



UNIVERSITY OF LEEDS

This is a repository copy of *Greening Big Data Networks: Velocity Impact*.

White Rose Research Online URL for this paper:

<http://eprints.whiterose.ac.uk/124447/>

Version: Accepted Version

Article:

Al-Salim, AM, El-Gorashi, TEH, Lawey, AQ et al. (1 more author) (2018) Greening Big Data Networks: Velocity Impact. IET Optoelectronics, 12 (3). pp. 126-135. ISSN 1751-8776

<https://doi.org/10.1049/iet-opt.2016.0165>

© Institution of Engineering and Technology. This paper is a postprint of a paper submitted to and accepted for publication in IET Optoelectronics and is subject to Institution of Engineering and Technology Copyright. The copy of record is available at IET Digital Library. Uploaded in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Greening Big Data Networks: Velocity Impact

Ali M. Al-Salim¹, Taisir E. H. El-Gorashi¹, Ahmed Q. Lawey¹, Jaafar M. H. Elmirghani^{1,2}

¹Institution of Communication and Power Networks (ICaPNet), School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, UK

²Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia
E-mail: elamas@leeds.ac.uk

Abstract: In this article, we investigate the impact of big data's velocity on greening IP over WDM networks. We classify the processing velocity of big data into two modes: expedited-data processing mode and relaxed-data processing mode. Expedited-data demands higher amount of computational resources to reduce the execution time compared to the relaxed-data. We developed a Mixed Integer Linear Programming (MILP) model to progressively process big data's raw traffic of both modes at strategic locations, dubbed processing nodes (PNs), built into the network along the path from the data source to the destination. During the processing of big data, the extracted information from the raw traffic is smaller in volume compared to the original big data traffic each time the data is processed, hence, reducing network power consumption. Our results showed that up to 60% network power saving is achieved when nearly 100% of the data in the network required relaxed-processing. In contrast, only 15% of network power saving is gained when nearly 100% of the data required expedited-processing. We obtained around 33% power saving in the mixed modes (i.e., when approximately 50% of the data is processed in the relaxed-mode and 50% of the data is processed in expedited-mode), compared to the classical approach where no PNs exist in the network and all the processing is achieved inside the centralized datacenters only.

1. Introduction

Velocity is data in motion, which is the speed at which data is fluxing in and processed in the data centers [1]. The flux rate can grow larger for applications collecting information from wide spatial or temporal domains. For instance, the Square Kilometre Array [2] telescope combines signals with a flow speed of 700TB/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 scrutinized in real time to identify potential fraud. Five hundred million daily call detail records are analysed in real-time to predict customer churn faster [3].

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. In [4] the authors study the minimization of overall cost for Big Data placement, processing, and movement across geo-distributed datacenters. In [5], the authors presented an optimization technique to execute a sequence of MapReduce jobs in Geo-distributed DCs to minimize the time and pecuniary cost. The authors in [6] introduced technique to execute MapReduce jobs on multiple IoT nodes to locally process as much data as possible the raw data. The authors in [7] aimed to minimize the communication cost by satisfying as many big data queries as possible over a number of time slots. In-network processing is proposed in [8] to achieve network-awareness to save more bandwidth using custom routing, redundancy elimination and on-path data reduction. In [9], the authors developed a Mixed Integer Linear programming models for energy efficient cloud computing services in IP over WDM core networks.

We developed in [10] and [11] MILP models to investigate the impact of the big data's volume, variety, and veracity on greening big data networks. Our work in [10] and [11] considered big data's volume, variety, and veracity respectively, which are three dimensions of big data that are

not related to the current work. In volume, our work in [10] considered the impact of the volume of the big data and in particular focused in this context on the following: (i) serving a single type of big data application, (ii) evaluating the impact of the power efficiency of PNs, (iii) presenting our new Energy Efficient Big Data Networks (EEBDN) heuristic and its complexity, where EEBDN focuses on greening big data considering its volume dimension, (iv) presenting new results under different network topologies, (v) introducing a software matching problem in big data networks.

In our work in [10] we also considered big data variety. Here we considered serving multiple types of big data applications using the same MILP model presented in volume dimension 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. In the veracity dimension, our work in [11] extended the MILP model of volume and variety to evaluate the impact of big data cleansing and backup operations on the energy efficiency of big data networks. Therefore, we added cleansing and backup requirements and constrains to the volume and variety MILP model. Big data cleansing deals with detecting and removing dirty data due to overlaps, errors, duplications, and contradictory materials from big data to improve its quality and to make it ready for big data analytics. It provides easy access to accurate, consistent and consolidated data of different data forms. The data backup process specifies the optimal location for a backup node to store the cleansed data before entering the big data analytics stage.

