Article:

Al-Salim, AM, Lawey, AQ, El-Gorashi, TEH et al. (1 more author) (2018) Energy Efficient Big Data Networks: Impact of Volume and Variety. IEEE Transactions on Network and Service Management, 15 (1). pp. 458-474. ISSN 1932-4537

https://doi.org/10.1109/TNSM.2017.2787624

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

Energy Efficient Big Data Networks: Impact of Volume and Variety

Ali M. Al-Salim, Ahmed Q. Lawey, Taisir E. H. El-Gorashi and Jaafar M. H. Elmirghani

Abstract—In this article, we study the impact of big data's volume and variety dimensions on Energy Efficient Big Data Networks (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 datacenters. Secondly, we validate the MILP operation by developing a heuristic that mimics, in real time, the behaviour of the MILP for the volume dimension. Thirdly, we test the energy efficiency limits of our green approach under several conditions where PNs are less energy efficient in terms of processing and communication compared to data centers. Fourthly, we test the performance limits in our energy efficient 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 centralized data centers. Our results revealed that up to 52% and 47% power saving can be achieved by the EEBDN approach compared to the CBDN approach, under the impact of volume and variety scenarios, respectively. Moreover, our results identify the limits of the progressive processing approach and in particular the conditions under which the CBDN centralized approach is more appropriate given certain PNs energy efficiency and software availability levels.

Index Terms —**Big data volume, big data variety, energy efficient networks, IP over WDM core networks, MILP, processing location optimization, software matching**.

I. INTRODUCTION

The significant growth in the Internet-connected devices is leading 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.

The first challenge facing the Data Centers (DCs) is the enormous volume of data fluxing to them. Facebook and Twitter create more than 18 Terabytes (TB) every single day [1]. More than 210 billion emails are sent every day [2]. The size of the climate change data repositories is projected to grow to nearly 350 Petabytes (PBs) by 2030 [3]. Five PBs is equivalent to the total number of letters delivered by the US Postal Service in one year [4]. Furthermore, the number of Internet-connected devices is expected to grow to 100 billion devices by 2020 [5]. The challenge is that much generated data, currently, is not analyzed or addressed to extract insights at all [1]. Growth in the speed of generating data, as well as growth in the volume of data and in the variety of data sources is causing reduction in the percentage of processed data of organizations due to the lack of resources and poor analysis tools [1]. 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. Fig.1 shows a decrease in the ratio of processed data to the overall huge volume of big data created continuously [6].

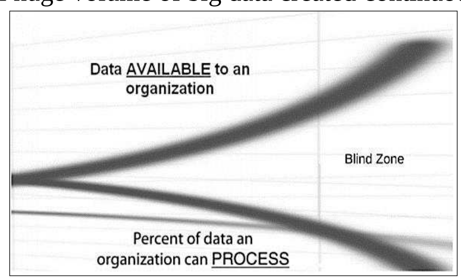


Fig 1: Volume of data is increasing, while percentage of data that can be processed is declining [6].

Managing massive data volumes calls for new processing and communications approaches. In addition, gathering and transmitting big data is exposing new challenges in terms of 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 has to 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 materializes. Such application level constraints impose even more challenges and hard trade-offs on the energy efficiency that can be attained from optimizing 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 itself, 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. The waveform hopefully indicates for years that the person is fine and therefore its transmission to emergency services or to data centers 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 (variety), we consider different data reduction factors that relate the volume of the original data to the volume of the knowledge extracted, all measured in bits; and consider different processing requirements for the same volume under different applications, another variety dimension.

Therefore, new processing and communication techniques are required to process and transmit such colossal amount of data. Data centers are located in core networks where IP over WDM is used [7]. In [8], however, the authors described the drivers, building blocks, architecture, and enabling technologies for elastic optical networks. The authors in [9] suggested that efficient bulk-data transfer in elastic optical networks (EONs) can be achieved with malleable reservation (MR). The authors in [10] discussed the technologies needed for realizing highly efficient data migration and backup for big data applications in elastic optical inter-data-center (inter-DC) networks. In [11], 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 [12], the authors studied the minimization of the overall cost for Big Data placement, processing, and movement across geo-distributed data centers. In [13], the authors proposed a streaming workflow allocation algorithm that takes into consideration the characteristics of streaming work and the price diversity among geo-distributed DCs. The authors in [14] aimed to keep the communication cost to a minimum by satisfying as many big data queries as possible over a number of time slots. The authors in [15] developed a framework to perform a sequence of MapReduce jobs in Geo-distributed DCs where the processing of jobs is optimized according to time and monetary cost. In [16], the authors developed an energy efficient cloud computing framework in IP over WDM core networks.

In [17], we presented preliminary results to demonstrate that the Network Power Consumption (NPC) is affected by 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 [18]. We examined improving the energy efficiency of big data networks by using a progressive processing technique in Processing Nodes (PNs) along the data journey from the source to the destination. We referred to such a network as an Energy Efficient Tapered Data Network since there is a significant reduction in the data transported over the network each time the data is processed. Our work in [19] and [20] considered big data velocity and veracity respectively, which are two dimensions of big data that are not related to the current work. In the velocity dimension, we presented several scenarios to process expedited-data and relaxed-data made up of big data Chunks progressively starting at the source nodes of the network, moving through the intermediate nodes of the network and finally processing at the centralized data centers. In the veracity dimension, we inspected the impact of cleansing and backup operations of big data on the energy efficient big data networks. In [21] we presented preliminary results that considered big data volume only. The current paper makes a number of new contributions beyond [21]: (i) it considers big data variety for the first time, (ii) it introduces and considers the new software matching problem in big data networks, (iii) compared to [21], it provides the MILP formulation which is not in [21] and a very wide range of results, (iv) it evaluates for the first time the impact of the power efficiency of PNs, (v) it presents our new Energy Efficient Big Data Networks (EEBDN) heuristic and its complexity, (vi) it presents new results under different network topologies.

II. EEBDN: Description with Illustrative Example

Four main Vs can characterize big data: volume, velocity, variety, and veracity. In this work, we introduce MILP models and a heuristic to observe the impact of volume and variety of big data on the NPC in bypass IP over WDM core networks.

A. Problem Statement

In this work, we focus on the big data large volume and variety and address the problem of power minimization 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 address four main problems. Firstly, where to process big data to minimize power consumption given limited processing capacity at source and intermediate nodes but large processing capability at central data centers. Secondly, we address the problem of how to optimally, from an energy efficiency point of view, deal with big data chunks that have variable processing requirements. Thirdly, we consider the problem of jointly optimizing 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 evaluate 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 focus 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.

