IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 29, 2020, accepted August 25, 2020, date of publication August 31, 2020, date of current version September 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020744

Energy Efficient Distributed Processing for IoT

BARZAN A. YOSUF, (Member, IEEE), MOHAMED MUSA, (Member, IEEE), TAISIR ELGORASHI, AND JAAFAR ELMIRGHANI, (Senior Member, IEEE)
School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, U.K.

Corresponding author: Barzan A. Yosuf (eenbayo@leeds.ac.uk)

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC), in part by the INTERNET Project under Grant EP/H040536/1, in part by the STAR Projects under Grant EP/K016873/1, and in part by the TOWS Project under Grant EP/S016570/1.

ABSTRACT In the near future, the number of Internet connected objects is expected to be between 26 - 50 billion devices. This figure is expected to grow even further due to the production of miniaturized portable devices that are lightweight, energy, and cost-efficient. In this article, the entire IoT-fog-cloud architecture is modeled, the service placement problem is formulated using Mixed Integer Linear Programming (MILP) and the total power consumption is jointly minimized for processing and networking. We evaluate the distributed processing paradigm for both the un-capacitated and capacitated design settings in order to provide solutions for the long-term and short-term basis, respectively. Furthermore, four aspects of the IoT processing placement problem are examined: 1) IoT services with non-splittable tasks, 2) IoT services with splittable tasks, 3) impact of processing overheads needed for inter-service communication and 4) deployment of special-purpose data centers (SP-DCs) as opposed to the conventional general-purpose data center (GP-DC) in the core network. The results showed that for the capacitated problem, task splitting introduces power savings of up to 86% compared to 46% with non-splittable tasks. Moreover, it is observed that the overheads due to inter-service communication greatly impacts the total number of splits. However much insignificant the overhead factor, the results showed that this is not a trivial matter and hence much attention needs to be paid to this area to make the best use of the resources that are available at the edge of the network.

INDEX TERMS IoT, energy efficiency, fog computing, MILP, IoT service placement, resource provisioning, resource allocation, Internet of Things.

I. INTRODUCTION

The number of objects connected to the Internet is growing at unprecedented levels, thereby leading to the concept of the Internet of Things (IoT) [1], [2]. In 2011, this number surpassed the world's population, and soon interconnected objects are expected to range between 26 billion to 50 billion devices [3], [4]. This increase is directly related to the technological advancement in the past decades that enabled the production of miniaturized portable devices that are lightweight, energy and cost-efficient together with the widespread use of the Internet and the added value organizations and individuals can gain from IoT devices. Multitudes of IoT applications have the potential to transform all aspects of life, some already exist while others are yet to be realized. The massive amounts of data produced if processed centrally by conventional clouds would lead to slow decision making and increased pressure on the already overwhelmed network. Autonomous vehicles, for example, are reported to generate data that is in the range of 1 GB per second [5]. Transporting all of this data to the cloud for processing is prohibitively costly in terms of bandwidth requirements and energy efficiency [6]. In the past, the focus of Information and Communication Technologies (ICT) was primarily fixated on performance only. Little or no attention was paid to the amount of power ICT based components consumed and consequently their adverse impact on our environment. The focus has now shifted towards energy efficiency, due to the rising cost of electricity, resource scarcity, and increasing emission of carbon dioxide (CO_2) [7]. It is reported that CO_2 emissions due to ICT technologies are increasing at an alarming rate of 6% per year. Given this growth rate, the Internet can become responsible for up to 12% of the global emissions by 2030, and cloud data centers which are

The associate editor coordinating the review of this manuscript and approving it for publication was Praveen Gunturi.

at the heart of the IoT are one of the major components of ICT [8].

In this direction, distributed processing has been proposed by industry and academia as an effective approach to curb the pressure imposed by the formidable scale of the IoT [9]. Fog computing, for instance, is a variant of distributed processing which promises to tackle the aforementioned challenges by utilizing the already available computational, storage, and networking resources for the processing of IoT data at the edge of the network [10]. In this article term fog is used interchangeably with distributed processing. Oftentimes, decision making can be made better and quicker if the collected data is processed closer to the source [11]. Currently, fog computing is still in its infancy and a standardized architecture has yet to be agreed. Thus alternative IoT architectures are increasingly being studied in the research community in terms of efficient resource management and the interplay between the edge devices (fog) and the core (cloud), since fog is regarded as a powerful complement to the cloud [12]. A proper resource management scheme is crucial in the fog, as services can be placed in an energy inefficient server or even further from the source node which results in higher communication latency [6]. Since contemporary fog devices have limited processing, storage, and communication capabilities, offloading services to more resourceful clouds or even fogs becomes a necessity [13]. It is expected that through cooperation between fogs and the centralized cloud, a more efficient and greener computing platform can be achieved [14].

Generally, in a fog architecture, a large number of devices exist at the edge of the network, which collectively provide enormous amounts of computational power, that, if used, may help in curbing the unnecessary data exchange between the IoT and the centralized cloud [15]. These devices are heterogeneous in terms of resources. This poses many challenges in the optimum design of architectures, protocols, and hardware of future IoT based networks. Hence, proper resource management and network design frameworks are needed [16]. These should take into account important dimensions such as but not limited to energy efficiency, due to its impact on our environment [17], resilience, due to mission-critical services [4], [17], [18] and end-device cooperation due to traffic bifurcations which lead to inter-service communication [9], [19]. Fog based solutions have been proposed to improve various performance metrics such as energy, latency, QoS, etc. through various approaches in terms of resource allocation [12], [20]–[25] and architectural design and planning [15], [26]–[32]. The reader is referred to the works in [17] and [33], for architectural design imperatives of fog networks and a detailed taxonomy of fog based solutions, respectively.

The work in this article benefits from our previous proposals for improving energy efficiency in a variety of areas such as big data processing in core networks [34], [35], core network optical architectures [36]–[38], server disaggregation in data centers [39], cloud content distribution in core networks [40], storage and caching in IPTV networks [41], network coding in core networks [42], NFV and big data in cellular networks [43], [44] and IoT virtualization, service embedding and fog in IoT based networks [45]–[49]. Moreover, the work in this article builds on the service placement optimization in [21] and extends the work by a) introducing a practical network architecture in which the entire IoT infrastructure is considered, which is composed of elements in the IoT, access, metro, and core layers, b) considering multiple IoT sites allowing to capture the impact of varying dynamics of source nodes' distribution, c) introducing a generic MILP formulation that is independent of the type of processing and/or networking technologies which allows for a holistic focus on energy-efficiency from the IoT network operators' perspective and developing a real-time heuristic to approximate the output of the MILP model, d) evaluating the impact of server packing through the prospect of task splitting, e) studying the impact of CPU overhead needed for inter-service communication on the total number of splits, f) deploying SP-DCs in the core as opposed to the traditional GP-DC, and g) casting the service placement problem into multiple design settings and evaluating solutions for both the long-term and short-term basis. One of our main contributions in this work is the inclusion of the optical core network to provide access to the Cloud DC which is currently not supported by any of the aforementioned studies in detail.

The remainder of this article is organized as follows.

Section II discusses the related work. Section III presents the proposed architecture followed by the MILP formulation of the service placement optimization problem. Section V presents and discusses the performance evaluation of the proposed model. The fog approach is evaluated in both un-capacitated and capacitated settings for IoT services with splittable and non-splittable tasks. Also, the performance of the fog approach is further examined by deploying SP-DC in the core network. Section VI focuses on the impact of CPU overhead on the total number of task splits. Section VII concludes the contributions of the paper.