This article, however, makes a number of new contributions beyond [10] and [11] as follows: Firstly, we develop 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 a delay sensitive applications is optimized to be as short as possible.

Secondly, we extended the objective of the MILP model so that it minimizes the network power consumption as well as minimizing 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 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 Source PNs (SPNs), at the Intermediate PNs (IPNs) or inside the centralized datacenters (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 reside in optimizing the processing locations of chunks, optimizing the locations of the DCs, ensuring the flow conservation of big data traffic, and minimizing 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.

1.1. Classical Big Data Networks vs. Green Big Data Networks

Classically, all big data Chunks traffic (CHT) generated by the source nodes is forwarded to the DCs to be processed there as shown in Figure 1-a. On the other hand, in the green big data network, shown in Figure 1-b, the PNs are attached to the core nodes of the IP over WDM network. Each PN is composed of internal switches and routers, limited storage, and limited number of servers depending on the available building space and its structure is similar to the cloud structure in [9]. DCs, however, are assumed to have large enough processing and storage capabilities. PNs are capable of processing different number of Chunks to extract the corresponding knowledge with small data size from each Chunk. We refer to such knowledge as (Info). These Infos' traffic (INF) is forwarded to location optimized DCs. When a given SPN is not capable of processing all its own chunks, it forwards these chunks to the nearest IPN through energy efficient routes to be processed there, hence, the amount of big data traffic is reduced significantly during the Chunks journey from the source to the DC.

Typically, the size of Info is very small compared to the Chunks [12] in many big data applications such as remote patient monitoring to capture only the abnormality in the heartbeat from huge amount of measured heartbeat rate time services. In equation (1), we introduce the relation between the size of Infos and Chunks as a ratio, termed as Processing Reduction Ratio (PRR). For instance, Chunk of 100 gigabit (Gb) and PRR of 0.001 results in Info of 0.1 Gb.

$$Volume\ of\ Info = PRR \times Volume\ of\ Chunk. \quad (1)$$

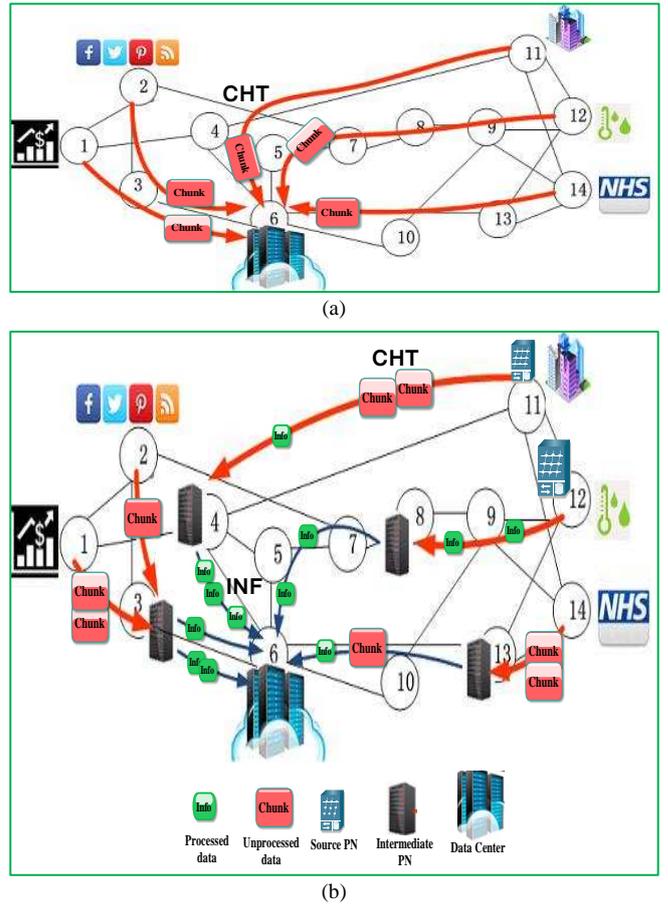


Figure 1. (a) Classical big data network. (b) Green big data network [13].

2. Velocity Impact on Greening Big Data Networks

The intention of the present section 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 the less busy hours, hence, gaining a higher overall rate of utilization. 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 maximize its value while fluxing into the DCs. For instance, sometimes a two minute delay is too much to catch fraud or it could lead to a disaster, such as the situation in 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 CPU workload will optimally be a large portion of the CPU processing capacity, so as to be served quickly. Therefore, increasing the CPU frequency will have a positive impact on decreasing the

The MILP variables for the green big data networks are defined in Table 2.