B. Contributions

Compared to previous work, we make the following fundamental contributions: (i) we evaluate, using MILP and heuristic, the volume dimension by analysing the 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 examine, 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 use our EEBDN processing technique to optimize the processing locations of the big data Chunks and compare the results to the Classical Big Data Networks (CBDN) technique where no PNs exist in the network. The processing locations are optimally chosen at either Source PNs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

(SPNs), inside "location optimized" DCs or at Intermediate PNs (IPNs). Thus, we jointly minimize the power consumption of the overall network and processing resources; (iv) we assess the impact of energy efficiency of PNs on our EEBDN approach where the PNs constituents (LAN switches, routers and servers) consume higher power compared to data centers equipment; (v) we consider 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. Table I defines the acronyms used in this paper.

TABLE I
LIST OF ACRONYMS DEFINITIONS

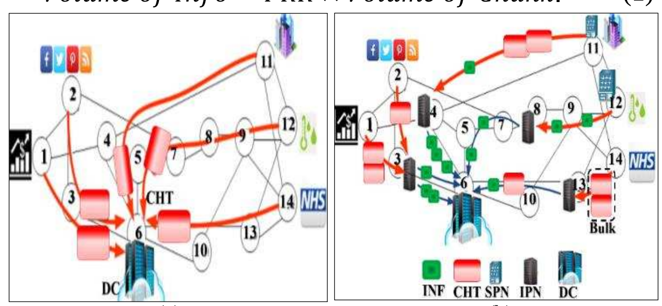
Acronyms	Definition
CBND	Classical big data networks
CHT	Chunk traffic (Unprocessed big data traffic)
DC	Data center
EEBDN	Energy efficient big data networks
INF	Info traffic (processed big data traffic)
IPN	Intermediate processing node
NPC	Network power consumption
PN	Processing node
PRR	Processing reduction ratio
SPN	Source processing node

C. CBDN vs. EEBDN

The concept of EEBDN is illustrated in the NSFFENT [22] of Fig. 2. Fig. 2-a shows a CBDN. The difference between the two approaches is that in CBDN, all big data Chunks traffic (CHT) generated by the source nodes is forwarded to the DCs directly to be processed there. For example, the source node # 14 of the National Health Service (NHS) in Fig. 2-a acts as a source. In the EEBDN, shown in Fig. 2-b, however, PNs are attached to the IP over WDM core nodes (e.g., node #12) to process as many Chunks as possible according to PNs processing capability to reduce the number of forwarded Chunks to the DCs, hence reduce the volume of big data traffic.

The extracted information traffic (INF) from the corresponding Chunks is forwarded to the DCs through energy efficient routes. We refer to the extracted information as Info. The structure of a PN is similar to the cloud structure presented in [16].PNs are located close to the user while data centers 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 centers 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 taken into account in our progressive processing approach. The number of processed Chunks inside the PNs is variable and depends on the PN's processing capacity which may be limited by the available space in the building housing the core network node. Note in Fig. 2-b Chunks can be processed either in SPNs or IPNs. Once the PNs' servers are fully utilized, no more edge or intermediate processing is performed, centralized 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. In many big data applications, the ratio of the input volume of data before processing to the output volume of data after processing is very high, i.e. the volume of Infos is much smaller than the volume of Chunks [18]. For example, the input Chunks can be video surveillance segments representing few minutes or hours. 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 introduce equation (1) where the volume of the Chunks is multiplied by different PRRs to produce the Infos, the knowledge in the Chunks. For instance, in MapReduce jobs, a Chunk of 1000 Gigabit (Gb) and PRR of 0.001 results in Info of 1 Gb [18]. Accordingly, significant network power saving is achieved if such Chunks are processed locally in the edge (SPNs) and progressively in the IPNs.

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



(a) (b)

Fig. 2. (a) CBDN, (b) EEBDN

D. *EEBDN: Illustrative Example*

To illustrate the concepts we propose in this work, consider the example network shown in Fig. 3. There are four zones in Fig. 3, 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 the SPN as it is the first PN in which Chunks are received from its corresponding zone and the Chunks are locally or centrally processed. Each SPN can locally process a different maximum number of Chunks depending on its processing, storage and internal switches and routers capacity. The remaining Chunks that cannot be processed locally in an SPN are forwarded either to another optimally selected PN or a DC. Those PNs that receive Chunks from other SPNs are called 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 Chunks either in SPNs, IPNs or in the DCs, knowledge is extracted in the form of smaller rate traffic that we call the Info Big Data Traffic (INF). INF propagates from PNs (SPN or IPNs) towards 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 these DCs. As mentioned before, a PN is built at a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

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.

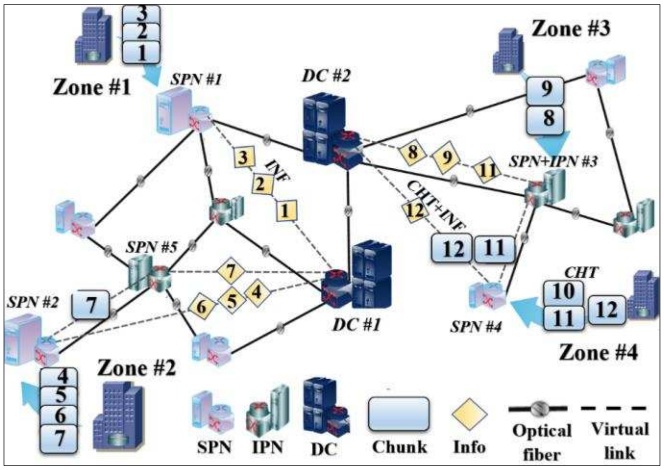


Fig. 3. EEBDN illustrative example

Each zone in Fig. 3 represents a probable scenario that our approach can optimize 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 the resources (CPU, storage, internal switches, and routers) of SPN #2 have been fully utilized. After Chunk #7 is forwarded to IPN #5, it will be processed there and the output (Info #7) will be 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 utilized. 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 Chunks (Chunk #10) and forwards any extra Chunks 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 Chunks by SPN #4 (i.e., Chunk #12) will be uploaded directly from SPN #4 to be processed by an optimally selected DC (DC #2 in Fig. 3). For such an event, CHT starts to dominate the network traffic from SPNs to DCs.

III. Impact of Volume on EEBDN

This section introduces a MILP model that focuses on the first V of big data, i.e., the volume, and evaluates several scenarios to study the impact of volume on EEBDN.

A. Volume MILP Model

In this section, we introduce a MILP model for EEBDN using a bypass IP over WDM network, (see [16] and [23] for details of MILP in IP over WDM networks). The PNs are attached to each core node of the NSFNET and consist of limited processing and storage resources, as depicted in Fig. 2, while the DCs comprise large enough resources. The NSFNET network consists of 14 nodes connected by 21 bidirectional links [22].

