d} R_{ij}^{sd} + \sum_{s \in N} \sum_{d \in N: s \neq d} BCH_{ij}^{sd} + \sum_{s \in N} \sum_{p \in N: s \neq p} CHT_{ij}^{sp} \right.$$
$$\left. + \sum_{p \in N} \sum_{d \in N: p \neq d} INF_{ij}^{pd} \right) \leq C_{ij} \cdot B \tag{6-12}$$

$$\forall i, j \in N: i \neq j.$$

Constraint (6-12) replaces constraint (4-24) as defined in Chapter 4. It ensures that the summation of all the traffic types flow through a virtual link and does not exceed its capacity.

6) Number of aggregation ports constraint

$$AB_i = \frac{1}{B} \cdot \sum_{d \in N:\, i \neq d} BCH_{id} \tag{6-13}$$

$$\forall i \in N.$$

Constraint (6-13) calculates the number of aggregation ports for each router that serves the back up traffic BCH_{sd}.

7) Amount of stored backup *Chunks* constraint

$$SBCH_d = \sum_{s \in N} \sum_{c \in CH_s} CHV_{sc} \cdot BN_d \tag{6-14}$$

$$\forall d \in N.$$

Constraint (6-14) represents the size of the backup *Chunks* stored in the BN *d.*

6.2 Veracity Model Results

The NSFNET network is also considered to examine the impact of veracity on network power consumption. In this section, we presented the results of two scenarios depending on the SPNs storage capacity.

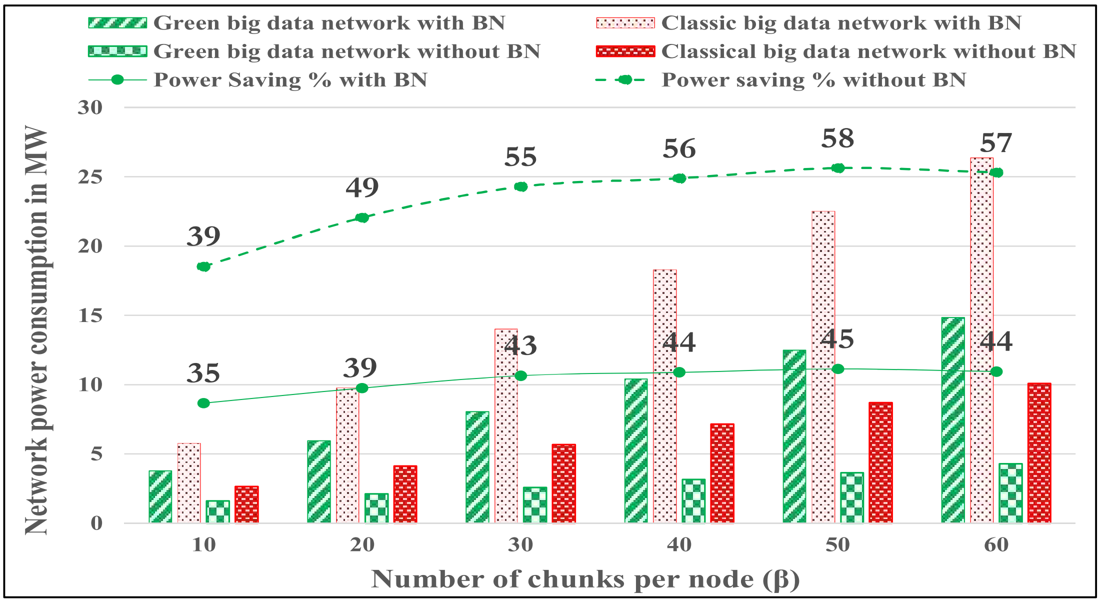
6.2.1 Scenario #1: Veracity with large enough storage capacity

As in previous sections, we compared the EEBDN approach (now with cleansing) to the classical approach where *Chunks* are not cleansed and directly sent to DCs. This means that the raw traffic volumes in the EEBDN are smaller compared to the classical approach where the cleansing and processing happen inside the DCs only. For each approach, we evaluated two modes of operation. In the first mode, there is a BN in the network and in the second mode no BN is employed. Therefore, we compared the two approaches (EEBDN vs CBDN) against each other for each mode. For the EEBDN approach, the cleansed *Chunk* volumes vary in a random uniform distribution between 10 Gb and 220 Gb. On the other hand, a larger volume range for the classical approach is assumed between 10 Gb and 330 Gb. In all cases, the SPNs storage is large enough to store the cleansed data. We used the input values shown in Table 6-2 to examine the influence of veracity on network power consumption.

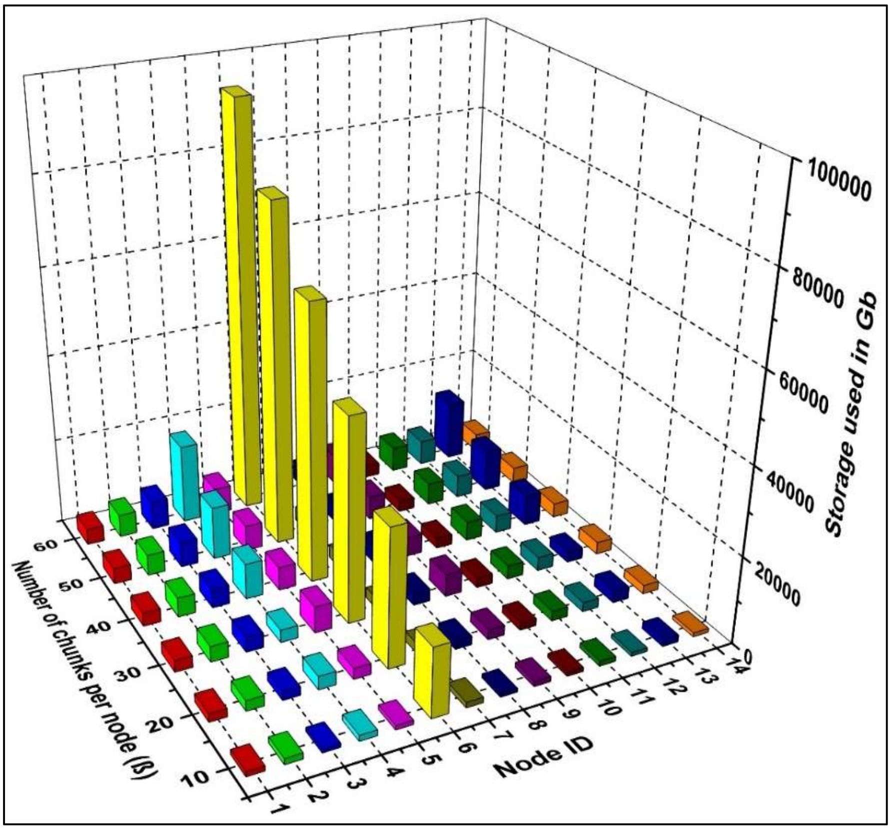
Number of *Chunks* per PN (β)	10-60
Number of servers per PN (NS_p)	10-30 (random uniform)
CPU workload per *Chunk* in GHz (W_{sc})	1-4 (random uniform)
Cleansed *Chunk* volume in Gb ($Chunk_{sc}$)	5-220 (random uniform)
Uncleansed *Chunk* volume in Gb ($Chunk_{sc}$)	10-330 (random uniform)
PRR_{sc}	0.001-1 (random uniform)