II. RELATED WORK

The focus in the literature has shifted towards making the whole IoT infrastructure energy efficient [26] as opposed to optimizing only individual layers namely the device layer, access layer, or the cloud. The works in [27] and [28] proposed the use of Passive Optical Networks (PONs) to extend cloud and fog services closer to the user premises, respectively. Optical based networks are expected to become increasingly important to support edge and fog computing in the next decades. Although no algorithmic or optimization model was proposed, nevertheless, detailed discussions were provided on how the architecture in question can improve QoS and how different distributed fog resources located in the user premises can efficiently be managed. The authors of [26] proposed an energy-efficient IoT architecture in which sensors' sleep intervals are predicted based on their remaining battery level and as a result resources of the cloud can be better utilized by re-provisioning them when the sensory

nodes are in sleep mode. The main contribution of the work is centered around developing a mechanism to predict the sleep intervals of sensor nodes based upon certain sensor variables such as battery level and previous usage history.

The work in [29] mathematically models the entire fog network from the end terminals (TNs) to the cloud data centers located in the core network. The TN nodes sense data and transmit the same to the fog tiers, either to be processed by fog nodes or to be forwarded to the cloud for further analysis. The performance of the fog approach in provisioning for IoT applications is investigated by considering several dimensions such as power consumption, CO2 emissions, and service latencies in the fog network compared to the baseline cloud system. Their results indicate that the fog computing approach is only beneficial when there is a high number of latency-sensitive applications. Although fog computing was comprehensively studied, the authors made no mention of the practical networking or processing hardware used in obtaining their results. In another work, the authors of [15] compare the efficiencies of highly distributed edge devices called nano data centers that can host and distribute user content in a P2P fashion. These edge servers are comprised of Raspberry Pi's that are low power single-board computers. The authors investigate the system performance through a time-based and a flow-based power consumption model. For the devices that are highly shared by many users and services, the authors adopt a flow-based model whilst a time-based model is used for equipment that is close to end-users.

The work of [30] proposes a framework for cloudlet based network design and planning. The focus of the work is primarily centered around designing a network based on TDM-PON to optimize the network infrastructure cost whilst meeting latency constraints. The problem is formulated as a Mixed Integer Non Linear Programming (MINLP) model which is utilized to identify efficient cloudlet placement locations and optimal assignment of ONUs to cloudlets. The feasibility of the proposed model is evaluated against urban, suburban, and rural scenarios, which guide the installation and maintenance costs. In another work [31], a generic fiber-wireless architecture is proposed which supports the coexistence of the centralized cloud and distributed mobile edge computing (MEC) for IoT connectivity. A distributed game-theoretic algorithm is developed to support collaborative computational offloading between the cloud and MEC. Numerical results show very low energy consumption is achieved compared to the baseline which is the optimal case that cannot be realized in practice; hence the distributed approach is used to reduce complexities. The authors of [32] put forth a capacity planning framework that improves the resource utilization of a hierarchical edge cloud network whilst simultaneously meeting QoS requirements in terms of response delay. They do this by taking advantage of diverse demands for CPU, GPU, and network resources.

The work in [21] formulates the service distribution problem in an IoT-Cloud architecture using a linear program whose solution results in the optimum placement of IoT service functions and the routing of network flows across a multi-layer architecture consisting of devices, access, and cloud layers. The total energy consumption is minimized whilst meeting the end-user latency demands. In another work [20], the service allocation problem is formulated as an integer programming optimization, whose objective function is to minimize the total latency experienced by IoT services, subject to capacity constraints at the various layers of the proposed fog architecture. The IoT service requests are generic, ranging between 10 – 50 homogenous requests. The delay is minimized by placing the less demanding services near IoT devices whilst the medium and high demanding services are placed higher up the fog network. In their work, IoT devices have been excluded from hosting any type of data processing.

Similarly, the authors in [22] propose a generic algorithm for the placement of IoT services in a fog-cloud framework. The IoT services are considered as multiple modules that are collectively used to deliver a full application. A specific algorithm is used to efficiently deploy application modules dynamically across the fog-cloud infrastructure close to the source terminals in the fog layer. The performance of the proposed solution is addressed through the evaluation of latency and efficient resource utilization and it is claimed that it can be extended to include further design dimensions. In [23], an Integer Linear Program (ILP) is proposed to model the problem of resource provisioning from the perspective of service providers, in the context of the heterogeneous Internet of Things, where the objective function is to minimize the total monetary costs subject to capacity and latency budgets. The heterogeneity of IoT is modeled through unique profiling of applications and as such four different types of applications are considered. The topology considered comprises of a Metropolitan Area Network (MAN) and consists of two hierarchical levels of interconnected rings. The results indicated that the total operational cost is directly impacted by the application's computational complexity, compression factor, and latency budget, coupled with proportions of local traffic versus global traffic. The authors in [12] put forth a convex optimization model that addresses the delay-power trade-off in a cloud-fog architecture which consists of four subsystems. The work demonstrated that compromising modestly on computational resources to save communication bandwidth and reduce transmission latency, fog computing platforms can significantly complement the performance of cloud computing. The proposed work has not considered the impact of local computation using the devices in the IoT layer.

The authors of [24] model the IoT service placement in a practical testbed using an ILP formulation by considering several objective functions that address service latency, service migrations, and energy efficiency. The optimization model is executed iteratively to allow for the retention of the objective values of previously executed models, thus, the feasibility region continuously decreases since iterations must satisfy previous results. The approach is generic and can be adapted to other resource placement problems. Their results show that for real-time services, latency becomes important

TABLE 1. A comparison of the contributions of this paper with that of the related works in the literature.

Ref	Processing Layer			Optimization Consideration				
	Cloud	Fog	IoT	Energy	QoS	Task Splitting	Inter-Service Communication	Practical Network
[12]	√	√	–	√	√	–	–	–
[20]	√	√	–	–	√	–	–	–
[21]	√	√	√	√	√	–	–	–
[22]	√	√	√	–	√	–	√	–
[23]	√	√	–	–	√	–	–	√
[24]	√	√	√	√	√	–	√	√
[25]	–	√	–	–	√	–	–	–
[26]	√	√		√	√	–	–	√
[27]	–	√	–	–	–	–	–	√
[28]	√	√	–	–	√	–	–	–
[29]	√	√	√	√	√	–	–	–
[15]	√	√	–	–	–	–	–	√
[30]	–	√	–	–	√	–	–	√
[31]	√	√		√	√	–	–	√
[32]	√	√	–	–	√	–	–	–
[50]	–	–	√	√	√	–	–	√
[51]		–	√	√	√	–	–	√
This work	√	√	√	√	–	√	√	√

and thus services are processed on the nearest fog, while the latency tolerant services can be offloaded to the distant cloud as energy consumption becomes the priority.

Contrary to the aforementioned studies, the authors of [25] approach the resource allocation problem of fog computing from a different angle, which involves fog nodes applying a profit-sharing policy to maximize service providers' profit subject to QoS constraints outlined by short-term SLAs. Reinforcement learning and meta-heuristic methods were utilized to find the optimal network configurations. The authors of [50] address the energy efficiency of data aggregations methods by proposing an improved duty cycling algorithm and comparing its performances to multiple existing approaches. The proposed approach utilizes two residual energy thresholds, one is used for the entire network and another one for the path of the Mobile Sink Node (MSN) which in this case is the base station. The authors state merits in terms of several metrics such as energy consumption, residual energy, throughput, and processing time. The work in [51] addresses the challenges posed by routing in low-power and lossy networks in the Internet of Things by proposing a priority-based routing methodology. The proposed approach is reported to outperform the QRP routing method which is one of the most successful routing techniques adopted in the IoT. In Table 1, the reviewed studies are compared to the work in this article in terms of a number of contribution metrics.