The NPC is comprised of the power consumption of the router ports, transponders, EDFAs, regenerators and optical switches. On the other hand, 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 assume, 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 [23].

Table II defines the parameters and variables used in the EEBDN model:

TABLE II
List of Parameters and Variables and Their Definitions.

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 fiber 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 fiber.
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, $A_{mn} = \left\lfloor \frac{D_{mn}}{S} - 1 \right\rfloor + 2$ [22].
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 optimized DCs.
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 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 $DC_d = 0$.

Under the bypass approach, the total IP over WDM NPC is composed of the following components

1) The power consumption of router ports

$$\sum_{i\in N} PR\cdot\ (AR_i + ACH_i + AI_i) + PR\cdot \sum_{j\in N:\, i\neq j} (C_{ij}). \quad (2)$$

2) The power consumption of transponders

$$\sum_{m\in N}\sum_{n\in N_m} PTR\cdot W_{mn}. \quad (3)$$

3) The power consumption of regenerators is

$$\sum_{m\in N}\sum_{n\in N_m} PRG\cdot\ W_{mn}\cdot RG_{mn}. \quad (4)$$

4) The power consumption of EDFAs

$$\sum_{m\in N}\sum_{n\in N_m} PE\cdot A_{mn}\cdot F_{mn}. \quad (5)$$

5) The power consumption of optical switches

$$\sum_{i\in N} PO_i. \quad (6)$$

Equation (2) 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 (3) and (4) evaluate the power consumption of all the transponders and regenerators in the optical layer. Equation (5) evaluates the total power consumption of the EDFAs in the optical layer. Equation (6) 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

$$PSR = \sum_{p\in N}\sum_{s\in N} CHT_{sp}\cdot\ (RS\cdot SEB + RR\cdot REB) + \sum_{p\in N}\sum_{d\in N}(CHT_{pd} + INF_{pd}) \cdot (RS\cdot SEB + RR\cdot REB) + \sum_{p\in N}\sum_{d\in N} INF_{pd}\cdot (RS\cdot SEB + RR\cdot REB). \quad (7)$$

Equation (7) 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 all servers inside PNs and DCs

$$\sum_{p\in N} \delta\cdot PNW_p +\ NS_p\cdot PIDLE. \quad (8)$$

3) The power consumption of the storage

$$\sum_{p\in N} SCH_p\cdot RSG\cdot PSG. \quad (9)$$

Note that server power consumption is a function of the idle power, maximum power and CPU utilization [24]. Therefore, the power consumption of all servers inside the PNs and DCs is calculated using equation (8). Equation (9) 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

$$PUN\cdot\left(\sum_{i\in N} PR\cdot\ (AR_i + ACH_i + AI_i) + PR\cdot \sum_{j\in N:i\neq j}(C_{ij}) + \sum_{m\in N}\sum_{n\in N_m} PTR\cdot W_{mn} + \sum_{m\in N}\sum_{n\in N_m} PRG\cdot W_{mn} \cdot RG_{mn} + \sum_{m\in N}\sum_{n\in N_m} PE\cdot A_{mn}\cdot F_{mn} + \sum_{i\in N} EO_i\right)$$
$$+\ PU\cdot\left(\sum_{p\in N}\delta\cdot PNW_p. +\ NS_p\cdot PIDLE + \sum_{p\in N}\sum_{s\in N} CHT_{sp}\cdot\ (RS\cdot SEB + RR\cdot REB) + \sum_{p\in N}\sum_{d\in N}(CHT_{pd} + INT_{pd}) \cdot (RS\cdot SEB + RR\cdot REB) + \sum_{p\in N}\sum_{d\in N} INF_{pd}\cdot (RS\cdot SEB + RR\cdot REB) + \sum_{p\in N} SCH_p\cdot RSG\cdot PSG\right). \quad (10)$$

Equation (10) gives the model objective, which is to minimize the IP over WDM NPC as well as the PNs' and DCs' power consumption.

Subject to: PNs and DCs Constraints:

1) Processing counter of big data Chunks constraint

$$\sum_{p\in N} Y_{spc} = 1, \forall s\in N, \forall c\in CH_s. \quad (11)$$

Constraint (11) 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

$$CHT_{sp} = \sum_{c\in CH_s} CHV_{sc}\cdot Y_{spc}, \forall s,p\ \in N. \quad (12)$$

Constraint (12) 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

$$\sum_{d\in N} INF_{pd} = \sum_{s\in N}\sum_{c\in CH_s} CHV_{sc}\cdot Y_{spc}\cdot PRR_{sc} \quad \forall p\in N. \quad (13)$$

Constraint (13) 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

$$\sum_{p\in N} INF_{pd} \geq DC_d, \quad \forall d \in N, \tag{14}$$

$$\sum_{p\in N} INF_{pd} \leq Z \cdot DC_d, \ \ \forall d \in N, \tag{15}$$

$$DCN = \sum_{d\in N} DC_d. \tag{16}$$

Constraints (14) and (15) 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 (16) limits the total number of built DCs to DCN.

5) PNs and DCs workload and processing capacity constraints

$$PNW_p = \sum_{s\in N}\sum_{c\in CH_s} SW_{sc} \cdot Y_{spc}, \qquad \forall p \in N, \tag{17}$$

$$PNW_p \leq NS_p \cdot MSW + (M \cdot DC_p), \qquad \forall p \in N. \tag{18}$$

Constraint (17) represents the total workload at PN p, which is the summation of the CPU workload of all the servers in that PN. Constraint (18) 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 minimize the NPC and number of active servers.

6) PNs and DCs storage constraints

$$SCH_p = \sum_{s\in N}\sum_{c\in CH_s} CHV_{sc} \cdot Y_{spc}, \qquad \forall p \in N, \tag{19}$$

$$SCH_p \leq MS_p + (H \cdot DC_p), \qquad \forall p \in N. \tag{20}$$

Constraint (19) represents the size of Chunks in Gb stored in PN p. Constraint (20) 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

$$\sum_{s\in N} CHT_{sp} \leq MSR_p + (A \cdot DC_p), \qquad \forall p \in N. \tag{21}$$

Constraint (21) 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 assume 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

$$\sum_{j\in N:\, i\neq j} R_{ij}^{sd} - \sum_{j\in N:\, i\neq j} R_{ji}^{sd} = \begin{cases} R_{sd} & i = s \\ -R_{sd} & i = d \\ 0 & otherwise \end{cases} \quad \forall s,d,i \in N: s\neq d \tag{22}$$