Table 2 MILP model variables definition	
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. $T_p = \text{Max}(ACET_{spc})$.
$MAXT$	Maximum CPU execution time needed to process all the Chunks in the network. $T = \text{Max}(T_p)$.
$MINT$	Minimum CPU execution time needed to process all the Chunks in the network. $T = \text{Min}(T_p)$.
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 utilized 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 utilized by big data Info traffic INF_{pd} .
F_{mn}	Number of fibres 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 $DC_d = 0$.

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
$$\sum_{i \in N} PR \cdot (AR_i + ACH_i + AI_i) + PR \cdot \sum_{j \in N: i \neq j} (C_{ij}). \quad (3)$$

- 2) The power consumption of transponders
$$\sum_{m \in N} \sum_{n \in N_m} PTR \cdot W_{mn}. \quad (4)$$

- 3) The power consumption of regenerators is
$$\sum_{m \in N} \sum_{n \in N_m} PRG \cdot W_{mn} \cdot RG_{mn}. \quad (5)$$

- 4) The power consumption of EDFAs
$$\sum_{m \in N} \sum_{n \in N_m} PE \cdot A_{mn} \cdot F_{mn}. \quad (6)$$

- 5) The power consumption of optical switches
$$\sum_{i \in N} PO_i. \quad (7)$$

Equation (3) 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) and (5) evaluate the power consumption of all the transponders and regenerators in the optical layer. Equation (6) evaluates the total power consumption of the EDFAs in the optical layer. Equation (7) 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

$$PSR = \sum_{p \in N} \sum_{s \in N} CHT_{sp} \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{d \in N} (CHT_{pd} + INT_{pd}) \cdot (RS \cdot SEB + RR \cdot REB) + \sum_{p \in N} \sum_{d \in N} INF_{pd} \cdot (RS \cdot SEB + RR \cdot REB). \quad (8)$$

Equation (8) 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 servers

$$\sum_{p \in N} \delta \cdot PNW_p. \quad (9)$$

Although the server power consumption is a function of the idle power, maximum power and CPU utilization [17], we consider only $\delta = SMP/MSW$ in equation (9) to calculate its power consumption. This yields a close approximation (when a large number of servers is considered) even when there is idle power in each server. The difference is only in the last powered on server. Note that in the PN and DC servers, each server in our case is either fully utilized or is off.

- 3) The power consumption of the storage

$$\sum_{p \in N} SCH_p \cdot RSG \cdot PSG. \quad (10)$$

Equation (10) represents the storage power consumption of node p. We performed the analysis by considering a network architecture where $RSG = 1$.

The model is defined as follows

Objective: Minimize

$$\begin{aligned}
& PUN \cdot \left(\sum_{i \in N} PR \cdot (AR_i + ACH_i + AI_i) + PR \right. \\
& \quad \cdot \sum_{j \in N: i \neq j} (C_{ij}) \\
& \quad + \sum_{m \in N} \sum_{n \in N_m} PTR \cdot W_{mn} \\
& \quad + \sum_{m \in N} \sum_{n \in N_m} PRG \cdot W_{mn} \cdot RG_{mn} \\
& \quad + \sum_{m \in N} \sum_{n \in N_m} PE \cdot A_{mn} \cdot F_{mn} \\
& \quad \left. + \sum_{i \in N} EO_i \right) \\
& + PU \cdot \left(\sum_{p \in N} \delta \cdot PNW_p \right. \\
& \quad + \sum_{p \in N} \sum_{s \in N} CHT_{sp} \\
& \quad \cdot (RS \cdot SEB + RR \cdot REB) \\
& \quad + \sum_{p \in N} \sum_{d \in N} (CHT_{pd} + INT_{pd}) \\
& \quad \cdot (RS \cdot SEB + RR \cdot REB) \\
& \quad + \sum_{p \in N} \sum_{d \in N} INF_{pd} \\
& \quad \cdot (RS \cdot SEB + RR \cdot REB) \\
& \quad \left. + \sum_{p \in N} SCH_p \cdot RSG \cdot PSG \right) \\
& - \Phi \cdot \sum_{p \in N} PNW_p. \tag{11}
\end{aligned}$$

Equation (11) gives the model objective that maximizes 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 (11)) 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 (11) 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:

PNs and DCs Constraints:

1) Processing counter of big data Chunks constraint

$$\sum_{p \in N} Y_{spc} = 1 \tag{12}$$

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

Constraint (12) ensures that a Chunk c generated by PN s is processed by no more than one PN p .

2) Big data Chunks traffic constraint

$$\begin{aligned}
CHT_{sp} &= \sum_{c \in CH_s} CHV_{sc} \cdot Y_{spc} \\
&\forall s, p \in N. \tag{13}
\end{aligned}$$

Constraint (13) calculates the big data Chunks traffic generated at source node s and directed to node p . This traffic is generated by transmitting Chunk_{sc} from node s to node p in one second.