Table 6-2. Veracity Scenario #1 parameters.

Figure 6-2-a shows the network power consumption of the classical approach and the EEBDN approach with and without performing the *Chunks* backup modes. The system performance yields noteworthy differences in the network power saving between the two modes. For instance, the maximum power saving is 45% in the backup mode while it is 58% for the no backup mode at β = 50. The average power saving in the backup mode is 41% and 52% in the no-backup mode. The reason for the lower power savings in the backup mode is due to the presence of the extra backup traffic between the SPNs and BNs that increases the network power consumption and reduces the network power savings. On the other hand, there is no backup traffic in the no-backup scenario, but only CHT appears in the network, which is either from SPNs to IPNs or from SPNs to DCs, thereby minimising the power consumption.



(a)



(b)

Figure 6-2 (a). Network power consumption for the CBDN and EEBDN with and without BN for veracity Scenario #1. (b). Storage used in the PNs and DCs and BN with different values of *β* for veracity Scenario #1.

Figure 6-2-b displays the PN, DC and BN storage utilisations with different values of β for the EEBDN approach. It shows that node 6 is selected as the BN at all values of β. This is due to the strategic location of node 6, which has the minimum number of hops to all other nodes. In addition, the DC locations are selected at nodes 4 and 13 for all values of β. Note that up to $\beta = 30$, the DC storage utilisation remains steady as the original *Chunks* are processed either locally in the SPNs or intermediately in the IPNs. At $\beta = 40$ the BCH dominates the network compared to the CHT and INF. At the stage where $40 < \beta \leq 50$, the DCs start to receive a considerable number of original *Chunks* because most PN resources are utilised. Accordingly, CHT increases considerably in addition to the existing BCH, thereby yielding an overall increase in network power consumption as discussed earlier. When $50 < \beta \leq 60$, all PN processing resources are depleted, thus, the increase in the DC storage utilisation is significant as any extra *Chunks* are forwarded to a DC for storing and processing. Consequently, the combined traffic (CHT and BCH) is now at its maximum value at $\beta = 60$.

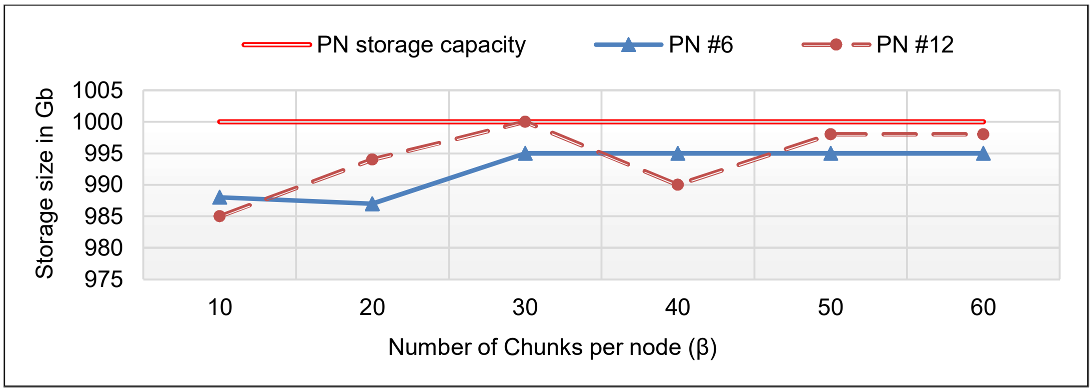
6.2.2 Scenario #2: Veracity with limited storage capacity per PN

In this scenario, we considered the impact of using limited storage in the PNs to allow the model to optimise the location of the cleansed *Chunks* for the EEBDN approach. We reused the same inputs that appeared in Table 6-2 and limited the PNs' storage capacities so they randomly varied between 1 Tb to 4 Tb per PN following a uniform distribution.

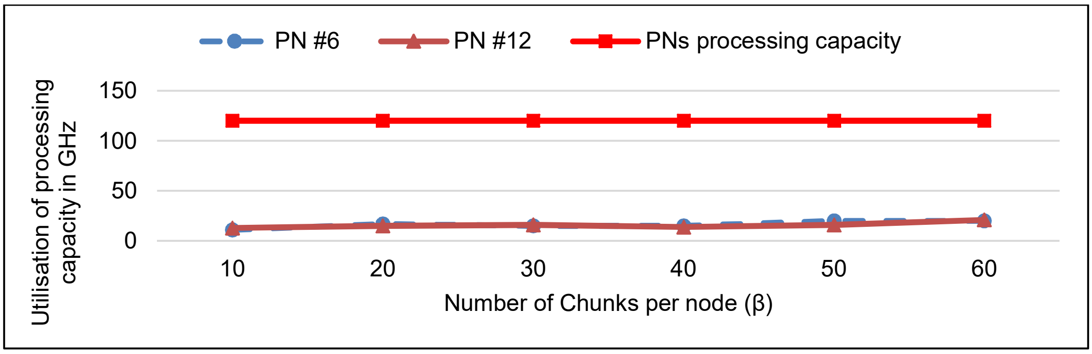
Figure 6-3-a & b displays the PN storage and processing utilisation, respectively. To illustrate the impact of limited storage capacity on the EEBDN with cleansing, we showed the results for two SPNs, #3 and #12. We assigned a low storage capacity of 1Tb to these two SPNs and a high processing capacity of 30 servers. Figure 6-3-a shows that for all values of β, each of the two SPNs could store a maximum cleansed *Chunk* volume ≤ 1 Tb only, and any cleansed *Chunks* above 1 Tb would be optimally forwarded and stored at another IPN or to the DCs for processing. Recall that the SPNs have no cleansing limitations as the cleansing temporary storage is shared among raw *Chunks*. However, the cleansed *Chunks* need to be stored for long term usage at certain PNs, as optimally selected by the model. For example, at $\beta = 60$, the total cleansed volume was 7617 Gb at PN #6 and 6087 Gb at PN #12, however, the actual stored amount of data was 995 Gb inside PN #6 and 998 Gb inside PN #12, due to the 1 Tb storage capacity of both PNs. The remaining cleansed *Chunks* by those two SPNs were optimally sent and stored at one of the DCs. Figure 6-3-b shows the processing utilisation for the two example PNs. Interestingly, for all values of β, the average processing utilisations of both PN #6 and PN #12 were around 16 GHz below the maximum processing capacity (which is 120 GHz: 30 servers with 4 GHz CPU per server). This is because the model skips those PNs after full utilisation of the storage capacity regardless of the availability of processing resources, which leads to a smaller number of locally processed *Chunks*.