III. THE EVALUATED DISTRIBUTED PROCESSING ARCHITECTURE

We begin by introducing and describing each layer of the evaluated fog architecture depicted in Figure 1. It comprises of four main processing layers in the optimization model, namely the IoT Devices, Access Fog (AF), Metro Fog (MF), and the Cloud DC (DC).

A. IoT DEVICES (IoT)

The bottom-most layer of the proposed architecture comprises of the IoT devices. These devices are smart, wireless nodes that are used to collect data and transmit the same via the connected access point (AP) to the next layer for processing and analysis, if local resources are insufficient [52]. A Wi-Fi link is considered between the devices and the APs due to its support for high data rates, which is particularly suitable for data-intensive applications as opposed to the other wireless technologies such as Bluetooth, Zigbee, LoRa, etc.

B. CPE FOG (CF)

The Customer Premises Equipment Fog (CF) layer consists of stationary processing units with processing capabilities usually higher than those found in the IoT layer [53]. The processing devices are connected to ONUs of the PON access network [15]. A PON deployment is used due to its suitability for data-intensive applications as they provide high bitrates, relatively low-cost, and high scalability [6]. PON has already paved the way for providing high-speed broadband access to end users through various flavors of FTTx. The ONUs are equipped with internal switches to provide connectivity to the CF servers and can route traffic to the higher layers of the architecture. Small organizations or even end-users can deploy their fog units at locations such as APs, routers, gateways, etc.

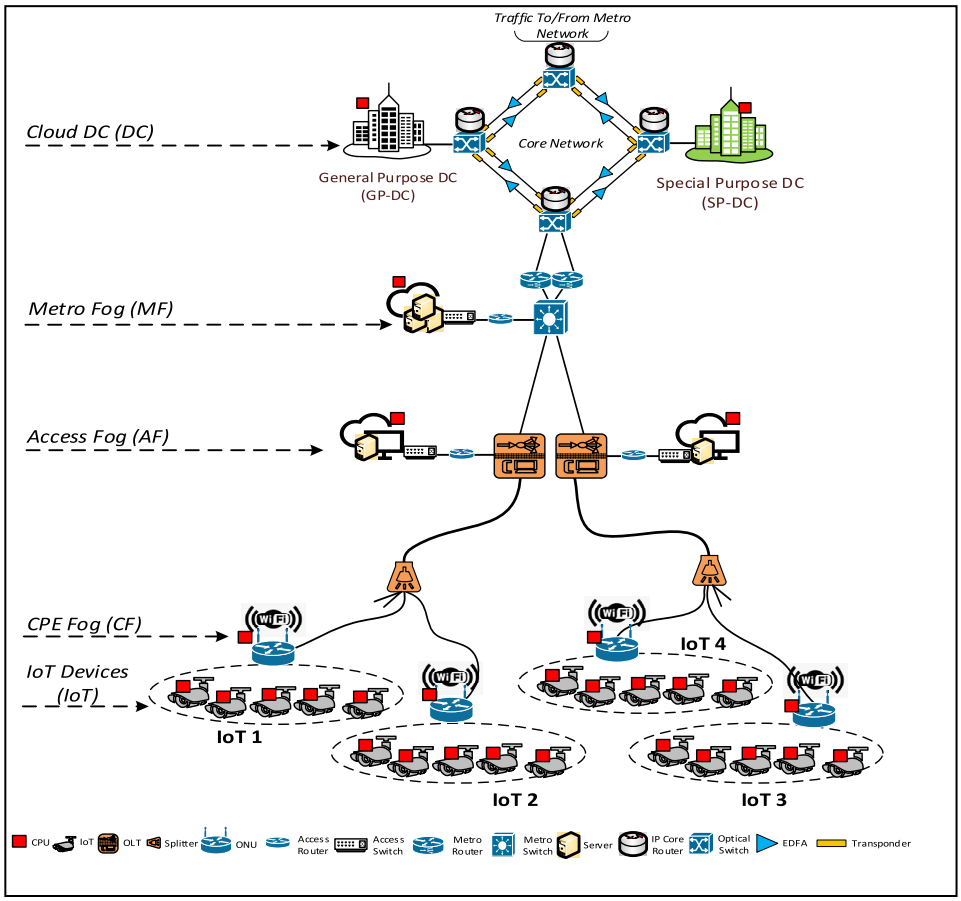


FIGURE 1. The evaluated PON-based IoT architecture supported by fog and cloud Infrastructures.

C. ACCESS FOG (AF)

The third layer is still part of the PON access network; however, it differs in terms of processing capability. Several high-end servers are used to form a fog collocated with the OLT [27], [47]. Thus, a substantial amount of service demands aggregated from the ONUs can be hosted and processed on the fog connected to the OLT's Ethernet input. However, the number of servers that can be collocated with the OLT is constrained by space limitations as these devices are normally enclosed in street cabinets and/or small offices. Therefore, more intensive service demands will need to be relayed to the next layer for processing.

D. METRO FOG (MF)

The metro network consists of a high-capacity Ethernet switch and a couple of edge routers that act as a gateway to the cloud data centers (DCS) via the core network. The Ethernet switch in the metro is usually found in facilities that provide access to public clouds and is primarily used for traffic aggregation from one or more access networks. The edge routers usually perform traffic management and authentication [6]. The computational resources available to the metro fog (MF) are substantially higher in comparison to the lower fog layers due to the number of users and services it supports, however it still is incomparable to the cloud DC [54].

E. CLOUD DC (DC)

The cloud layer comprises of a set of data centers that are accessed via the core network. The core network uses IP/WDM technology and it consists of two layers, the IP layer and the optical layer. In the IP layer, an IP core router is deployed at each node to aggregate network traffic from the metro routers. The optical layer is used to interconnect the IP core routers through optical switches and IP/WDM technologies such as EDFAs, transponders, and regenerators. Two types of data centers are considered: 1) a general-purpose data center (GP-DC), and 2) a special-purpose data center (SP-DC). Due to space allowances, cloud DCs are usually deployed to support a large number of applications and services, hence such clouds have significant storage and processing resources [4]. Motivated by the sheer computational power of Graphical Processing Units (GPUs) as well as the breakthrough performances in terms of power consumption efficiencies for visual-based deep learning algorithms, it is of interest to investigate the impact of deploying such servers inside DCs connected to the core. A SP-DC only performs a specific service i.e. visual processing. In contrast, the

general-purpose data center (GP-DC) is designed to execute a range of generic services. Therefore, it is not as power-efficient as the SP-DC. A leading manufacturer of CPUs and GPUs, Nvidia, has reported GPUs to be at least 10 times more efficient than CPUs in terms of energy expended per instruction.