3) Aggregated processed big data traffic constraint

$$\begin{aligned}
\sum_{d \in N} INF_{pd} &= \sum_{s \in N} \sum_{c \in CH_s} CHV_{sc} \cdot Y_{spc} \cdot PRR_{sc} \\
&\forall p \in N. \tag{14}
\end{aligned}$$

Constraint (14) represents the aggregated big data info traffic INF_{pd} generated by PN p and destined to DC d .

4) Number and locations of DCs constraints

$$\sum_{p \in N} INT_{pd} \geq DC_d \tag{15}$$

$$\begin{aligned}
&\forall d \in N, \\
&\sum_{p \in N} INT_{pd} \leq Z \cdot DC_d \tag{16}
\end{aligned}$$

$$\begin{aligned}
&\forall d \in N, \text{ and} \\
&DCN = \sum_{d \in N} DC_d. \tag{17}
\end{aligned}$$

Constraints (15) and (16) 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 $\sum_{p \in N} INT_{pd}$ is greater than zero. Constraint (17) limits the total number of built DCs to DCN.

5) PNs and DCs storage constraints

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

$$\begin{aligned}
SCH_p &\leq MS_p + (H \cdot DC_p) \\
&\forall p \in N. \tag{19}
\end{aligned}$$

Constraint (18) represents the size of Chunks in Gb stored in PN p . Constraint (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.

6) PNs and DCs internal switches and routers constraint

$$\begin{aligned}
\sum_{s \in N} CHT_{sp} &\leq MSR_p + (A \cdot DC_p) \\
&\forall p \in N. \tag{20}
\end{aligned}$$

Constraint (20) ensures that the total amount of big data traffic between the PNs will 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.

The IP over WDM Network Constraints:

1) Flow conservation constraints for the regular traffic

$$\sum_{j \in N: i \neq j} R_{ij}^{sd} - \sum_{j \in N: i \neq j} R_{ji}^{sd} = \begin{cases} R_{sd} & i = s \\ -R_{sd} & i = d \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

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

2) Flow conservation constraints for the big data Chunks traffic

$$\sum_{j \in N: i \neq j} CHT_{ij}^{sp} - \sum_{j \in N: i \neq j} CHT_{ji}^{sp} = \begin{cases} CHT_{sp} & i = s \\ -CHT_{sp} & i = p \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

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

3) Flow conservation constraints for the big data Info traffic

$$\sum_{j \in N: i \neq j} INF_{ij}^{pd} - \sum_{j \in N: i \neq j} INF_{ji}^{pd} = \begin{cases} INF_{pd} & i = p \\ -INF_{pd} & i = d \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

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

Constraints (21-23) represent the flow conservation constraints R_{sd} , CHT_{sp} and INF_{pd} traffic 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.

4) Virtual link capacity constraint

$$\left(\sum_{s \in N} \sum_{d \in N: s \neq d} R_{ij}^{sd} + \sum_{s \in N} \sum_{p \in N: s \neq p} CHT_{ij}^{sp} + \sum_{p \in N} \sum_{d \in N: p \neq d} INF_{ij}^{pd} \right) \leq C_{ij} \cdot B \quad (24)$$

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

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

5) Optical layer flow conservation constraints:

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

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

Constraint (25) represents the flow conservation constraints in the optical layer. It assumes 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.

6) Physical link capacity constraints

$$\sum_{i \in N} \sum_{j \in N: i \neq j} W_{mn}^{ij} \leq WL \cdot F_{mn}. \quad (26)$$

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

Constraint (26) ensures that the summation of the wavelengths in a virtual link traversing a physical link do not exceed the capacity of the fibre in the physical link.

7) Wavelengths capacity constraint

$$\sum_{i \in N} \sum_{j \in N: i \neq j} W_{mn}^{ij} = W_{mn} \quad (27)$$

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

Constraint (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 utilized by regular traffic constraint

$$AR_i = \frac{1}{B} \cdot \sum_{d \in N: i \neq d} R_{id} \quad (28)$$

$\forall i \in N.$

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

$$ACH_i = \frac{1}{B} \cdot \sum_{\substack{p \in N: i \neq p \\ \forall i \in N}} CHT_{ip} \quad (29)$$

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

$$AI_i = \frac{1}{B} \cdot \sum_{\substack{d \in N: i \neq p \\ \forall i \in N}} INF_{id} \quad (30)$$