Figure 6-3-c illustrates the impact of considering limited PNs storage on network power consumption for the classical approach and the EEBDN approach with and without cleansing of backup *Chunks*. The figure shows a decrease in the network power saving for both backup and no-backup modes compared to the veracity Scenario #1 where the PNs have a large enough storage capacity. The maximum power saving obtained in this scenario declined to 40% for the backup mode and 51% for the no-backup mode at $\beta = 50$, while it was 45% and 58% at $\beta = 50$ with and without backup mode in the veracity Scenario #1, respectively. The reason behind this decrease in power saving is that the limited storage capacity of the PNs leads to a smaller number of cleansed

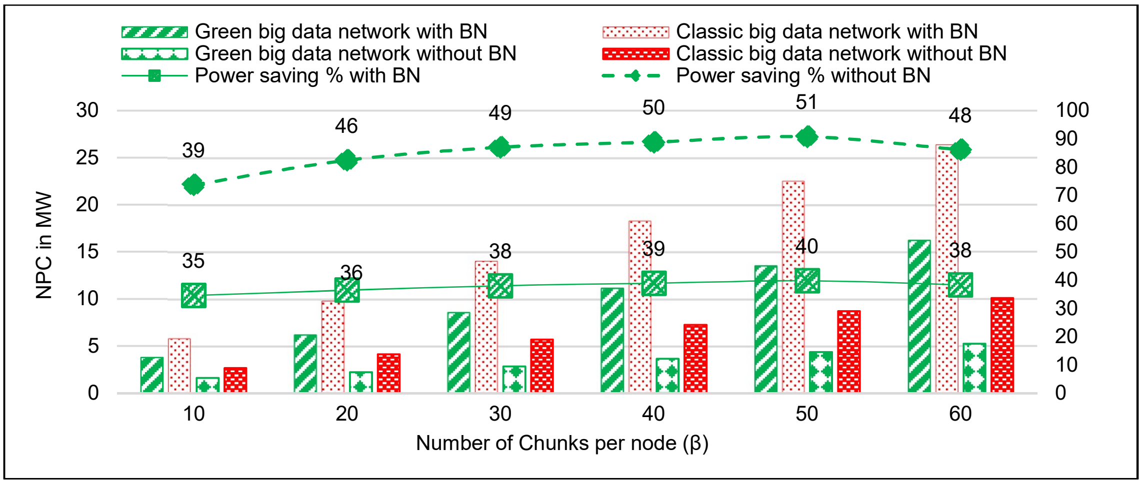
Chunks being processed locally in the edge and progressively in the INPs although there are still available processing resources.



(a)



(b)



(c)

Figure 6-3 (a) PNs storage size with different values of β for veracity Scenario #2. (b) Utilisation of processing capacity for different values of β when considering limited storage per PN for veracity Scenario #2. (c) Network power consumption for the CBDN and EEBDN with and without BN with limited storage per PN for veracity Scenario #2.

Thus, increasing the amount of CHT in the EEBDN results in a higher network power consumption. The maximum network power saving obtained in the present scenario for the backup mode was 40%, while it was 51% for the no-backup mode.

6.3 Summary

This work introduced a Mixed Integer Linear Programming (MILP) model to investigate the impact of the veracity of big data on the EEBDN in bypass IP over WDM core networks. We presented the EEBDN approach by introducing Processing Nodes (PNs) that are attached to the ISP network centres which host the IP over WDM nodes. A PN is a small version of a datacentre (DC) with a capacity that is limited by the available space to build the PN inside the network centre. We introduced a progressive processing technique to serve big data applications in source PNs, intermediate PNs and DCs taking into consideration the veracity dimension. We optimised the storage locations of the cleansed data as well as optimising the location of a single backup node to store a copy of the cleansed *Chunks* for future use. The veracity scenarios had a maximum network power savings of up to 58% in the no backup mode and up to 45% in the backup mode. The lower saving for the no backup mode is due to the movement of *Chunks* from the source PNs to the backup PN without processing them during that journey. In addition, we noted that the veracity scenario under the PNs storage limitations utilises fewer of the available processing resources as it is influenced by the PNs limited storage capacity.