2) Flow conservation constraints for the big data Chunks traffic

$$\sum_{j\in N:\, i\neq j} CHT_{ij}^{sp} - \sum_{j\in N:\, i\neq j} CHT_{ji}^{sp} = \begin{cases} CHT_{sp} & i = s \\ -CHT_{sp} & i = p \\ 0 & otherwise \end{cases} \quad \forall s,p,i \in N: s\neq p. \tag{23}$$

3) Flow conservation constraints for the big data Info traffic

$$\sum_{j\in N:\, i\neq j} INF_{ij}^{pd} - \sum_{j\in N:\, i\neq j} INF_{ji}^{pd} = \begin{cases} INF_{pd} & m = p \\ -INF_{pd} & m = d \\ 0 & otherwise \end{cases} \quad \forall p,i \in N, \forall d \in N: p\neq d. \tag{24}$$

Constraints (22-24) 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

$$\left(\sum_{s\in N}\sum_{d\in N:\, s\neq d} R_{ij}^{sd} + \sum_{s\in N}\sum_{p\in N:\, s\neq p} CHT_{ij}^{sp} + \sum_{p\in N}\sum_{d\in N:\, p\neq d} INF_{ij}^{pd}\right) \leq C_{ij}.B \quad \forall i,j \in N: i\neq j. \tag{25}$$

Constraint (25) ensures that the summation of all traffic flows through a virtual link does not exceed its capacity.

5) Optical layer flow conservation constraints:

$$\sum_{n\in N_m} W_{mn}^{ij} - \sum_{n\in N_m} W_{mn}^{ij} = \begin{cases} C_{ij} & m = i \\ -C_{ij} & m = j \\ 0 & otherwise \end{cases} \quad \forall i,j,m \in N: i\neq j. \tag{26}$$

Constraint (26) 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 utilization of bandwidth and reduce blocking probabilities.

6) Physical link capacity constraints

$$\sum_{i\in N}\sum_{j\in N:\, i\neq j} W_{mn}^{ij} \leq WL \cdot F_{mn}. \qquad \forall m \in N, n \in N_m. \tag{27}$$

Constraint (27) ensures that the summation of the wavelengths in a virtual link traversing a physical link does not exceed the capacity of the fibers in the physical link.

7) Wavelengths capacity constraint

$$\sum_{i\in N}\sum_{j\in N:\, i\neq j} W_{mn}^{ij} = W_{mn} \qquad \forall m \in N, n \in N_m. \tag{28}$$

Constraint (28) ensures that the summation of the wavelengths traversing a physical link does not exceed the total number of wavelengths in that link.

8) Number of aggregation ports utilized by regular traffic constraint

$$AR_i = \frac{1}{B} \cdot \sum_{d\in N:\, i\neq d} R_{id} \qquad \forall i \in N. \tag{29}$$

9) Number of aggregation ports utilized by CHT traffic constraint

$$ACH_i = \frac{1}{B} \cdot \sum_{p\in N:\, i\neq p} CHT_{ip} \qquad \forall i \in N. \tag{30}$$

10) Number of aggregation ports utilized by INF traffic constraint

$$AI_i = \frac{1}{B} \cdot \sum_{d\in N:\, i\neq p} INF_{id} \qquad \forall i \in N. \tag{31}$$

Constraints (29-31) 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}.

B. EEBDN Heuristic

In this section, we validate 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

insight into what minimizes 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 chunks 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 minimize 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 in Fig. 4 shows the heuristic, which aims to process the incoming Chunks by utilizing the minimum number of resources so that minimal power is consumed. The heuristic is initialized 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 characterized 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 [22]. 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.

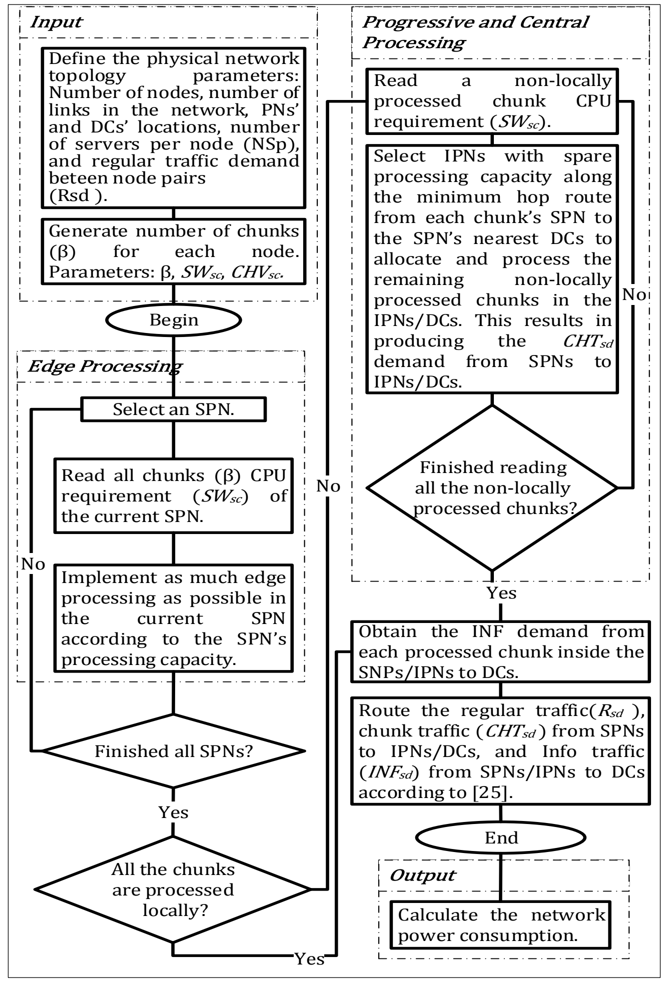


Fig. 4. EEBDN: volume heuristic.

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 [22] and the total NPC is calculated.

C. Complexity Analysis

The proposed EEBDN heuristic aims to work around the NP-hard complexity [25] 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 Chunks per node (β)) of different volumes must be packed into a finite number of bins (servers) each of capacity C in a way that minimizes 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 [26]. Secondly, the generation of initial set of paths is based on minimum hop

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

routing algorithm, which has a complexity of the order $O(N)$ [27], 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.

IV. RESULTS OF VOLUME SCENARIOS

Our MILP model and the heuristic were evaluated using the NSFNET network depicted in Fig. 2. 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 [28]. Table III summarizes the input parameters. We performed the MILP optimization 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.

TABLE III
INPUT DATA FOR VOLUME MODEL.