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

PNs workload constrains:

$$PNW_p = \sum_{s \in N} \sum_{c \in CH_s} APW_{spc} \quad (31)$$

$\forall p \in N,$

$$PW = \sum_{p \in N} PNW_p, \quad (32)$$

$$APW_{spc} \geq Y_{spc} \quad (33)$$

$$\forall s, p \in N, \forall c \in CH_s, \quad (34)$$

$$APW_{spc} \leq M \cdot Y_{spc} \quad (34)$$

$$\forall s, p \in N, \forall c \in CH_s, \quad (35)$$

$$APW_{spc} \leq MSW \quad (35)$$

$$\forall s, p \in N, \forall c \in CH_s \text{ and} \quad (36)$$

$$\sum_{p \in N} APW_{spc} \geq RPW_{sc} \quad (36)$$

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

Constraint (31) calculates each PN's workload by summing the CPU workload allocated to each individual Chunk processed at that PN. Constraint (32) calculates the total Chunks allocated to the processing workload per node. Constraints (33) and (34) specify the processing location of Chunk c generated by node s and processed at node p , where M is a large enough unit-less number to ensure that $Y_{spc} = 1$ when APW_{spc} is greater than zero. Constraint (35) ensures that the processing workload allocated for each Chunk does not exceed the maximum processing threshold MSW . Constraint (36) ensures that the allocated processing workload for each Chunk satisfies the minimum processing workload requested for that Chunk.

Note that we calculate the $ACET$ in equation (37) as follows:

$$ACET_{spc} = \frac{CPI \cdot IC \cdot Y_{spc}}{(APW_{spc} \cdot Y_{spc} + e)} \quad (37)$$

where e is a very small number to ensure that $ACET_{spc}$ equals zero when Y_{spc} is zero. Note that equation (39) is calculated offline after running the model and obtaining Y_{spc} and APW_{spc} .

2.2. Results of Volume Scenarios

Our MILP model was evaluated using the NSFNET network depicted in Figure 2. Note that we used processor cycles in GHz as a measure of the total processing capability of a node [18]. Table 3 summarizes the input parameters to the model.

Table 3 Input data for velocity model.

PNs storage capacity (MS_p) $\forall p \in N$	10 Pb - 70 Pb (random uniform)
Number of servers per PN (NS_p) $\forall p \in N$	5-30 (random uniform)
Server CPU capacity in GHz (MSW)	4 GHz
Max server power consumption (MSP)	300 W [9]
Energy per bit of the PN and DCs switch (SEB)	11.875 W/Gbps [9]
Energy per bit of the PN and DCs router (REB)	7.727 W/Gbps [9]
Storage power consumption (PSG)	0.008 W/Gb [9]
IP over WDM router power consumption (PR)	825 W [19]
IP over WDM regenerator power consumption (PRG)	334 W [19]
IP over WDM transponder power consumption (PTR)	167 W [19]
IP over WDM optical switch power consumption (PO_i) $\forall i \in N$	85 W [19]
IP over WDM EFDA power consumption (PE)	55 W [19]
Wavelength bit rate (B)	40 Gbps
Span distance between EDFAs (S)	80 km
Number of wavelengths per fibre (WL)	32
Number of location optimized DCs (DCN)	2
IP over WDM power usage effectiveness (PUN)	1.5 [9]
PNs and DCs power usage effectiveness (PU)	2.5 [9]

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 5 and 30 servers. The MILP in this section is used to evaluate the proposed big data networks. In addition, the same model can be used to evaluate the classical approach by introducing a constraint that prevents the processing of big data outside the DCs.

We evaluate the proposed work in several volume scenarios as follows:

2.2.1 Deterministic Volume and RCET per Chunk

In this scenario, we assume that there is only relaxed-data Chunks in the network with a PRR per Chunk of 0.01 and a volume per Chunk of 50 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 [20]. $RCET$ is assumed to be one second (i.e., $RPW = 1GHz$). However, Chunks are allowed to have $ACET < RCET$, by optimally selecting $APW > RPW$. Table 4 shows the input values used in this scenario.

Table 4 Velocity scenario 2.2.1 parameters.

Number of Chunks per node per second (CH_s)	100
Volume per Chunk in Gb ($Chunk_{sc}$)	50
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

Figure 3-a illustrates the green 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 green big data networks and up to $\Phi = 1000$, a slight increase in power consumption appears in the network (from 2.16 (relaxed-data which is not velocity sensitive) to 2.19 MW as indicated in Figure 3-a). This means that a considerable number of Chunks are processed locally inside the PN 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 $\Phi = 1100$, however, the effect of Φ becomes evident as the network power consumption increases dramatically and the power saving decreases to 33%. This means that PN have 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 $\Phi = 1200$ and above, where it causes a maximum level of network power consumption and a minimum power saving of 15%. Therefore, the green big data networks are always better than the classical approach in terms of network power savings even when they serve computationally demanding requests.