Chapter 7 Conclusions and Future work

7.1 Conclusions

This work introduced Mixed Integer Linear Programming (MILP) models to investigate the impact of the 4Vs of big data on the EEBDN in bypass IP over WDM core networks. Each V poses a number of challenges in the EEBDN. We presented the EEBDN approach by introducing Processing Nodes (PNs) that are attached to the ISP network centres that host the IP over WDM nodes. A PN is a small version of a datacentre (DC) with a capacity that is limited by the cost and avaliable space to build the PN inside the network centre. We introduced a progressive processing technique to serve big data applications in source PNs, intermediate PNs and DCs taking into consideration the volume, variety, velocity and veracity dimensions. The first challenge facing the Data Centres (DCs) is the enormous volume of data fluxing to them. The volume scenarios showed that the network power saving was mainly affected by the volume of the served *Chunks* and its correspoding Processing Reduction Ratio (PRR), which is the ratio of the output after processing *Chunks* to the input. This power saving is higher for larger volumes compared to smaller ones under the same PRR due to the high reduction in the router ports. On the other hand, applying different PRRs has a significant impact on the power saving, i.e., the larger the reduction ratio, the greater the network power saving. We obtained up to 52% and 34% of network power saving in two different volume scenrios, which is a saving compared to the power consumption of the classical networking and processing approach where the *Chunks* are directly forwarded from the source node to the DCs for processing. **Variety** means that there are different types of big data such as CPU intensive, memory intensive, Input/output (IO) intensive, CPU-Memory intensive, CPU/IO intensive, and memory-IO intensive applications. Each requires difference amount of processing, memory, storage, and networking resources. The variety scenarios revealed the impact of serving *Chunks* with different CPU workloads on the power saving since various big data applications require different amounts of computing resources. In view of that, processing *Chunks* that utilise small portions of the CPU help to process as many *Chunk*s as possible inside the local server, hence, reducing the number of unprocessed *Chunks* in the network. We obtained up to 47% and 43 % network power savings in two different variety scenarios. In the velocity section, we served big data in two modes: expedited-data mode and relaxed-data mode. **Expedited-data** processing mode for CPU hungry applications that need to be processed in real time, e.g. remote patient monitoring. **Relaxed-data** processing can tolerate some delay and can be processed in a batch processing mode after being stored inside DCs, such as digital image processing and automated transaction processing. In the first mode, the *Chunks* are processed quickly by utilising a greater number of computational resources compared to the second mode. The average network power saving was 60% and 15% in the first and second mode respectively. The reason for this decline in power saving is that in the second mode more servers are employed to implement less edge and progressive processing, hence smaller number of *Chunks* can be processed locally in the source PNs and along the route in the intermediate PNs due to the higher CPU workload per *Chunk* needed. **Veracity** of big data is a more serious challenge to data scientists since they need to distinguish between the meaningful data and the dirty data. **Veracity** specifies trustworthiness, data protection, data backup, and data cleansing constraints. **Data cleansing** deals with detecting and removing dirty data due to overlaps, errors, duplications, and contradictory materials from big data to improve its quality. It provides easy access to accurate, consistent and consolidated data of different data forms. In the veracity section, we optimised the storage location of the cleansed data as well as optimising the location of a single backup node to store a copy of the cleansed *Chunks* for future use. The veracity scenarios had network power consumption savings of 58% in the no backup mode and 45% in the backup mode due to the movement of *Chunks* from the source PNs to the backup PN without processing them during that journey. In addition, we noted that the veracity scenario under the PNs storage limitations utilises fewer of the available processing resources as it is influenced by the PNs limited storage capacity. The main advantages of the EEBDN can be summarized in the following:

- Achieving considerable saving in the network power consumption even the energy efficiency of the network equipment is enhanced by 315x. Therefore, our model can be applied under different power profiles of the IP over WDM network, PNs, and DCs.
- Reducing the load of big data processing in the DCs.
- Serving expedited big data applications by processing as much applications as possible in the edge so that the processing is achieved in the locations that are closer to the user level. Furthermore, expedited data processing can be prioritized to be served earlier than the delay-tolerant applications.
- The EEBDN algorithm and its MILP model can be generalized and used by other industrial areas and businesses apart from the ICT, such as electric power planet and oil and gas industries.

However, there are several disadvantages that can be tackled in this work such as the overhead big data traffic that might be upscaled when *Inof* size is greater than the *Chunk* size. Therefore, reducing the big data traffic overhead in big data processing using control node with intelligent algorithm is important. Also, the PN that is included in the core network node is not scalable and limited to the available building space of the core network. Moreover, this algorithm is assumed to be use by a single operator in the whole network, having a multi-operator services can, however, enhance the performance of big data processing and build a more flexible and reliable services.

7.2 Future Research Directions

In the context of the energy efficient big data, open research challenges include optimum energy efficient virtualisation, optimum placement of processing nodes, optimum control and management and network processing algorithms. Below we outline some of the research directions that can be pursued to mitigate the impact of big data on the energy consumption of networks.

1. Attaching a metric to each *Chunk* that specifies how many times this *Chunk* will likely be used in the future (frequency). For example, a *Chunk* made up of temperature readings (where the reduction is based on the number of readings above a threshold) may only be used once, as the readings become dated.
2. Attaching a metric that specifies the popularity of *Chunk* where a *Chunk* that is popular is demanded by several other PNs, so there is a PN to PN communications. For example, weather readings where a value of temperature or pressure (extracted) above a certain value is demanded and is useful in several nodes to predict / report future weather trends; another example is the patient set of readings which are confidential, therefore, those readings will likely be of interest to the source node, data centre and doctor node.
3. Our approach can easily be generalised to handle big data bulks that are partially or fully processed at each node, where each bulk contains several *Chunks*. Some applications produce bulks of data *Chunks*. Our study can be generalised to model this scenario. In this case, *Chunks* belonging to a certain big data bulk can be progressively processed in different PNs along different paths and the results can be aggregated to the DCs. This helps perform partial and/or full processing of the bulk (depending on PNs processing capacity). Therefore, it is advantageous to find a "window" of contiguous spare capacity at intermediate nodes. If such a window can be identified, the efficiency improves as each intermediate node processes a bit

more the bulk until one node on the way extracts *Info* from the corresponding *Chunk,* otherwise, the final data centre has to process part or the whole bulk.

4. Clustering can be implemented where SPNs and IPNs form clusters that complement each other in terms of the availability of software packages, e.g. each PN has a different software package.
5. Scheduling can be implemented by introducing storage nodes that have less processing capabilities to store Chunks until a processor of the correct software type is free. Furthermore, the latency analysis plugin is important to provide minimum (best case) and maximum (worst case) latency time for each Chunk in big data processing in the EEBDN.
6. Progressive processing in access, edge and core nodes: Rather than starting big data processing at core nodes, it may be more efficient to start processing at the access and edge layers where much of the big data sources are located. This could be achieved through installing mini processing units along the way at each layer until the core network is reached. Such a scenario can help distribute the processing load of big data and can help extract knowledge gradually and reduce the volume of data handled. The large number of sites where routers, switches and home gateways are placed can be utilised optimally to extract the right *Info* at the right place if such sites are equipped with processing capabilities.
7. Virtualisation: The virtualization of processing and network functions can increase the network energy savings by optimally sharing the big data infrastructure among several operators. For instance, in Figure 3, PNs resources (e.g. servers, storage and internal LAN) and the external core network resources, such as core routers, can be virtualised and controlled at different granularity levels that vary with time according to the jobs specifications. Game theoretical approaches can be beneficially employed here among the different operators to

share their resources as for example some operators might momentarily have specific resource types that are urgently needed by another operator. Virtualisation naturally provides the means to elastically discover and share such available resources.