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, 29]
PNs and DCs switch capacity (CS)	320 Gbps [16, 29]
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, 29]
PNs and DCs router capacity (CR)	660 Gbps [16, 29]
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 [30]
IP over WDM regenerator power consumption (PRG)	334 W [30]
IP over WDM transponder power consumption (PTR)	167 W [30]
IP over WDM optical switch power consumption (PO_i) $\forall i \in N$	85 W [30]
IP over WDM EDFA power consumption (PE)	55 W [30]
Wavelength bit rate (B)	40 Gbps
Distance between EDFAs (S)	80 km
Number of wavelengths per fiber (WL)	32
Number of location optimized DCs (DCN)	2
IP over WDM power usage effectiveness (PUN)	1.5 [16]
PNs and DCs power usage effectiveness (PU)	2.5 [16]

The MILP in Section III.A 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.

Note that the amount of computational resources required to process the data is the same in our approach and the classical approach where all Chunks are processed inside DCs. To provide a holistic assessment of the impact of the volume dimension on the EEBDN, we evaluate the proposed progressive processing approach in two volume scenarios.

A. Deterministic Chunks Volume, PRR = 0.001, Number of Servers per PN = 5-15 Server

In this scenario, we consider 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 consider 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 β = 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 will take one second for the transmission of the given Chunks and the corresponding Infos. This is a reasonable assumption as we consider 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 IV summarizes the input values needed in this scenario.

TABLE IV
VOLUME SCENARIO A PARAMETERS.

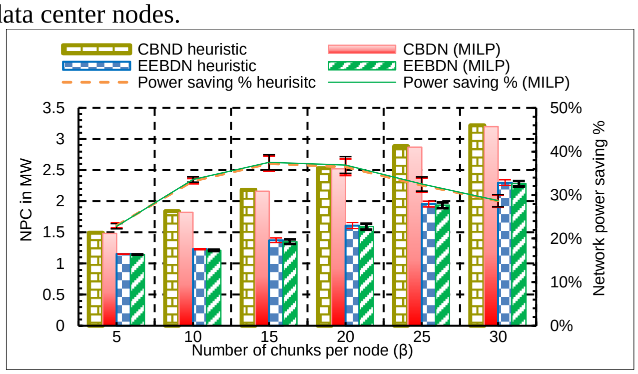
Number of Chunks per node (β)	Number of servers per PN (NS_p)	CPU workload per Chunk in GHz (SW_{sc})	Chunk volume in Gb (CHV_{sc})	PRR_{sc}
5-30	5-15	3 [31]	80 [18]	0.001 [18]

The results in Fig. 5-a are based on our MILP optimization 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.

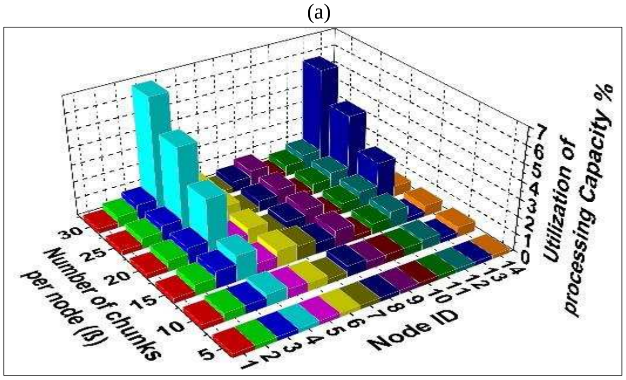
For the MILP, and for all cases, the NPC increased when β increased as more Chunks are delivered to the network. Introducing the PNs has, however, greatly bounded the growth in power consumption when the number of Chunks increased which leads to network power savings compared to the classical approach in all the cases of the considered values of β due to processing near the source. At β = 5, the network power saving is smaller than that at β = 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 β = 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 β = 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 β = 15 which has more Infos. Note that a maximum network power saving of 38% is achieved at β = 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 IV are used. We used the heuristic to evaluate the impact of volume in the current scenario only due to paper length limitations. The performance of the EEBDN heuristic was compared to the MILP performance in Fig. 5-a and the heuristic and MILP are in close agreement. From Fig. 5-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 show the average values. The 95% confidence interval [32] is shown as error-bars.

Fig. 5-b shows the utilization of the processing capacity as a % of the PN processing capacity. At β = 5 all the SPNs are capable of performing 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. This results in a CHT between SPNs and IPNs, and very small amount of CHT between SPNs and DCs, thereby minimizing the DCs processing utilization. Note that PN #12, which has the capacity to process up to 20 Chunks, is 100% utilized at β = 15. This is because this node processed its own 15 Chunks and handled an extra five progressed Chunks from other SPNs. At β > 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. This is why the DCs processing utilization increases dramatically. Note that nodes 4 and 13 have high utilization as they are the two data center nodes.



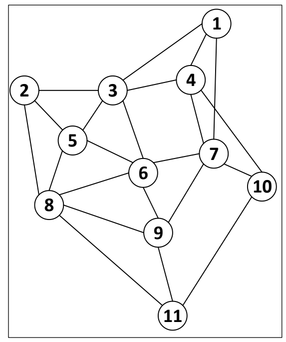
(a)



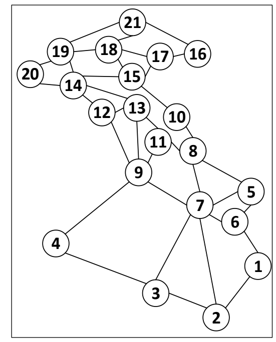
(b)

Fig. 5. (a) CBDN power consumption vs EEBDN power consumption (MILP and heuristic) for volume scenario A. (b) Utilization of processing capacity % in the EEBDN with different values of β for volume scenario A.

The main goal in this article is to show the effectiveness of our progressive processing approach compared to the classical centralized processing approach. We carried out a comparison with the classical (centralized) case, which is the case that is known in the literature and can act as a benchmark. Furthermore, we have evaluated the complexity of our heuristic in Section III.C and therefore provide details relating to complexity / efficiency of our heuristic. The effectiveness of our heuristic was evaluated and it is shown to produce results close to the optimum MILP results, for example Fig. 5-a.

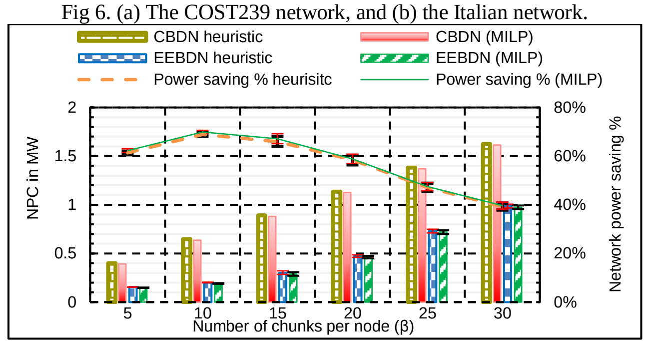


(a)

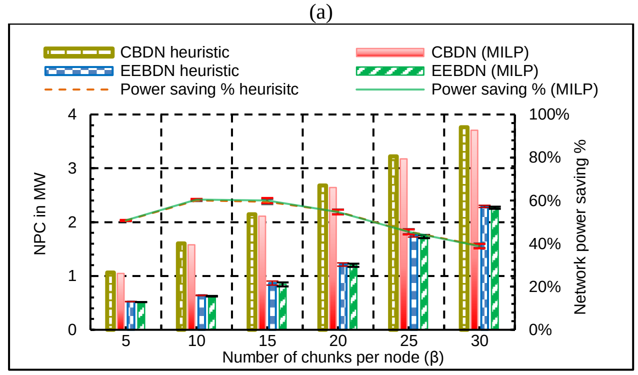


(b)

Fig 6. (a) The COST239 network, and (b) the Italian network.