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 show a reduction in network power saving to 32% at $\Phi = 0$ & 800, and 15% at $\Phi = 1500$. This is because CPU workload is proportional to the IC in this case. This leads to higher processing requirements that might exceed PN's processing capacity, hence increasing the central processing at DCs, thereby increasing network power consumption as more **CHT** will flow in the network.

To conclude, the ratio between Φ and percentage of the amount of data requires expedited-processing is proportionally-related. For example, if 100% of data require 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 a 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 3-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 happens 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 3-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. $PW = APW_{spc} \cdot 100 \text{ chunks per node per second } (CH_s) \cdot 14 \text{ nodes} = 1400 \text{ GHz}$). PW increases gradually when increasing Φ until all Chunks in overall network demand the maximum allowable processing value (MSW) when $\Phi \geq 1200$.

Figure 3-d illustrates that the processing resources of all PNs are fully utilized at all values of Φ . At low values of Φ the processing resources of the PNs are fully utilized 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 utilized 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 3-d Chunks that require processing resources beyond the ability of the PNs are forwarded to the DCs. This is why the processing utilization 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.

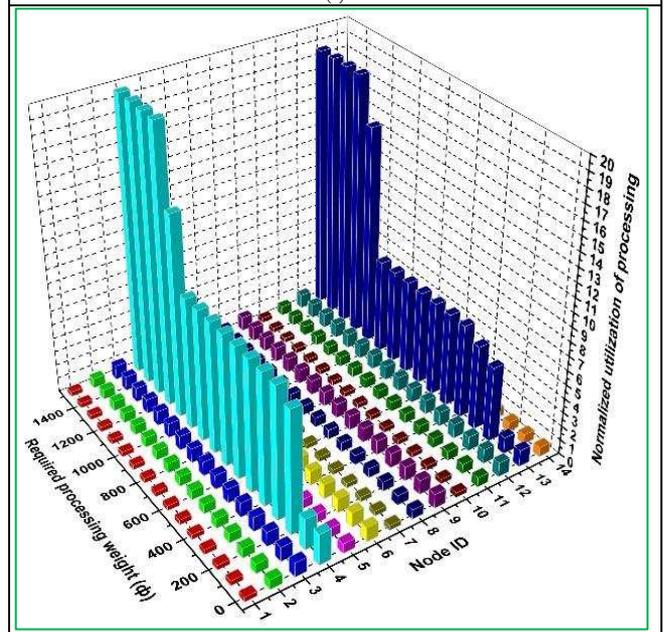
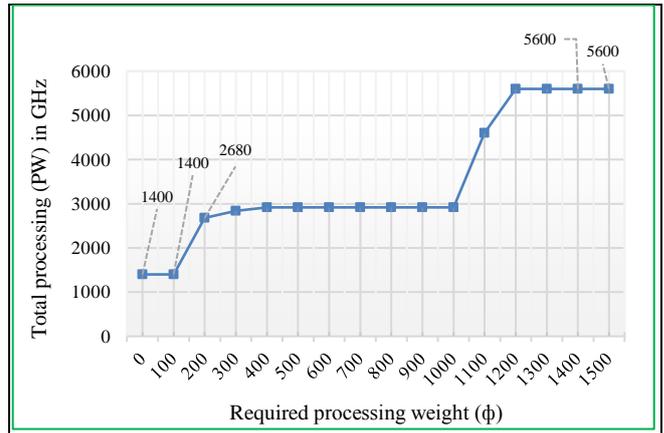
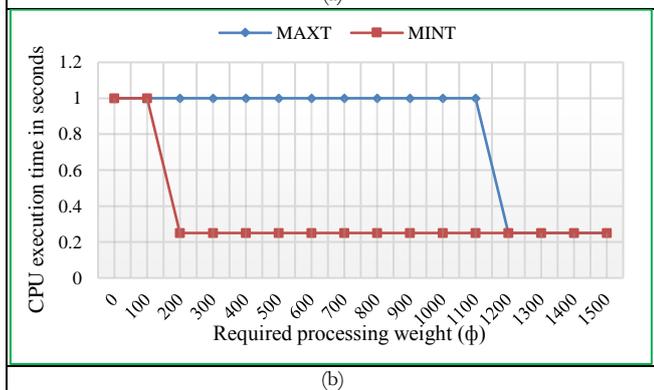
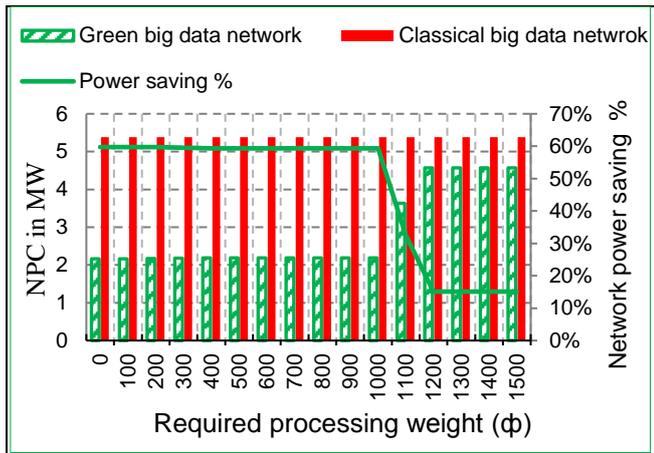