8. Control and management: Resource optimisation can be done in distributed or centralised fashion. In both cases it is important to employ the correct control plane that targets specific big data jobs requirements. For instance, some big data jobs are more suited to processing under a centralised controller such as batch processing jobs that are aggregated in one location and have flexible execution duration. On the other hand, distributed control planes can facilitate the execution of small size and mission critical data as well as ensuring the service availability. To minimise energy consumption on a global level, it is vital to jointly optimise processing and network control planes, such as SDN, to orchestrate big data *Chunk* analysis and transport in the network.
9. Development of optimum network processing algorithms: The above challenges require the development of efficient algorithms that map the theoretical models' insights into real time applications. Big data energy minimisation algorithms in the network can be built based on the already available experience gained during the last 10 years in green ICT research. However, big data comes with its unique properties and prerequisites that establish upper bounds on the energy savings that can be gained by algorithms working on generic types of traffic. For instance, Geo distributed MapReduce algorithms can be equipped with the capability to implement coding across *Infos* to reduce traffic, hence energy consumption; however, *Info* protection against loss has to be balanced against the energy saving achieved by *Info* coding.

10. Incorporation of renewable energy sources to focus on reducing the CO_2 emissions of backbone IP over WDM networks. A MILP optimization model for hybrid-power" (i.e. renewable and non-renewable energy sources) IP over WDM networks can be set up to minimize the non-renewable energy consumption and CO_2 emission using the EEBDN approach.

References

[1] X. Dong, A. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Energy-efficient core networks," in *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*, 2012, pp. 1-9.

[2] B. G. Bathula and J. M. Elmirghani, "Green networks: Energy efficient design for optical networks," in *Wireless and Optical Communications Networks, 2009. WOCN'09. IFIP International Conference on*, 2009, pp. 1-5.

[3] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Energy efficiency in telecom optical networks," *IEEE Communications Surveys & Tutorials,* vol. 12, 2010, pp. 441-458.

[4] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "IP over WDM networks employing renewable energy sources," *Lightwave Technology, Journal of,* vol. 29, 2011, pp. 3-14.

[5] Internetlivestats, "Internet Live Stats - Internet Usage & Social Media Statistics". [Online]. Available: http://www.internetlivestats.com. [Accessed: 12- Jan- 2017]."

[6] Intel and Tancent, "Tencent Explores Datacenter Resource Pooling Using Intel® Rack Scale Architecture (Intel® RSA)". [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/rsa-tencent-paper.pdf. [Accessed: 01- Dec- 2016]."

[7] E. Schmidt, "Eric Schmidt: Every 2 Days We Create As Much Information As We Did Up To 2003". [Online]. Available: https://techcrunch.com/2010/08/04/schmidt-data/. [Accessed: 19- Jan- 2017]."

[8] P. Zikopoulos and C. Eaton, *Understanding big data: Analytics for enterprise class hadoop and streaming data*: McGraw-Hill Osborne Media, 2011.

[9] T. Wirtz and R. Ge, "Improving mapreduce energy efficiency for computation intensive workloads," in *Green Computing Conference and Workshops (IGCC), International*, 2011, pp. 1-8.

[10] B. d. v. K. Normandeau, variety, velocity and veracity", insideBIGDATA, 2016. [Online]. Available: http://insidebigdata.com/2013/09/12/beyond-volume-variety-velocity-issue-big-data-veracity/. [Accessed: 09- Nov- 2016].

[11] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.,* vol. 23, 2000, pp. 3-13.

[12] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, 2008, pp. 63-74.

[13] A. Hammadi, T. El-Gorashi, and J. Elmirghani, "Energy-Efficient Software-Defined AWGR-Based PON Data Center Network," 2016.

[14] B. Kantarci, L. Foschini, A. Corradi, and H. T. Mouftah, "Inter-and-intra data center VM-placement for energy-efficient large-scale cloud systems," in *2012 IEEE Globecom Workshops*, 2012, pp. 708-713.

[15] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review,* vol. 39, 2008, pp. 68-73.

[16] A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Distributed energy efficient clouds over core networks," *Journal of Lightwave Technology,* vol. 32, 2014, pp. 1261-1281.

[17] L. Nonde, T. E. El-Gorashi, and J. M. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *Journal of Lightwave Technology,* vol. 33, 2015, pp. 1828-1849.

[18] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks," *IEEE Network,* vol. 29, 2015, pp. 36-42.

[19] C. Jayalath, J. Stephen, and P. Eugster, "From the cloud to the atmosphere: running mapreduce across data centers," *Computers, IEEE Transactions on,* vol. 63, 2014, pp. 74-87.

[20] D. Zeng, L. Gu, and S. Guo, "Cost minimization for big data processing in geo-distributed data centers," in *Cloud Networking for Big Data*, ed: Springer, 2015, pp. 59-78.

[21] Y. Ishizuka, W. Chen, and I. Paik, "Workflow Transformation for Real-Time Big Data Processing," in *2016 IEEE International Congress on Big Data (BigData Congress)*, 2016, pp. 315-318.

[22] Q. Xia, W. Liang, and Z. Xu, "Data locality-aware query evaluation for big data analytics in distributed clouds," in *Advanced Cloud and Big Data (CBD), Second International Conference on*, 2014, pp. 1-8.

[23] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating mapreduce performance using workload suites," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*, 2011, pp. 390-399.

[24] Optics.org, *NEC and Corning achieve petabit optical transmission". [Online]. Available: http://optics.org/news/4/1/29. [Accessed: 19- Jan- 2017].*

[25] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical networks: a practical perspective*: Morgan Kaufmann, 2009.

[26] B. Mukherjee, *Optical WDM networks*: Springer Science & Business Media, 2006.

[27] R. C. Dixon, *Spread spectrum systems: with commercial applications* vol. 994: Wiley New York, 1994.

[28] S. Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *IEEE transactions on Communication Technology,* vol. 19, 1971, pp. 628-634.

[29] N. Frigo, P. Iannone, P. Magill, T. Darcie, M. Downs, B. Desai*, et al.*, "A wavelength-division multiplexed passive optical network with cost-shared components," *IEEE Photonics technology letters,* vol. 6, 1994, pp. 1365-1367.

[30] F. Musumeci, M. Tornatore, and A. Pattavina, "A power consumption analysis for IP-over-WDM core network architectures," *Journal of Optical Communications and Networking,* vol. 4, 2012, pp. 108-117.