IV. MILP MODEL

In this section, we develop a MILP model to minimize the total power consumption of the proposed architecture shown in Figure 1, by optimizing both the placement of IoT service demands. Each demand is characterized by a tuple $D(CPU, BW)$, where CPU is the amount of processing requirement in Million Instructions Per Second (MIPS) and BW is the serice data rate in Mbps. The IoT network is modeled as a graph $G(N, L)$, where N is the set of all nodes in the network, and L is the set of bidirectional links connecting those nodes. A subset $I \subset N$ represents the set of the IoT devices in the considered network, whilst a subset $S \subseteq I$ acts as source nodes. A subset of nodes, where $P \subset N$ acts as processing nodes. The processing node $d \in P$ has a maximum computational capacity $\left(C_d^{(CPU)}\right)$ measured in MIPS. Also, each link (m, n), where $m \in N$ and $n \in N_m$, has a maximum bit rate (BR) measured in bps. Before introducing the model, we define the sets, parameters, and variables used, as follows:

Sets:

$\mathbb{N}$	Set of all nodes.
$\mathbb{N}_m$	Set of neighbor nodes of node m in the proposed architecture
$\mathbb{C}$	Set of core nodes in the IP/WDM network, where $\mathbb{C} \subset \mathbb{N}$.
$\mathbb{O}$	Set of ONUs in the PON network, where $\mathbb{O} \subset \mathbb{N}$.
$\mathbb{OLT}$	Set of OLTs in the PON network, where $\mathbb{OLT} \subset \mathbb{N}$.
$\mathbb{MS}$	Set of metro switches, where $\mathbb{MS} \subset \mathbb{N}$.
$\mathbb{MR}$	Set of metro routers, where $\mathbb{MR} \subset \mathbb{N}$.
$\mathbb{DC}$	Set of data center nodes, where $\mathbb{DC} \subset \mathbb{N}$.
$\mathbb{I}$	Set of all IoT devices, where $\mathbb{I} \subset \mathbb{N}$.
$\mathbb{P}$	Set of nodes with processing devices, where $\mathbb{P} \subset \mathbb{N}$ and $\mathbb{P} = \mathbb{I} \bigcup \mathbb{O} \bigcup \mathbb{OLT} \bigcup \mathbb{MS} \bigcup \mathbb{DC}$.
$\mathbb{S}$	Set of IoT devices acting as source nodes where $\mathbb{S} \subset \mathbb{I}$.

Core Network Parameters:

MPR	Maximum power consumption of an IP router port.
MPT	Maximum power consumption of a transponder in the core network.
MPE	Maximum power consumption of an EDFA in the core network.
$MPOS$	Maximum power consumption of an optical switch in the core network.
MPG	Maximum power consumption of a regenerator in the core network.
IPR	Idle power consumption of an IP router port in the core network.
IPT	Idle power consumption of a transponder in the core network.
IPE	Idle power consumption of an EDFA in the core network.
$IPOS$	Idle power consumption of an optical switch in the core network.
IPG	Idle power consumption of a regenerator in the core network.
B	Maximum bit rate of a single wavelength.
W	Number of wavelengths in a fibre in the core network.
ER	Energy per bit of a router port, where $ER = \left(\frac{MPR-IPR}{B}\right)$.
ET	Energy per bit of a transponder, where $ET = \left(\frac{MPT-IPT}{B}\right)$.
EE	Energy per bit of the EDFAs, where $EE = \left(\frac{MPE-IPE}{B}\right)$.
EOS	Energy per bit of optical switches, where $EOS = \left(\frac{MPOS-\text{IPOS}}{B}\right)$.
EG	Energy per bit of regenerators, where $EG = \left(\frac{MPG-IPG}{B}\right)$.
D_{mn}	Distance between two core nodes m and n, where $m, n \in C$.
SE	Span distance between two EDFAs.
SG	Span distance between two regenerators.
A_{mn}	Number of EDFAs used on each fiber in the core network from node $m \in C$ to $n \in C$, $A_{mn} = \left\lfloor \left(\left(\frac{D_{mn}}{SE}\right) - 1 \right) \right\rfloor + 2$.
R_{mn}	Number of regenerators used between core node $m \in C$ and core node $n \in C$, $R_{mn} = \left\lfloor \left(\frac{D_{mn}}{SG}\right) - 1 \right\rfloor$.
$PUEc$	Power Usage Effectiveness of IP/WDM core network node.

Cloud Data Center Parameters:

$MPDS$	Maximum power consumption of Cloud DC switch.
$IPDS$	Idle power consumption of Cloud DC switch.
BDS	Bit rate of Cloud DC switch.
EDS	Cloud DC switch energy per bit, where $EDS = \left(\frac{MPDS-\text{IPDS}}{BDS}\right)$.
$MPDR$	Maximum power consumption of Cloud DC router.
$IPDR$	Idle power consumption of Cloud DC router.
BDR	Cloud DC router bit rate.
EDR	Energy per bit of a Cloud DC router, where $EDR = \left(\frac{MPDR-IPDR}{BDR}\right)$.
$PUEd$	Power Usage Effectiveness of DC node, for processing and networking.

Metro Network and Fog Parameters:

$MPMS$	Maximum power consumption of a metro switch.

$IPMS$ Idle power consumption of a metro switch.
BMS Bit rate of a metro switch.
EMS Metro switch energy per bit, where $EMS = \left(\frac{MPMS - IPMS}{BMS}\right)$.
$MPMFS$ Maximum power consumption of a metro fog switch.
$IPMFS$ Idle power consumption of a metro fog switch.
$BMFS$ Bit rate of a metro fog switch.
$EMFS$ Metro fog switch energy per bit, where $EMFS = \left(\frac{MPMFS - \text{IPMFS}}{BMFS}\right)$.
$MPMR$ Maximum power consumption of a metro router.
$IPMR$ Idle power consumption of a metro router.
BMR Bit rate of a metro router.
EMR Metro router energy per bit, where $EMR = \left(\frac{MPMR - IPMR}{BMR}\right)$.
$MPMFR$ Maximum power consumption of a metro fog router.
$IPMFR$ Idle power consumption of a metro fog router.
$BMFR$ Bit rate of a metro fog router.
$EMFR$ Metro fog router energy per bit, where $EMFR = \left(\frac{MPMFR - IPMFR}{BMFR}\right)$..
$PUEm$ Power Usage Effectiveness of a metro node, for processing and networking.
$\mathcal{R}$ Metro router port redundancy.

Access Network and Fog Parameters:

$MPOT$ Maximum power consumption of OLT in the PON network.
$IPOT$ Idle power consumption of OLT in the PON network.
BOT Bit rate of OLT in the PON network.
EOT OLT router energy per bit, where $EOT = \left(\frac{MPOT - IPOT}{BOT}\right)$.
MPO Maximum power consumption of an ONU in the PON network.
IPO Idle power consumption of an ONU in the PON network.
BO Bit rate of the WiFi interface of an ONU in the PON network.
EO ONU energy per bit, where $EO = \left(\frac{MPO - IPO}{BO}\right)$.
$MPAFS$ Maximum power consumption of an access fog switch.
$IPAFS$ Idle power consumption of an access fog switch.
$BAFS$ Bit rate of an access fog switch.
$EAFS$ Access fog switch energy per bit, where $EAFS = \left(\frac{MPAFS - IPAFS}{BAFS}\right)$.
$MPAFR$ Maximum power consumption of an access fog router.
$IPAFR$ Idle power consumption of an access fog router.
$BAFR$ Bit rate of an access fog router.
$EAFR$ Access fog router energy per bit, where $EAFR = \left(\frac{MPAFR - IPAFR}{BAFR}\right)$.
$MPCFS$ Maximum power consumption of CPE fog switch.
$IPCFS$ Idle power consumption of a CPE fog switch.
$BCFS$ Bit rate of a CPE fog switch.
$ECFS$ CPE fog switch energy per bit, where $ECFS = \left(\frac{MPCFS - IPCFS}{BCFS}\right)$.
$PUEa$ Power Usage Effectiveness of an access fog node, for processing and networking.

IoT Devices' Parameters:

$MPIT$ Maximum power consumption of an IoT transceiver.
$IPIT$ Idle power consumption of an IoT transceiver.
BIT Bit rate of the Wi-Fi interface of an IoT device.
EIT IoT Wi-Fi interface energy per bit, where $EIT = \left(\frac{MPIT - \text{IPIT}}{BIT}\right)$.