Figure 3 (a). Network power consumption for classical and green big data networks vs ϕ when $CH_s=100$, for velocity scenario 2.2.1. (b) Max and Min CPU execution time needed to process the Chunks in the network vs Φ when $CH_s=100$, for velocity scenario 2.2.1. (c) Relation between Φ and the total amount of computational resources needed when $CH_s=100$, for velocity scenario 2.2.1. Normalized utilization of processing in PNs and DCs vs ϕ when $CH_s=100$, for velocity scenario 2.2.1.



To conclude, 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.

2.2.2 Different Volume, PRR and RCET per Chunk

In this scenario, we assume that each node generates 100 Chunks per second with random uniform distribution that ranges between 10 Gb and 220 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 0.02 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 displays the input values for this scenario.

Table 5 Velocity scenario 2.2.2 parameters.

Number of Chunks per node per second (CH_s)	100
Volume per Chunk in Gb ($Chunk_{sc}$)	10-220 (random uniform)
PRR _{sc} per Chunk	0.02-1 (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

Figure 4-a shows a gradual increase in the network power consumption for the green big data networks 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 2.2.1 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 220 Gb compared to the 50 Gb Chunk size in the velocity scenario 2.2.1. 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 2.2.1, 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 2.2.1. 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 4-b explains the effect of Φ on the T_p for each PN and DC. It can be seen that 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 $0 \leq \Phi \leq 100$. 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 will be 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 $\Phi = 1100$, while it affects later PN #11 (20 servers) at $\Phi = 1900$ and PN #7 (with only 10 servers) at $\Phi = 2200$. 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 $ACET$ s. The T_p of all the PNs is at the lowest value when $\Phi > 2600$, which means that all the PNs' processing resources are fully

utilized 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.

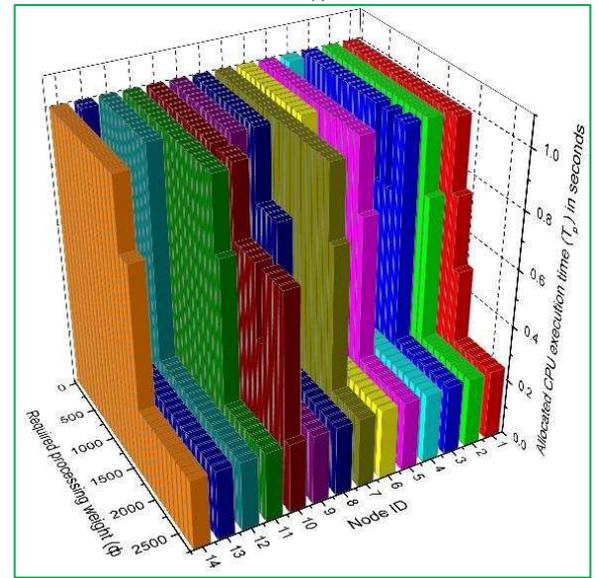
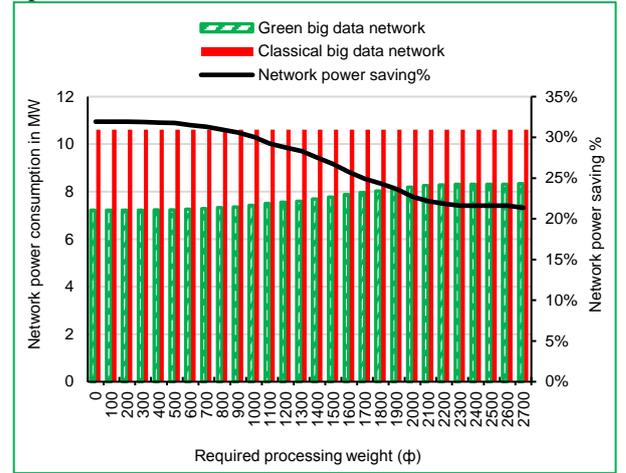


Figure 4. (a) Network power consumption for classical and green big data networks vs ϕ when $CH_s=100$, for velocity scenario 2.2.2. (b) CPU execution time (T_p) allocated to process Chunks at each PN and each DC vs ϕ when $CH_s=100$, for scenario 2.2.2.