[31] K. J. Barker, A. Benner, R. Hoare, A. Hoisie, A. K. Jones, D. K. Kerbyson*, et al.*, "On the feasibility of optical circuit switching for high performance computing systems," in *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, 2005, p. 16.

[32] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing*, et al.*, *Integrating microsecond circuit switching into the data center* vol. 43: ACM, 2013.

[33] C. Qiao and M. Yoo, "Optical burst switching (OBS)–a new paradigm for an Optical Internet^{1}," *Journal of high speed networks,* vol. 8, 1999, pp. 69-84.

[34] L. Xu, H. G. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE communications Magazine,* vol. 39, 2001, pp. 136-142.

[35] C. Qiao, "Labeled optical burst switching for IP-over-WDM integration," *IEEE communications Magazine,* vol. 38, 2000, pp. 104-114.

[36] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching: a new area in optical networking research," *IEEE network,* vol. 18, 2004, pp. 16-23.

[37] S. B. Yoo, H. J. Lee, Z. Pan, J. Cao, Z. Yanda, K. Okamoto*, et al.*, "Rapidly switching all-optical packet routing system with optical-label swapping incorporating tunable wavelength conversion and a uniform-loss cyclic frequency AWGR," *IEEE Photonics Technology Letters,* vol. 14, 2002, pp. 1211-1213.

[38] W. I. Way, Y.-M. Lin, and G.-K. Chang, "A novel optical label swapping technique using erasable optical single-sideband subcarrier label," in *Optical Fiber Communication Conference, 2000*, pp. 59-61.

[39] S. B. Yoo, F. Xue, Y. Bansal, J. Taylor, Z. Pan, J. Cao*, et al.*, "High-performance optical-label switching packet routers and smart edge routers for the next-generation Internet," *IEEE Journal on Selected Areas in Communications,* vol. 21, 2003, pp. 1041-1051.

[40] N. Ghani, S. Dixit, and T.-S. Wang, "On IP-over-WDM integration," *IEEE Communications Magazine,* vol. 38, 2000, pp. 72-84.

[41] B. Rajagopalan, J. Luciani, and D. Awduche, "IP over optical networks: A framework," 2004, pp. 2070-1721.

[42] G. Shen and R. S. Tucker, "Energy-minimized design for IP over WDM networks," *Journal of Optical Communications and Networking,* vol. 1, 2009, pp. 176-186.

[43] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Green IP over WDM networks with data centers," *Journal of Lightwave Technology,* vol. 29, 2011, pp. 1861-1880.

[44] A. Wang, J. Zhao, and M. Green, "24% efficient silicon solar cells," *Applied physics letters,* vol. 57, 1990, pp. 602-604.

[45] A. Autenrieth, A. K. Tilwankar, C. M. Machuca, and J.-P. Elbers, "Power consumption analysis of opaque and transparent optical core networks," in *Transparent Optical Networks (ICTON), 2011 13th International Conference on*, 2011, pp. 1-5.

[46] Z. Hasan, H. Boostanimehr, and V. K. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Communications surveys & tutorials,* vol. 13, 2011, pp. 524-540.

[47] X. Dong, T. E. El-Gorashi, and J. M. Elmirghani, "On the energy efficiency of physical topology design for IP over WDM networks," *Journal of Lightwave Technology,* vol. 30, 2012, pp. 1694-1705.

[48] P. P. Biswas, A. Singh, and D. Chadha, "Energy efficient design for Green optical core network," in *Communications (NCC), 2013 National Conference on*, 2013, pp. 1-5.

[49] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, "Energy consumption in optical IP networks," *Journal of Lightwave Technology,* vol. 27, 2009, pp. 2391-2403.

[50] M. Liu, M. Tornatore, and B. Mukherjee, "New strategies for connection protection in mixed-line-rate optical WDM networks," *Journal of Optical Communications and Networking,* vol. 3, 2011, pp. 641-650.

[51] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "BitTorrent Content Distribution in Optical Networks," *Journal of Lightwave Technology,* vol. 32, 2014, pp. 4209-4225.

[52] B. Skubic, A. Lindström, E. I. de Betou, and I. Pappa, "Energy saving potential of cyclic sleep in optical access systems," in *Online Conference on Green Communications (GreenCom), 2011 IEEE*, 2011, pp. 124-127.

[53] C. Cavdar, M. Ruiz, P. Monti, L. Velasco, and L. Wosinska, "Design of green optical networks with signal quality guarantee," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 3025-3030.

[54] GreenTouch, "GreenTouch Final Results from Green Meter Research Study Reducing the Net Energy Consumption in Communications Networks by up to 98% by 2020," *A GreenTouch White Paper,* vol. Version 1, 15 August 2015.

[55] A. Ros, R. Fernández-Pascual, and M. E. Acacio, "Using Heterogeneous Networks to Improve Energy Efficiency in Direct Coherence Protocols for Many-Core CMPs," in *Computer Architecture and High Performance Computing (SBAC-PAD), 2012 IEEE 24th International Symposium on*, 2012, pp. 43-50.

[56] M. Baldi and Y. Ofek, "Time for a" greener" internet," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, 2009, pp. 1-6.

[57] A. A. Saleh and J. M. Simmons, "Evolution toward the next-generation core optical network," *Journal of lightwave Technology,* vol. 24, 2006, p. 3303.

[58] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, 2008, pp. 457-465.

[59] A. Schrijver, *Theory of linear and integer programming*: John Wiley & Sons, 1998.

[60] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems," *Management science,* vol. 27, 1981, pp. 1-18.

[61] D. G. Luenberger and Y. Ye, "The Simplex Method," in *Linear and Nonlinear Programming*, ed: Springer, 2016, pp. 33-82.

[62] G. B. Dantzig and M. N. Thapa, *Linear programming 1: introduction*: Springer Science & Business Media, 2006.

[63] R. H. Bartels and G. H. Golub, "The simplex method of linear programming using LU decomposition," *Communications of the ACM,* vol. 12, 1969, pp. 266-268.

[64] M. Pióro and D. Medhi, *Routing, flow, and capacity design in communication and computer networks*: Elsevier, 2004.

[65] P. C. Zikopoulos, C. Eaton, D. DeRoos, T. Deutsch, and G. Lapis, *Understanding Big Data – Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill: Aspen Institute, 2012.