Processing Devices' Parameters:

MP_d Maximum power consumption of processing device $d \in \mathbb{P}$, in Watts.
IP_d Idle power consumption of processing device $d \in \mathbb{P}$, in Watts.
CAP_d Maximum capacity of processing device $d \in \mathbb{P}$ in MIPS.
EPI_d Energy per instruction of processing device $d \in \mathbb{P}$, where $EPI_d = \left(\frac{\text{MP}_\text{d} - IP_d}{CAP_d}\right)$.

Application Parameters:

CPU_s Processing request of IoT node $s \in \mathbb{S}$ in MIPS.
BW_s Traffic request of IoT node $s \in \mathbb{S}$ in Mbps.
δ Portion of the idle power of equipment attributed to the application.
K Number of sub-tasks an IoT service can be split into.
Δ Number of MIPS required to process 1 Mb of traffic.
$C_{m,n}$ Capacity of link $(m, n) \in \mathbb{N}$ where $(m, n) \in \mathbb{C}$.
M Large enough number.

Variables:

$\lambda^{s,d}$ Traffic demand between IoT source node $s \in \mathbb{S}$ and processing device $d \in \mathbb{P}$.
$\lambda^{s,d}_{m,n}$ Traffic flow between IoT source node $s \in \mathbb{S}$ and processing device $d \in \mathbb{P}$, traversing link (m, n), where $m \in \mathbb{N}$, $n \in \mathbb{N}_m$.
λ_d Volume of traffic aggregated by node $d \in \mathbb{N}$.
$\mathcal{B}_m$ $\mathcal{B}_m = 1$, if network node $m \in \mathbb{N}$ is activated, otherwise $\mathcal{B}_m = 0$.
θ_d Traffic in node $d \in \mathbb{P}$ for processing, where $\theta_d = \lambda_\text{d}\Omega_d$.
$\Gamma_{m,n}$ $\Gamma_{m,n} = 1$, if core network link m, n, where $m \in \mathbb{C}$, $n \in (\mathbb{N}_m \cap \mathbb{C})$ is activated, otherwise $\Gamma_{m,n} = 0$.

$\rho^{s,d}$ Processing demand of IoT source node $s \in \mathbb{S}$ hosted at processing device $d \in \mathbb{P}$.

$\Omega^{s,d}$ $\Omega^{s,d} = 1$, if processing demand of IoT source node $s \in \mathbb{S}$ is hosted at destination node $d \in \mathbb{P}$, otherwise $\Omega^{s,d} = 0$.

Ω^{d} $\Omega^{d} = 1$, if processing node $d \in \mathbb{P}$ is activated, otherwise $\Omega^{d} = 0$.

$\mathcal{N}_d$ Number of processing servers activated at node $d \in \mathbb{P}$.

$W_{m,n}$ Number of wavelengths used in fiber link (m, n) in the core network, where link $m, n \in \mathbb{C}$.

$F_{m,n}$ Number of fibers used on link $m, n \in \mathbb{C}$.

Ag_m Number of aggregation router ports activated at IP node $m \in \mathbb{C}$.

The total power consumption of the fog architecture depicted in Figure 1 is composed of three parts: A) Network Power Consumption, B) Processing Power Consumption and C) Intra Server Network Power Consumption. While it is valid to assume that, the desirable power consumption profile should be fully load proportional, however in practical circumstances, this is not the case. It is reported in [55], that almost all devices adopt a linear power profile that consists of an idle and proportional part as depicted in Figure 2. With the former, power is consumed as soon as the device is activated regardless of the load however the latter dependents on various parameters such as frequency, voltage, or workload. In practice, idle power draws a large proportion of the maximum power of a networking/ processing device and hence it cannot be ignored. Since the devices involved in the considered architecture span multiple heterogeneous layers, it becomes a necessity to fairly represent the utilization characteristics of these devices. For example, high-capacity networking elements such as OLTs, metro/core routers and switches are used by many other types of applications in addition to the IoT and it would not make a fair evaluation if the total idle power consumption of these devices were wholly attributed to a small number of IoT services. The total power consumption (TPC) considering the linear profile with idle power consumption of a networking or processing device is calculated using equation (1):

$$TPC = \left(\frac{Pmax - Pidle}{C}\right)\lambda + Pidle \tag{1}$$

where $Pidle$ is the idle power consumption of the device which is consumed as soon as the device is activated regardless of the load λ and ($Pmax$) is the maximum power consumption of the device, when it is 100% utilized at full capacity C (either in bps or MIPS). The proportional section of the power profile model for networking devices is expressed as energy per bit and likewise, for processing, it is expressed as energy per instruction.

A. NETWORK POWER CONSUMPTION (Net_PC)

Under the non-bypass light path approach [56], the core network's IP/WDM total power consumption is composed of:

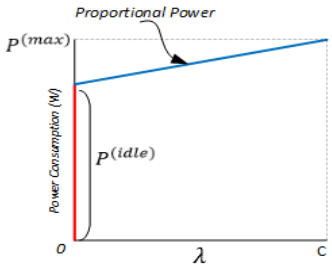


FIGURE 2. Linear power profile with idle part.

The power consumption of router ports:

$$PUEc\left[\sum_{m\in\mathbb{C}} \mathrm{ER}.\lambda_m + \sum_{m\in\mathbb{C}} (\delta.IPR.(Ag_m + \sum_{n\in(\mathbb{N}_m\cap\mathbb{C})} W_{m,n}))\right] \tag{2}$$

The power consumption of transponders:

$$PUEc\left[\sum_{m\in\mathbb{C}} \mathrm{ET}.\lambda_m + \sum_{m\in\mathbb{C}}\sum_{n\in(\mathbb{N}_m\cap\mathbb{C})} \delta.IPT.W_{m,n}\right] \tag{3}$$

The power consumption of EDFAs:

$$PUEc\left[\sum_{m\in\mathbb{C}} EE.\lambda_m.A_{m,n}.F_{m,n} + \sum_{m\in\mathbb{C}}\sum_{n\in(\mathbb{N}_m\cap\mathbb{C})} \delta.IPE.A_{m,n}.F_{m,n}\right] \tag{4}$$

The power consumption of optical switches:

$$PUEc\left[\sum_{m\in\mathbb{C}} EOS.\lambda_m + \sum_{m\in\mathbb{C}} \delta.IPOS.\mathcal{B}_m\right] \tag{5}$$

The power consumption of regenerators:

$$PUEc\left[\sum_{m\in\mathbb{C}} EG.\lambda_m.Rg_{m,n}.W_{m,n} + \sum_{m\in\mathbb{C}}\sum_{n\in(\mathbb{N}_m\cap\mathbb{C})} IPG.Rg_{m,n}.W_{m,n}\right] \tag{6}$$

The metro network's power consumption consists of the power consumption of metro routers and switch, which is given as:

$$PUEm\left[\mathcal{R}\sum_{m\in\mathbb{MR}} EMR.\lambda_m + \mathcal{R}\sum_{m\in\mathbb{MR}} \delta.IPMR.\mathcal{B}_m + \sum_{m\in\mathbb{MS}} EMS.\lambda_m + \sum_{m\in\mathbb{MS}} \delta.IPMS.\mathcal{B}_m\right] \tag{7}$$

The access network's power consumption consists of the power consumption of OLT and ONU devices, which is given as:

$$PUEa\left[\sum_{m\in\mathbb{OT}} EOT.\lambda_m + \sum_{m\in\mathbb{OT}} \delta.IPOT.\mathcal{B}_m + \sum_{m\in\mathbb{O}} EO.\lambda_m + \sum_{m\in\mathbb{O}} IPO.\mathcal{B}_m\right] \quad (8)$$