3. Conclusions

This paper introduced a Mixed Integer Linear Programming (MILP) model to investigate the impact of the velocity of big data on greening big data networks in bypass IP over WDM core networks. We used our green big data network 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 Chunks have to be processed quickly by utilizing 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 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.

4. Acknowledgments

The authors would like to acknowledge funding from the Engineering and Physical Sciences Research Council (EPSRC), INTERNET (EP/H040536/1) and STAR (EP/K016873/1). The first author would like to thank the Higher Committee for Education Development in Iraq (HCED Iraq) for funding his PhD. All data is provided in full in the results section of this paper. All data are provided in full in the results section of this paper.

5. References

- [1] Zikopoulos, P. and Eaton, C., *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, (McGraw-Hill Osborne Media, 2011)
- [2] NASA, "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].
- [3] IBM"Top tips for securing big data environments | IBM Big Data & Analytics Hub", I.c., 2016. [Online]. Available: <http://www.ibmbigdatahub.com/blog/top-tips-securing-big-data-environments>. [Accessed: 09- Nov- 2016].
- [4] Zeng, D., Gu, L., and Guo, S., *Cost Minimization for Big Data Processing in Geo-Distributed Data Centers*, *Cloud Networking for Big Data*, (Springer, 2015)
- [5] Jayalath, C., Stephen, J., and Eugster, P., 'From the Cloud to the Atmosphere: Running Mapreduce across Data Centers', *Computers, IEEE Transactions on*, 2014, 63, (1), pp. 74-87.
- [6] Satoh, I., '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)*
- [7] Xia, Q., Liang, W., and Xu, Z., 'Data Locality-Aware Query Evaluation for Big Data Analytics in Distributed Clouds', in, *Advanced Cloud and Big Data (CBD), 2014 Second International Conference on, (IEEE, 2014)*
- [8] Rupprecht, L., 'Exploiting in-Network Processing for Big Data Management', in, *Proceedings of the 2013 SIGMOD/PODS Ph. D. symposium, (ACM, 2013)*
- [9] Lawey, A.Q., El-Gorashi, T.E., and Elmirghani, J.M., 'Distributed Energy Efficient Clouds over Core Networks', *Journal of Lightwave Technology*, 2014, 32, (7), pp. 1261-1281.
- [10] Al-Salim, A.M., El-Gorashi, T., Lawey, A.Q., and Elmirghani, J.M., 'Energy Efficient Big Data Networks: Impact of Volume and Variety', *Journal of Transactions on Networking and Service Management (TNSM)*, Submitted on Sept, 2017.
- [11] Al-Salim, A.M., El-Gorashi, T., Lawey, A.Q., and Elmirghani, J.M., 'Greening Big Data Networks: Veractiy Impact', *Springer Journal of Photoic Communications*, Submitted on Aug, 2017.
- [12] Chen, Y., Ganapathi, A., Griffith, R., and Katz, R., '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, (IEEE, 2011)*
- [13] Al-Salim, A.M., Lawey, A.Q., El-Gorashi, T., and Elmirghani, J.M., 'Energy Efficient Tapered Data Networks for Big Data Processing in Ip/Wdm Networks', in, *Transparent Optical Networks (ICTON), 2015 17th International Conference on, (IEEE, 2015)*
- [14] Wirtz, T. and Ge, R., 'Improving Mapreduce Energy Efficiency for Computation Intensive Workloads', in, *Green Computing Conference and Workshops (IGCC), 2011 International, (IEEE, 2011)*
- [15] Patterson, D.A. and Hennessy, J.L., *Computer Organization and Design: The Hardware/Software Interface*, (Newnes, 2013)
- [16] Shen, G. and Tucker, R.S., 'Energy-Minimized Design for Ip over Wdm Networks', *Journal of Optical Communications and Networking*, 2009, 1, (1), pp. 176-186.
- [17] Beloglazov, A., Abawajy, J., and Buyya, R., 'Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing', *Future Generation Computer Systems*, 2012, 28, (5), pp. 755-768.
- [18] Rizvandi, N.B., Taheri, J., Moraveji, R., and Zomaya, A.Y., *On Modelling and Prediction of Total Cpu Usage for Applications in Mapreduce Environments*, *Algorithms and Architectures for Parallel Processing*, (Springer, 2012)
- [19] 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*, 2015, Version 1.
- [20] ARM, "Processors – ARM ARM | The Architecture for the Digital World", A.L., 2016. [Online]. Available: <http://www.arm.com/products/processors>. [Accessed: 09- Nov- 2016]