[66] Facilitiesnet, "Big Data Offers Route To Energy Savings" .[Online]. Available: http://www.facilitiesnet.com/energyefficiency/article/How-Energy-Managers-Can-%20Leverage-Big-Data-Right-Now--13976 .[Accessed: 09- Nov- 2016]."

[67] E. M. A. EMA™, "Operationalizing the Buzz: Big Data 2013.[Online]. Available: https://sandhill.com/wp-content/files_mf/emabigdata2013operationalizingthebuzz.pdf. [Accessed: 19- Jan- 2017]."

[68] Worldmeters, "Worldometers - real time world statistics". [Online]. Available: http://www.worldometers.info/. [Accessed: 09- Nov- 2016]."

[69] IBM, "The Four Vs of Big Data". [Online]. Available: http://www-01.ibm.com/software/data/bigdata/ .[Accessed: 09- Nov- 2016]."

[70] CNET, "Facebook processes more than 500 TB of data daily". https://www.cnet.com/news/facebook-processes-more-than-500-tb-of-data-daily/ .[Accessed: 09- Nov- 2016]."

[71] InternetLiveStats, ".[Online]. Available: http://www.internetlivestats.com/ .[Accessed: 09- Nov- 2016]."

[72] NASA, "What is NASA doing with Big Data today?. [Online]. Available: https://open.nasa.gov/blog/what-is-nasa-doing-with-big-data-today/. [Accessed: 09- Nov- 2016]."

[73] K. Normandeau, "Big data volume, variety, velocity and veracity". [Online]. Available: http://insidebigdata.com/2013/09/12/beyond-volume-variety-velocity-issue-big-data-veracity/. [Accessed: 09- Nov- 2016]."

[74] Dataconomy, "Understanding Big Data: The Seven V's. [Online]. Available: http://dataconomy.com/2014/05/seven-vs-big-data/ .[Accessed: 09- Nov- 2016]."

[75] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM,* vol. 51, 2008, pp. 107-113.

[76] T. White, *Hadoop: The definitive guide*: " O'Reilly Media, Inc.", 2012.

[77] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *ACM SIGOPS operating systems review*, 2003, pp. 29-43.

[78] N. Tiwari, "Scheduling and energy efficiency improvement techniques for Hadoop Map-reduce: State of art and directions for future research," Indian Institute of Technology, Bombay Mumbai, 2012.

[79] B. Hedlund, "Understanding Hadoop Clusters and the Network". [Online]. Available: http://bradhedlund.com/2011/09/10/understanding-hadoop-clusters-and-the-network/ .[Accessed: 09- Nov- 2016]."

[80] Y. W. Zhao, W.-J. van den Heuvel, and X. Ye, "Exploring big data in small forms: A multi-layered knowledge extraction of social networks," in *Big Data, 2013 IEEE International Conference on*, 2013, pp. 60-67.

[81] W. Tan, M. B. Blake, I. Saleh, and S. Dustdar, "Social-network-sourced big data analytics," *IEEE Internet Computing,* vol. 17, 2013, pp. 62-69.

[82] X. Han, L. Tian, M. Yoon, and M. Lee, "A big data model supporting information recommendation in social networks," in *Cloud and Green Computing (CGC), 2012 Second International Conference on*, 2012, pp. 810-813.

[83] Y. Chen, S. Alspaugh, D. Borthakur, and R. Katz, "Energy efficiency for large-scale mapreduce workloads with significant interactive analysis," in *Proceedings of the 7th ACM european conference on Computer Systems*, 2012, pp. 43-56.

[84] P. DeMar, D. Dykstra, G. Garzoglio, P. Mhashikar, A. Rajendran, and W. Wu, "Networking Research Activities at Fermilab for Big Data Analysis," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, 2012, pp. 1398-1399.

[85] Z. U. Shamszaman, S. S. Ara, and I. Chong, "Feasibility considerations of multipath TCP in dealing with big data application," in *Information Networking (ICOIN), 2013 International Conference on*, 2013, pp. 708-713.

[86] J. Liu, S. Byna, and Y. Chen, "Segmented analysis for reducing data movement," in *Big Data, 2013 IEEE International Conference on*, 2013, pp. 344-349.

[87] A. Rajendran, P. Mhashilkar, H. Kim, D. Dykstra, G. Garzoglio, and I. Raicu, "Optimizing large data transfers over 100Gbps wide area networks," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, 2013, pp. 26-33.

[88] P. Bajcsy, A. Vandecreme, J. Amelot, P. Nguyen, J. Chalfoun, and M. Brady, "Terabyte-sized image computations on hadoop cluster platforms," in *Big Data, 2013 IEEE International Conference on*, 2013, pp. 729-737.

[89] K. Lee, L. Liu, Y. Tang, Q. Zhang, and Y. Zhou, "Efficient and Customizable Data Partitioning Framework for Distributed Big RDF Data Processing in the Cloud," in *IEEE CLOUD*, 2013, pp. 327-334.

[90] M. Shtern, B. Simmons, M. Smit, and M. Litoiu, "Toward an ecosystem for precision sharing of segmented big data," in *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, 2013, pp. 335-342.

[91] R. Zhou, M. Liu, and T. Li, "Characterizing the efficiency of data deduplication for big data storage management," in *workload characterization (IISWC), 2013 IEEE International Symposium on*, 2013, pp. 98-108.

[92] I. Satoh, "MapReduce-Based Data Processing on IoT," in *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications*

(GreenCom), IEEE and Cyber, Physical and Social Computing(CPSCom), IEEE, 2014, pp. 161-168.

[93] Y. Wu, Z. Zhang, C. Wu, C. Guo, Z. Li, and F. C. M. Lau, "Orchestrating Bulk Data Transfers across Geo-Distributed Datacenters," *IEEE Transactions on Cloud Computing,* vol. PP, 2015, pp. 1-1.

[94] L. Rupprecht, "Exploiting in-network processing for big data management," in *Proceedings of the 2013 SIGMOD/PODS Ph. D. symposium*, 2013, pp. 1-6.

[95] G. N. Operators, "Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges, & Call for Action," white paper, Oct. 2012.

[96] C. Develder, B. Dhoedt, B. Mukherjee, and P. Demeester, "On dimensioning optical grids and the impact of scheduling," *Photonic Network Communications,* vol. 17, 2009, pp. 255-265.

[97] B. Kantarci and H. T. Mouftah, "Designing an energy-efficient cloud network," *Journal of Optical Communications and Networking,* vol. 4, 2012, pp. B101-B113.