The IoT devices' communication interfaces power consumption is given as:

$$\sum_{m\in\mathbb{I}} EIT.\lambda_m + \sum_{m\in\mathbb{I}} IPIT.\mathcal{B}_m \quad (9)$$

B. PROCESSING POWER CONSUMPTION (Pr_PC)

The total power consumption of the processing devices (or servers) is composed of:

The processing power consumption of IoT devices:

$$\sum_{s\in\mathbb{S}}\sum_{d\in\mathbb{I}} EPI.\rho^{s,d} + \sum_{d\in\mathbb{I}} IP_d.\mathcal{N}_d \quad (10)$$

The processing power consumption of CPE fog (CF) servers:

$$\sum_{s\in\mathbb{S}}\sum_{d\in\mathbb{O}} EPI_d.\rho^{s,d} + \sum_{d\in\mathbb{O}} IP_d\mathcal{N}_d \quad (11)$$

The processing power consumption of access fog (AF) servers:

$$PUEa\left[\sum_{s\in\mathbb{S}}\sum_{d\in\mathbb{OLT}} EPI_d.\rho^{s,d} + \sum_{d\in\mathbb{OLT}} IP_d.\mathcal{N}_d\right] \quad (12)$$

The processing power consumption of metro fog (MF) servers:

$$PUEm\left[\sum_{s\in\mathbb{S}}\sum_{d\in\mathbb{MS}} EPI_d.\rho^{s,d} + \sum_{d\in\mathbb{MS}} IP_d.\mathcal{N}_d\right] \quad (13)$$

The processing power consumption of data center (DC) servers:

$$PUEd\left[\sum_{s\in\mathbb{S}}\sum_{d\in\mathbb{DC}} EPI_d.\rho^{s,d} + \sum_{d\in\mathbb{DC}} IP_d.\mathcal{N}_d\right] \quad (14)$$

C. INTRA SERVER NETWORK POWER CONSUMPTION (Intra_PC)

The cloud DCs' network power consumption is composed of the power consumption of cloud DC routers and switches:

$$PUEd\left[\sum_{d\in\mathbb{DC}} EDR.\theta_d + \sum_{d\in\mathbb{DC}} \delta.IPDR.\Omega^d + \sum_{d\in\mathbb{DC}} (EDS.\theta_d) + \sum_{d\in\mathbb{DC}} \left(\delta.IPDS.\Omega^d\right)\right] \quad (15)$$

The metro fog network power consumption consists of power consumption of metro fog routers and switches, which is given as:

$$PUEm\left[\sum_{d\in\mathbb{MS}} EMFR.\theta_d + \sum_{d\in\mathbb{MS}} \delta.IPFMR.\Omega^d + \sum_{d\in\mathbb{MS}} EMFS.\theta_d + \sum_{d\in\mathbb{MS}} \delta.IPFMS.\Omega^d\right] \quad (16)$$

The CF network power consumption consists of power consumptions of CF switches which is given as:

$$\sum_{d\in\mathbb{O}} ECFS.\theta_d + \sum_{d\in\mathbb{O}} IPCFS.\Omega^d \quad (17)$$

The MILP model's objective function is to minimize the total power consumption as follows:

Minimize: $Net\_PC + Pr\_PC + Intra\_PC$

Subject to the following constraints:

$$\sum_{n\in\mathbb{N}_m} \lambda^{s,d}_{m,n} - \sum_{n\in\mathbb{N}_m} \lambda^{s,d}_{n,m} = \begin{cases} \lambda_{s,d}, & m = s \\ -\lambda_{s,d}, & m = d \\ 0, & otherwise \end{cases} \quad \forall s \in \mathbb{S}, d \in \mathbb{P}, m \in \mathbb{N} : s \neq d. \quad (18)$$

Constraint (18) conserves traffic from a source node to a destination node in the considered topology depicted in Figure 1. It ensures that the total incoming traffic at a node is equal to the total outgoing traffic of that node; unless the node in question is either the source node or the destination node.

$$\sum_{d\in\mathbb{P}} \rho^{s,d} = CPU_s \quad \forall s \in \mathbb{S} \quad (19)$$

Constraint (19) ensures that processing service demand per IoT source node $s \in \mathbb{S}$ is met at a given destination node.

$$\rho^{s,d} \geq \Omega^{s,d} \quad \forall s \in \mathbb{S}, d \in P \quad (20)$$

$$\rho^{s,d} \leq M.\Omega^{s,d} \quad \forall s \in \mathbb{S}, d \in P \quad (21)$$

Constraints (20) and (21) are used to ensure that the binary variable $\rho^{s,d} = 1$ if destination node $d \in \mathbb{P}$ is activated to host the processing demand of source node $s \in \mathbb{S}$.

$$\sum_{d\in\mathbb{P}} \Omega^{s,d} \leq K \quad \forall s \in \mathbb{S} \quad (22)$$

Constraint (22) ensures that the number of sub-tasks a processing demand can be divided into is less than or equal to K, hence $K = 1$ implies no task splitting is allowed.

$$\mathcal{N}_d \leq \mathcal{V}_d \quad \forall d \in \mathbb{P} \quad (23)$$

Constraint (23) ensures that the number of servers activated at a processing node $d \in \mathbb{P}$, does not exceed the maximum available number of servers in that node.

$$\sum_{s\in\mathbb{I}} \Omega^{s,d} \geq \Omega^d \quad \forall d \in \mathbb{P} \quad (24)$$

$$\sum_{s\in\mathbb{I}} \Omega^{s,d} \leq M\Omega^d \quad \forall d \in \mathbb{P} \quad (25)$$

Constraints (24) and (25) are used to ensure that, the binary variable $\Omega^d = 1$ if processing node $d \in P$ is activated, otherwise $\Omega^d = 0$.

$$\lambda_m = \sum_{\substack{s\in\mathbb{S}:\\ m=s}} \sum_{d\in\mathbb{P}} \sum_{n\in\mathbb{N}_m} \lambda_{m,n}^{s,d} + \sum_{\substack{s\in\mathbb{S}:\\ m\neq s}} \sum_{\substack{d\in\mathbb{P}:\\ s\neq d}} \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{s,d} \quad \forall m \in \mathbb{S} \tag{26}$$

$$\lambda_m = \sum_{\substack{s\in\mathbb{S}:\\ m\neq s}} \sum_{\substack{d\in\mathbb{P}:\\ s\neq d}} \sum_{n\in\mathbb{N}_m} \lambda_{n,m}^{s,d} \quad \forall m \in (\mathbb{I}\cup\mathbb{O}\cup\mathbb{OLT}\cup\mathbb{MS}\cup\mathbb{MR}\cup\mathbb{DC}) \tag{27}$$

$$\lambda_m = \sum_{s\in\mathbb{S}} \sum_{\substack{d\in\mathbb{P}:\\ s\neq d}} \sum_{\substack{n\in\mathbb{N}_m:\\ n\in(\mathbb{N}_m\cap\mathbb{C})}} \lambda_{m,n}^{s,d} \quad \forall m \in \mathbb{C} \tag{28}$$

Constraint (26) gives the traffic generated or received by an IoT node with the first term representing its role as a source and the second term representing IoT node serving demands of other IoT nodes. Constraint (27) gives the traffic traversing/ received by a node of the access, metro, and cloud network. Constraint (28) gives the traffic traversing the core nodes.