(a)



(b)

Fig. 7. CBDN power consumption vs EEBDN power consumption (a) COST239 network, (b) Italian network.

We re-evaluated the volume scenario of MILP model and heuristic on two more different networks. The COST239 network [33], (see Fig 6-a), which is smaller than the NSFNET and consists of 11 nodes and 25 bidirectional links, and the Italian network [34, 35], (see Fig 6-b) which is bigger than the NSFNET and consists of 21 nodes and 36 bidirectional links.

The results in Fig. 7-a and Fig. 7-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. On the other hand, the power savings of the MILP and heuristic obtained in the Italian network are 52%

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

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 optimization similar to that in [7]. The heuristic and MILP results are in close agreement in the cases described above.

B. Deterministic Chunks Volume, PRR = 0.001, Number of Servers per PN = 10-30 Servers

In this scenario, we have 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 V.

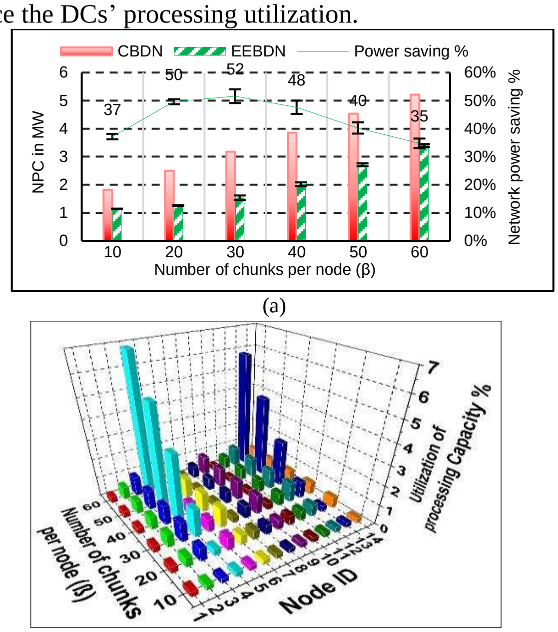
TABLE V
VOLUME SCENARIO B PARAMETERS.

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 [31]	80 [18]	0.001[18]

Fig. 8-a displays the NPC of the classical networks and EEBDN. The power saving increased at $10 \le \beta \le 30$ and reached a maximum value of 52% at $\beta = 30$ (compared to the maximum power saving of 38% at $\beta = 15$ in scenario A). This is because the majority of the big data traffic in the network is the INF when $10 \le \beta \le 30$ 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 $10 \ge \beta \ge 60$ (higher than the average power saving of 32% in scenario A) 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 center 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 has to 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 centers, about 90% of the power consumption of IT is attributed to servers and 10% to communication equipment [36]. 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.

Fig. 8-b shows that the SPNs now have the ability to locally process all the Chunks when $\beta = 10$ since their processing capacity has increased. At $20 \le \beta \le 40$, PNs start to reach their maximum processing capacity, such as PNs #2 and #3 at $\beta = 20$ and 30, respectively. Note that only at $\beta = 20$ is the processing utilization 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 A. We also note that the processing utilization of the DCs of the present scenario is smaller than that of scenario A at $\beta = 30$, at which the DC utilization reaches the maximum value for scenario A. This is due to the growth in the PNs' processing capability in the current scenario, which helps to reduce the DCs' processing utilization.



(a)

(b)

Fig. 8. (a) CBDN power consumption vs EEBDN power consumption for volume scenario B. (b) Utilization of processing capacity % in EEBDN with different values of β for volume scenario B.

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 Chunks 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.

C. 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 centralized DCs. This might be due to technology, economic and/or space limitations in PN sites. In this section, we analyze the impact on the power consumption of EEBDN of utilizing 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 (Chunks per PN) in both cases.

Results are shown in Fig. 9 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 centers 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 Fig. 9 (red bars). In addition, and to reduce the complexity of the analysis, the DCs are fixed in the optimal locations obtained in Section IV.A and IV.B as their location is not the critical element that we want to assess.

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 notice that our approach is useful only when the PNs equipment power consumption is between 0% to 20% greater than the DCs equipment power consumption. Beyond this range, the optimal solution is processing the majority of the Chunks in the centralized 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 centers. Total processing in the central data centers becomes more attractive at the point when PNs are 90% less energy efficient than data centers and here the long journey to central data centers 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 center equipment.

An extreme potential scenario may be a situation where the central data center 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 centers in future [37, 38]. This situation is represented by the 90% case in Fig. 9. We conclude that at lower PRR, the EEBDN has the ability to 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 centers, e.g. within 20%.

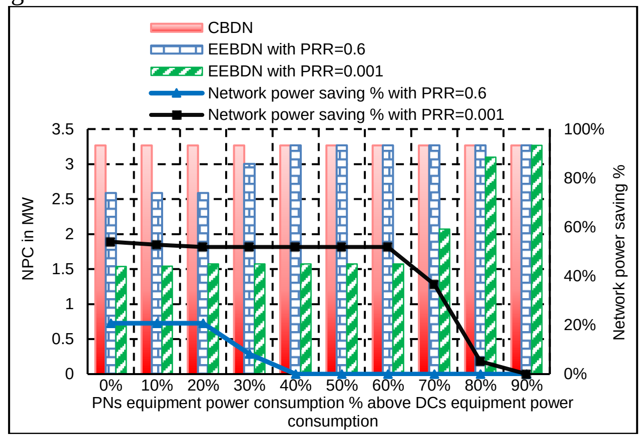


Fig. 9. CBDN power consumption vs EEBDN power consumption when PNs equipment consume more power than DCs equipment at PRR=0.001 and PRR=0.6 with β=30.

D. 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 Chunks, i.e. they are provided with all the necessary software packages that correspond to all the possible types of Chunks. This is obviously not possible in small sized PNs due to processing and storage limitations. Therefore, in this section, we assess the impact of software shortage in PNs on the performance of the overall EEBDN approach in terms of number of processed Chunks at the edge. This analysis is carried by extending our model to include a software matching dimension where Chunks 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 have to be segregated.