[98] J. Buysse, C. Cavdar, M. De Leenheer, B. Dhoedt, and C. Develder, "Improving energy efficiency in optical cloud networks by exploiting anycast routing," in *Asia Communications and Photonics Conference and Exhibition*, 2011, p. 83100X.

[99] J. Buysse, K. Georgakilas, A. Tzanakaki, M. De Leenheer, B. Dhoedt, and C. Develder, "Energy-efficient resource-provisioning algorithms for optical clouds," *Journal of Optical Communications and Networking,* vol. 5, 2013, pp. 226-239.

[100] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Energy-information transmission tradeoff in green cloud computing," *Carbon,* vol. 100, 2010, p. 200.

[101] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, "Elastic optical networking: A new dawn for the optical layer?," *IEEE Communications Magazine,* vol. 50, 2012.

[102] W. Lu and Z. Zhu, "Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks," *Journal of Lightwave Technology,* vol. 33, 2015, pp. 2078-2086.

[103] L. Zhang and Z. Zhu, "Spectrum-efficient anycast in elastic optical inter-datacenter networks," *Optical Switching and Networking,* vol. 14, 2014, pp. 250-259.

[104] A. M. Al-Salim, A. Q. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy Efficient Tapered Data Networks for Big Data Processing in IP/WDM Networks," in *Transparent Optical Networks (ICTON), 2015 17th International Conference on*, 2015, pp. 1-5.

[105] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012, pp. 750-757.

[106] F. Brian, "Software crisis 2.0," *Computer,* vol. 45, 2012, pp. 89-91.

[107] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the future,* vol. 2007, 2012, pp. 1-16.

[108] W. "Worldometers - real time world statistics", 2016. [Online]. Available: http://www.worldometers.info/. [Accessed: 09- Nov- 2016].

[109] O. "What is NASA doing with Big Data today? | openNASA", 2016. [Online]. Available: https://open.nasa.gov/blog/what-is-nasa-doing-with-big-data-today/. [Accessed: 09- Nov- 2016].

[110] K. Cukier, *Data, data everywhere: A special report on managing information*: Economist Newspaper, 2010.

[111] A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "BitTorrent content distribution in optical networks," *Journal of Lightwave Technology,* vol. 32, 2014, pp. 3607-3623.

[112] H. M. Ali, A. Lawey, T. E. Elgorashi, and J. Elmirghani, "Energy Efficient Resource Provisioning in Disaggregated Data Centres," in *Asia Communications and Photonics Conference*, 2015, p. AM1H. 1.

[113] M. Ali, M. Howraa, A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Energy efficient disaggregated servers for future data centers," in *Networks and Optical Communications-(NOC), 20th European Conference on*, 2015, pp. 1-6.

[114] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems,* vol. 28, 2012, pp. 755-768.

[115] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*: Courier Corporation, 1998.

[116] A. C.-C. Yao, "New algorithms for bin packing," *Journal of the ACM (JACM),* vol. 27, 1980, pp. 207-227.

[117] R. G. Gallager, "Distributed minimum hop algorithms," Massachusetts Inst Of Tech Cambridge Lab For Information And Decision Systems, 1982.

[118] N. B. Rizvandi, J. Taheri, R. Moraveji, and A. Y. Zomaya, "On modelling and prediction of total CPU usage for applications in mapreduce environments," in *Algorithms and Architectures for Parallel Processing*, ed: Springer, 2012, pp. 414-427.

[119] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," *Proceedings of the IEEE,* vol. 99, 2011, pp. 149-167.

[120] C. J. Ter Braak and I. C. Prentice, "A theory of gradient analysis," *Advances in ecological research,* vol. 18, 1988, pp. 271-317.

[121] M. O'mahony, "Results from the COST 239 project. Ultra-high capacity optical transmission networks," in *Optical Communication, 1996. ECOC'96. 22nd European Conference on*, 1996, pp. 11-18.

[122] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications,* vol. 29, 2011, pp. 1765-1775.

[123] E. N. Power, "Energy logic: Reducing data center energy consumption by creating savings that cascade across systems," *White Paper of Emerson Electric Co,* 2009.

[124] W. Chen, J. Yu, W. Zhang, L. Jiang, G. Zhang, and Z. Chai, "Parallel Implementation of Dense Optical Flow Computation on Many-Core Processor," in *International Conference on Algorithms and Architectures for Parallel Processing*, 2015, pp. 119-132.

[125] M. Berezecki, E. Frachtenberg, M. Paleczny, and K. Steele, "Many-core key-value store," in *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011, pp. 1-8.

[126] S. Huang, J. Huang, J. Dai, T. Xie, and B. Huang, "The HiBench benchmark suite: Characterization of the MapReduce-based data analysis," in *New Frontiers in Information and Software as Services*, ed: Springer, 2011, pp. 209-228.

[127] J. M. Elmirghani, T. Klein, K. Hinton, T. El-Gorashi, A. Q. Lawey, and X. Dong, "GreenTouch GreenMeter core network power consumption models and results," in

Green Communications (OnlineGreencomm), 2014 IEEE Online Conference on, 2014, pp. 1-8.

[128] J. Elmirghani, T. Klein, K. Hinton, L. Nonde, A. Lawey, T. El-Gorashi*, et al.*, "GreenTouch GreenMeter core network energy-efficiency improvement measures and optimization," *Journal of Optical Communications and Networking,* vol. 10, 2018, pp. A250-A269.

[129] I. c. "Top tips for securing big data environments | IBM Big Data & Analytics Hub", 2016. [Online]. Available: http://www.ibmbigdatahub.com/blog/top-tips-securing-big-data-environments. [Accessed: 09- Nov- 2016].

[130] D. A. Patterson and J. L. Hennessy, *Computer organization and design: the hardware/software interface*: Newnes, 2013.

[131] A. L. "Processors – ARM ARM | The Architecture for the Digital World", 2016. [Online]. Available: http://www.arm.com/products/processors. [Accessed: 09- Nov- 2016]

[132] A. J. "Dirty Data: One of the Largest Hidden Costs of Doing Business HelpSystems", 2016. [Online]. Available: https://www.helpsystems.com/resources/articles/dirty-data-one-largest-hidden-costs-doing-business. [Accessed: 09- Nov- 2016].

[133] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Information Science and Systems,* vol. 2, 2014, pp. 1-10.

[134] B. Larson, *Delivering Business Intelligence*: New York, 2009.