$$\theta_d \leq M.\Omega^d \quad \forall d \in \mathbb{P} \tag{29}$$

$$\theta_d \leq \lambda_d \quad \forall d \in \mathbb{P} \tag{30}$$

$$\theta_d \geq \lambda_d - \left(1 - \Omega^d\right).M \quad \forall d \in \mathbb{P} \tag{31}$$

Constraints (29), (30) and (31) are used to linearize the non-linear equation $\lambda_d.\Omega_d$, where $d \in \mathbb{P}$. This ensures that traffic on a processing node $d \in \mathbb{P}$ is only accounted for if it is destined to that node for processing.

$$\lambda_m \geq \mathcal{B}_m \quad \forall m \in \mathbb{N} \tag{32}$$

$$\lambda_m \leq M\mathcal{B}_m \quad \forall m \in \mathbb{N} \tag{33}$$

Constraints (32) and (33) are used to ensure that, the binary variable $\mathcal{B}_m = 1$ if network node $m \in \mathbb{N}$ is activated, otherwise $\mathcal{B}_m = 0$.

$$\lambda^{s,d} = BW_s.\Omega_{s,d} \quad \forall s \in \mathbb{S}, d \in \mathbb{P} \tag{34}$$

Constraint (34) ensures that traffic is only directed to the destination node that is hosting a processing service.

$$\sum_{s\in\mathbb{S}} \sum_{\substack{d\in\mathbb{P}:\\ s\neq d}} \lambda_{m,n}^{s,d} \leq \mathbb{C}_{m,n} \quad \forall m \in (\mathbb{I}\cup\mathbb{O}\cup\mathbb{OLT}\cup\mathbb{MS}\cup\mathbb{MR}\cup\mathbb{DC}) : n \in \mathbb{N}_m \tag{35}$$

Constraint (35) ensures that the total traffic carried on link m, n, in the metro and access layer, does not exceed its capacity in Mbps.

$$Ag_m \geq \frac{\lambda_m}{B} \quad \forall m \in C \tag{36}$$

Constraint (36) gives the number of aggregation router ports at each IP/WDM node.

$$\sum_{s\in S} \sum_{\substack{d\in P:\\ s\neq d}} \lambda_{mn}^{sd} \leq W_{mn}B \quad \forall m \in C : n \in (C \cap N_m) \tag{37}$$

$$W_{mn} \leq WF_{mn} \quad \forall m \in C : n \in (C \cap N_m) \tag{38}$$

Constraints (37) and (38) represent the physical link capacity of the IP/WDM optical links. Constraint (37) ensures that the total traffic on a link does not exceed the capacity of a single wavelength while constraint (38) ensures the total number of wavelength channels does not exceed the capacity of a single fiber link.

V. PERFORMANCE EVALUATION

To evaluate the performance of the energy-efficient distributed processing models, we use an architecture that consists of 20 IoT devices that are divided into 4 groups uniformly, hence each group is connected to the PON network via a single ONU. The total number of IoT devices in each group is based on a representative home LAN network which typically connects a single to few users [63]. It is important to note that such IoT deployment represents only a snapshot of the evaluated architecture in space and time with limited number of generated workloads and IoT devices. With the expansion of the evaluated architecture to include realistic number of IoTs, the volume of the generated workloads increases too. Thus, the allocation scheme and the consumed power are expected to remain the same. We have assumed that the IoT devices are ''intelligent'' in nature in terms of their processing ability, hence service requests can be fulfilled through local computation. Table 2 – *Table 5*. contain the parameters of all the networking devices, processing servers, intra-processing networking devices, PUE values, and the IP/WDM core network elements, respectively.

TABLE 2. Network devices' data for the MILP model.

Device	Pmax(W)	Pidle(W)	δ	Gbps
IoT (WiFi)	0.56 [57]	0.34 [58]	-	0.1 [57]
ONU (WiFi)	15 [59]	9 [59]	-	0.3 [59]
OLT	1940 [60]	60 [60]	3%	8600 [60]
Metro Router Port	30 [61]	27	3%	40 [61]
Metro Ethernet Switch	470 [62]	423	3%	600 [62]
Metro Router Port Redundancy ($\mathcal{R}$)	2 [6]			

A. WORKLOAD DEFINITION

In our evaluations, we have made CPU requirements proportional to traffic (BW), such that, for every bit of traffic 1000 MIPS is required for processing. Although, it is beyond the scope of the work in this article, measuring CPU efficiency by MIPS is not an accurate benchmark, since different CPUs have different architectures, hence varied performances for the same task. Nevertheless, this does not stop us

TABLE 3. Processing servers' data.

Node	Device Model	P(W)	I(W)	GHz	k MIPS	W /MIPS	I/C
SP-DC Server	NVidia T4 GPU	75 [66]	45	1.25 [66]	1080	27.7μ	864
GP-DC Server	Intel Xeon E5-2680	130 [67]	78	2.7 [67]	108	481μ	5
MF Server	Intel X5675	95 [68]	57	3.06 [68]	73.44	517μ	4
AF Server	Intel Xeon E5-2420	95 [69]	57	1.9 [69]	34.2	1111μ	3
CF Server	RPi 3 Model B	12.5 [70]	2	1.2 [71]	2.4	4375μ	2
IoT Device	RPi Zero W	3.96 [70]	0.5	1 [72]	1	3460μ	1

TABLE 4. Intra-processing network devices' data.

Device	Pmax (W)	Pidle (W)	BR (Gb/s)	Eb (W/Gb/s)
CF Switch	1.78W [73]	0.36[73]	1.6[73]	0.89
AF Router	13W[61]	11.7	40[61]	0.03
AF Switch	210W[62]	189	240[62]	0.08
MF Router	13W[61]	11.7	40[61]	0.03
MF Switch	210W [62]	189	600[62]	0.04
DC LAN Router	30[61]	27	40[63]	0.08
DC LAN Switch	470[62]	423	600[62]	0.08

TABLE 5. PUE values used in the MILP model.

Network Layer	PUE
IoT Devices	1
CPE Fog (CF)	1
Access Fog ($PUEa$)	1.5
Metro Fog ($PUEm$)	1.4
Cloud DC ($PUEd$)	1.12 [79]
Core Network (($PUEc$))	1.5 [78]

from making a starting point by consulting the literature to obtain realistic values. In [64], the authors have reported that for a specific visual processing algorithm referred to as Analyze Then Compress (ATC), for a file of 10kB, 69.23 MIPS are required for processing for visual object recognition. Thus, through simple calculations we derived how many MIPS are required (Δ) to process 1Mb of traffic as follows, using equation (39):

$$\Delta = \frac{69.23}{0.08} \cong 865.4. \tag{39}$$

For the sake of simplicity and staying conservative, we assume that each 1Mb of traffic requires approximately 1000 MIPS for processing. As for the bandwidth requirement, we used an online tool to estimate the required data rates for different resolutions and this was estimated to be between 1 – 10 Mbps, which covers video resolutions between 1024×720 to 1600×1200 at 30 frames per second [65]. The CPU workload intensity is then calculated by multiplying the Δ by the amount of traffic. Thus, this makes the CPU demand proportional to the size of the traffic due to the assumption that the higher the traffic, the more features a video file will hold, thus more CPU instructions are required to process that file.