D.1 MILP Model Extension Description

In addition to the parameters mentioned in Section III.A, we defined the following parameters:

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 Chunks c generated at node s needs software package g; otherwise, $CSF_{s,c,g} = 0$.

In addition to the constraints mentioned in Section III.A, we define the following constraint:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

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

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.

D.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 analyze 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 leave 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.

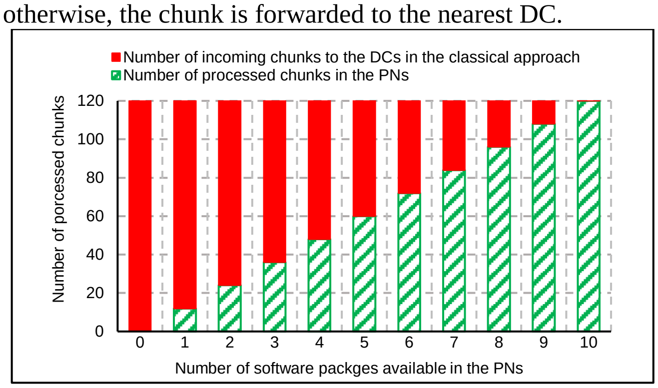


Fig. 10. Software packages availability and its impact on EEBDN performance at β=10.

Fig. 10 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 assume 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 Fig. 10 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 paper. 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. Fig. 10 shows 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.

V. Impact of Variety on EEBDN

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 analyze big data to obtain useful information since they are not related to the relational database technologies. Successful organizations that rely on big data to enrich their decision-making should be able to handle the variety of data [1].

A. Variety MILP Model

The MILP model presented in Section III.A 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.

B. Results of Variety Scenarios

We present in this section the following two scenarios.

B.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 VI demonstrates the input parameters used in this section.

TABLE VI
VARIETY SCENARIO B.1 PARAMETERS.

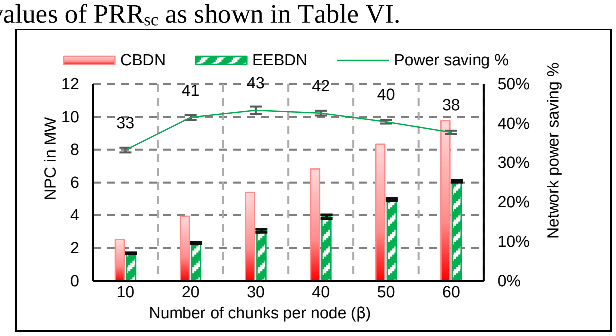
Number of Chunks per PN	Number of Servers per PN (NS_p)	CPU workload per Chunk in GHz (SW_{sc})	Chunk volume in Gb (CHV_{sc})	PRR_{sc}
10-60	10-30	3 [31]	10-330 (random uniform) [18]	0.001-1 (random uniform) [18]

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. Fig. 11-a shows that the maximum power saving is 43% at β = 30 Chunks. An interesting feature

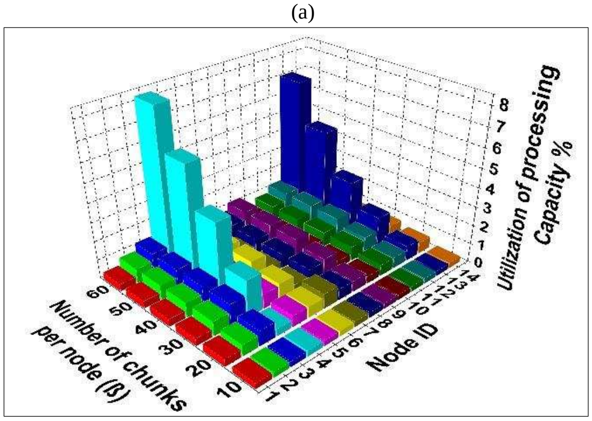
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

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, B, where we obtained a 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 in B, which reduces the power savings achieved.

From the results in Fig. 11-b, it is apparent that the system shows similar performance to volume scenario B in which some PNs exhaust their processing capacity earlier than other nodes, such as PNs #1, #7 and #11 at $\beta = 10$, while other PNs are fully utilized later as is the case for PN #9 at $\beta = 40$. 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 A is due to the assumed insensitivity of CPU utilization to different values of PRR_{sc} as shown in Table VI.



(a)



(b)

Fig. 11 (a) CBDN power consumption vs EEBDN power consumption for variety scenario B.1. (b) Utilization of processing capacity % in EEBDN with different values of β for variety scenario B.1

B.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 take into account 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 VII 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. Fig. 12 shows a sample of the input data for this scenario considering node #8 at $\beta = 10$. Note the variation among different Chunks 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).

TABLE VII
VARIETY SCENARIO B.2 PARAMETERS.

Number of Chunks per PN (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	1-4 (random uniform) [31]	10-330 (random uniform) [18]	0.001-1 (random uniform) [18]

This can represent a WordCount program [39], 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. 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. Fig. 13 shows those two points in the explored space and displays their corresponding applications.

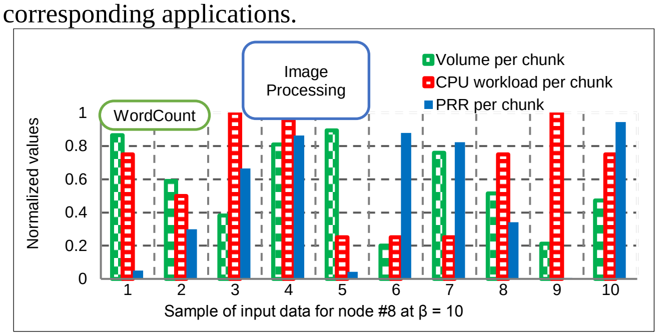
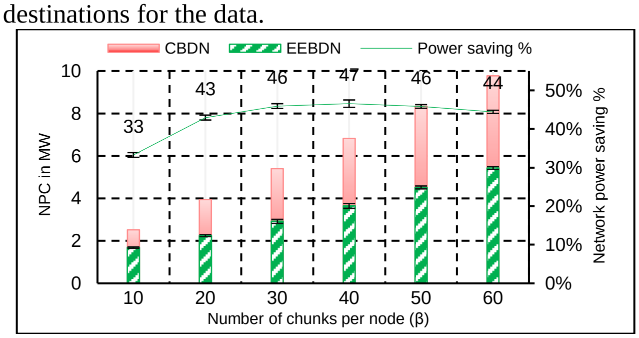


Fig. 12. Sample of input data for variety scenario B.2 for node #8 at β = 10.

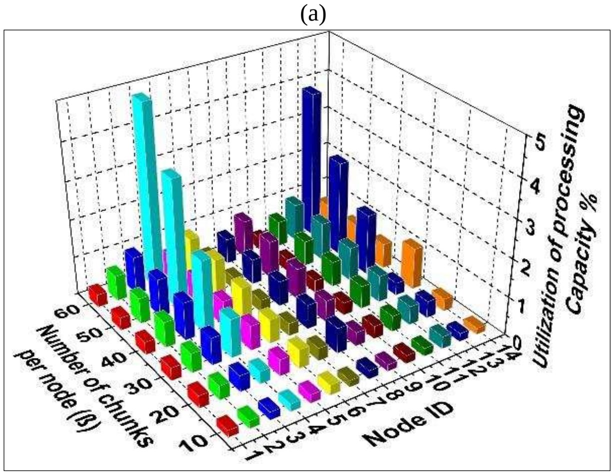
Fig. 13-a displays the main findings and differences with the previous variety scenario B.1, where the CPU workload per Chunk was fixed at 3 GHz. The main observations are as follows: first, the maximum power saving (47%) exceeds the one obtained in the variety scenario B.1 (43%). This is due to the ability to consolidate the CPU processing for more Chunks by PNs as some Chunks arrive with lower processing requirements compared to the variety scenario B.1. Second, the maximum power saving occurred at $\beta = 40$ and not $\beta = 30$ as observed in the variety scenario B.1. This is also due to the extra available processing space at the IPNs due to processing Chunks with low processing requirements.

This paper 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. Fig. 13-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 Fig. 13-a, however, beyond a certain big data traffic volume, the processing capability of PNs at the edge get depleted and the power savings drop (in Fig. 13-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. Fig 12 shows example applications and their PRR values.

Fig. 13-b shows the processing utilization for the different PNs and DCs. First, we note that some PNs are capable now of serving more Chunks compared to variety scenario B.1. This shows the impact of having various CPU workloads per Chunk, which extend the PNs ability to serve more Chunks with lower CPU requirements. Secondly, the model, in most stages, selected optimally the prevailing DC locations at nodes #4 and #13. Note that we re-evaluated our results where we optimized 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.



(a)



(b)

Fig. 13. (a) CBDN power consumption vs EEBDN power consumption for variety scenario B.2. (b) Utilization of processing capacity % in the EEBDN with different values of β for variety scenario B.2.

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 centers 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 [30], [40], [7]. Our work here considers big data traffic as well as regular traffic. For regular traffic see equations 2, 22, 25, 29 and for example the explanation of Fig. 13-a. The interest in big data is attributed to its large volume and the ability to reduce this volume through processing, hence saving power.

VI. Future Work

Future research directions include:

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 generalized 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 generalized 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 center 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.

VII. Conclusions

This paper presented a Mixed Integer Linear Programming (MILP) model to study the impact of big data's volume and variety on network power saving carrying 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 PNs in the ISP network centers that host the IP over WDM nodes. The volume scenarios captured generic results that show how the processing capability of the PNs dictates the big data volume that exists in SPNs, IPNs and DCs. 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2787624, IEEE Transactions on Network and Service Management

energy efficiency compared to the DCs equipment led to lower utilization of our approach. Furthermore, we analyzed the 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 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 utilize 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.

REFERENCES

[1] P. Zikopoulos and C. Eaton, Understanding big data: Analytics for enterprise class hadoop and streaming data: McGraw-Hill Osborne Media, 2011.
[2] W. "Worldometers - real time world statistics", 2016. [Online]. Available: http://www.worldometers.info/. [Accessed: 09- Nov- 2016].
[3] 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].
[4] K. Cukier, Data, data everywhere: A special report on managing information: Economist Newspaper, 2010.
[5] F. Brian, "Software crisis 2.0," Computer, vol. 45, pp. 89-91, 2012.
[6] 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.
[7] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Green IP over WDM networks with data centers," Journal of Lightwave Technology, vol. 29, pp. 1861-1880, 2011.
[8] 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.
[9] 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, pp. 2078-2086, 2015.
[10] 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, pp. 36-42, 2015.
[11] L. Zhang and Z. Zhu, "Spectrum-efficient anycast in elastic optical inter-datacenter networks," Optical Switching and Networking, vol. 14, pp. 250-259, 2014.
[12] 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.
[13] W. Chen, I. Paik, and Z. Li, "Cost-Aware Streaming Workflow Allocation on Geo-Distributed Data Centers," IEEE Transactions on Computers, vol. 66, pp. 256-271, 2017.
[14] 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), 2014 Second International Conference on, 2014, pp. 1-8.
[15] C. Jayalath, J. Stephen, and P. Eugster, "From the cloud to the atmosphere: running mapreduce across data centers," Computers, IEEE Transactions on, vol. 63, pp. 74-87, 2014.
[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, pp. 1261-1281, 2014.
[17] 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.
[18] 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.
[19] A. M. Al-Salim, T. El-Gorashi, A. Q. Lawey, and J. M. Elmirghani, "Greening Big Data Networks: Velocity Impact," IET Optoelectronics. 2017.
[20] A. M. Al-Salim, T. El-Gorashi, A. Q. Lawey, and J. M. Elmirghani, "Greening Big Data Networks: Veractiy Impact," IEEE/ACM Transactions on Networking, Submitted on Nov. 2016.
[21] 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.
[22] G. Shen and R. S. Tucker, "Energy-minimized design for IP over WDM networks," Journal of Optical Communications and Networking, vol. 1, pp. 176-186, 2009.
[23] 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, pp. 1694-1705, 2012.
[24] 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, pp. 755-768, 2012.
[25] C. H. Papadimitriou and K. Steiglitz, Combinatorial optimization: algorithms and complexity: Courier Corporation, 1998.
[26] A. C.-C. Yao, "New algorithms for bin packing," Journal of the ACM (JACM), vol. 27, pp. 207-227, 1980.
[27] R. G. Gallager, "Distributed minimum hop algorithms,"Massachusetts inst of tech cambridge lab for information and decision systems", 1982.
[28] 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.
[29] 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, pp. 149-167, 2011.
[30] 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.
[31] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, pp. 107-113, 2008.
[32] C. J. Ter Braak and I. C. Prentice, "A theory of gradient analysis," Advances in ecological research, vol. 18, pp. 271-317, 1988.
[33] 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.
[34] 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, pp. 1765-1775, 2011.
[35] 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, pp. 4209-4225, 2014.
[36] E. N. Power, "Energy logic: Reducing data center energy consumption by creating savings that cascade across systems," White Paper of Emerson Electric Co, 2009.
[37] 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.
[38] 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.
[39] 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.
[40] 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.