B. POWER CONSUMPTION DATA

The network data in Table 2 consist of the maximum and idle power consumption, bit rate (Gbps), and a portion of the idle power (δ) if it applies. We have made use of equipment datasheets, where possible, however, it is not always feasible to obtain this information, hence, we make realistic assumptions based on the literature. In terms of idle power consumption, based on [15], most high-capacity networking equipment such as metro/core routers and switches consume 90% of equipment's maximum power consumption. As for processing servers' idle power consumption, based on [74], we assume it is 60% of the maximum power consumption of the CPU. Moreover, we assume that IoT applications are only responsible for a portion of the maximum idle power consumption. This assumption is valid. For instance, metro switches are used to serve thousands of different users simultaneously, thus it would not make a fair analysis if all of *Pidle* was attributed to a specific application like the one considered in this article. Therefore, we make use of Cisco's visual networking index for the years 2017-2022 to estimate the total traffic of surveillance type application services. It is reported that, globally, 3% of all video traffic on the Internet is due to surveillance services, hence the value of δ attributed to the application in question is 3% [75]. The processing devices' input data are summarized in Table 3. To estimate the processing capacity of the servers in MIPS, we have made use of a technical benchmark, in which, it is reported that Intel high-end servers process 4 instructions/cycle (I/C) [76]. Thus, to determine the maximum capacity of a processing device we have used the following

$$IPS = clock \times \frac{I}{C} \tag{40}$$

where $\frac{I}{C}$ is the number of instructions a CPU can execute per clock cycle which is given in GHz. To differentiate between the types CPUs and their efficiencies, we set the $\frac{I}{C}$ of the MF server as a reference point. We use approximate modeling of CPU data based on the approach in [15]. The efficiency of the processing decreases as one moves down the network hierarchy (from the core to the edge). At those layers where multiple servers can be deployed, a networking infrastructure becomes a necessity to establish a LAN network between multiple active servers. Hence, we have used routers and switches accordingly to achieve this. We refer to these devices as intra processing networking equipment. Generally, lower layers have been assigned lower specification devices where applicable, for instance, an L3 metro switch is much more power-consuming than an L2 switch at the access. Table 4 summarizes the networking equipment used inside processing nodes.

C. POWER USAGE EFFECTIVENESS (PUE)

In our evaluations, PUE is not considered for IoT and ONU devices, as there is generally no cooling requirements for them [77]. The power usage effectiveness (PUE) is the ratio of the total power consumed by a facility (i.e. ISP networks, data centers) to the total power consumption of the equipment within the facility (i.e. servers, switches, routers, etc. In 2018, Google reported that one of its data centers is was operating at a PUE of 1.15. We make use of a report published in 2016 which estimates the PUE values of various data centers based on ''Space Type'' [94]. Within the report, it is shown that PUE values progressively decrease with the increase in the ''Space Type''. Similarly, we increase PUE progressively in the proposed network architecture since the largest ''Space Type'' is generally hyper-scale data centers connected to the core network. It is assumed that at the access and metro layers, processing and networking equipment have the same PUE. The PUE value of the core network is consistent with one of our previous works, which is 1.5 [78]. Table 5 is a summary of the PUE values used in the model.

We have considered the data center nodes to be only a single hop from the user traffic and the average distance between two neighboring core nodes is assumed to span 2010 km (estimated using google maps based on the AT&T US network topology) [80]. The power consumption of the core network elements is consistent with our previous work in [81] and all the parameters are summarized in Table 6.

TABLE 6. Input data of the core network elements for the MILP model.

Distance between two neighboring EDFAs (SE)	80 (km) [81]
Number of wavelengths in a fiber (W)	32 [81]
Bitrate of a wavelength (B)	40 Gb/s
Distance between two neighboring core nodes ($D_{m,n}$)	2500km
Maximum power consumption of a router port (MPR)	638 (W) [81]
Idle power consumption of a router port (IPE)	574.2 (W)
Energy per bit of a router port (ER)	1.6 W/Gb/s
Maximum power consumption of a transponder (MPT)	129 (W) [81]
Idle power consumption of a transponder (IPT)	116 (W)
Energy per bit of a transponder (ET)	0.32 (W/Gb/s)
Maximum power consumption of an EDFA (MPE)	11(W) [81]
Idle power consumption of an EDFA (IPE)	0.3 (W)
Energy per bit of an EDFA (EE)	0.03 (W/Gbps)
Maximum power consumption of an optical switch (MPO)	85 (W) [81]
Idle power consumption of an optical switch (IPO)	77 (W)
Energy per bit of an optical switch (EO)	0.2 (W/Gb/s)
Maximum power consumption of a regenerator that can reach 2500km (MPG).	71.4 (W) [81]
Idle power consumption of a regenerator (IPG)	64 (W)
Energy per bit of a regenerator (EG)	0.19 (W/Gb/s)

D. CASE STUDY

In next-generation smart cities, video surveillance plays an important role. For instance, distributed cameras on a stretch of a busy road or a shopping center could bring city security onto a higher level, thus providing the public with a strong sense of assurance [82]. China has already implemented a system called Skynet, which consists of a massive network of smart CCTV cameras that have AI incorporated into them and it is claimed to cover the whole of Beijing [83]. The sheer data rates associated with high-resolution video streams by a large-scale network of intelligent cameras make it virtually impractical to transport all of that data to the cloud for processing. Thus, this motivates us to investigate, through the concept of fog computing to help reduce the implications of unnecessary data exchange with the centralized cloud by hosting parts and/or all the service requests in the distributed layers of the fog framework.

E. POWER CONSUMPTION EVALUATION

We approach the service placement optimization problem using two design approaches: 1) the long-term un-capacitated approach and 2) the short-term capacitated approach. In approach 1), we consider non-splittable service tasks because resources are assumed to be un-capacitated (or unlimited) whereas the opposite is true in approach 2). Thus, here we consider splittable tasks to make better use of the underutilized low-power edge nodes through improved server packing. Note that IoT devices are in all cases capacitated in terms of processing resources but the network resources are always sufficient to carry the traffic. Therefore, the 'capacitated restriction' applies only to the number of processing servers available at the higher fog layers. Most CCTV cameras can be fitted with a PIR motion sensor, hence traffic conditions can be monitored and used to switch these devices between active and idle modes. Thus. for each design approach, we have evaluated the power consumption by considering four scenarios that capture the different distribution of IoT source nodes. Scenario #1 consisted of a single IoT source node generating workloads, Scenario #2 consisted of 5 IoT sources nodes within the same IoT group, Scenario #3 consisted of 4 IoT source nodes, 1 per group and Scenario #4 consisted of the other end of the extremes which has all the IoTs (20 devices) generating workload requests. Table 7 summarizes the source nodes' distribution in the evaluated architecture under each scenario together with the minimum and the maximum workloads. Furthermore, the impact of deploying SP-DCs is also investigated together with the inter-service processing overhead needed for synchronization among sub-tasks due to splitting. Consistent with our previous work [79], we adopt the energy efficient power profile depicted in Figure 2 in which any unused networking and/or processing element is shut down. This can be extended to account for the implications posed by such an approach in future work [84]. Also, the un-capacitated and the capacitated processing devices in this work refer to those nodes that can have their processing resources expanded through a LAN network as opposed to those nodes that are not able to expand such as the low-power IoT devices.

TABLE 7. Distribution of source nodes and workloads in all scenarios.

Scenario	Source Node Distribution	Total # of IoT Sources	Total Workload (MIPS)	
			Min	Max
#1	1 source in a random group	1	1k	10k
#2	5 sources in the same group	5	5k	100k
#3	1 source per group	4	4k	80k
#4	5 sources per group	20	20k	200k

1) UN-CAPACITATED DESIGN WITH NON-SPLITTABLE SERVICE TASKS

In the un-capacitated design approach, it is assumed that the number of processing devices deployed at each node is unrestricted except in those nodes located in the IoT layer due to their limited features. This approach aims to determine for a given workload volume, the optimum resources needed to host a given service if there are no restrictions on the network equipment capacity and no restrictions on the number of servers that can be hosted at each site. The goal is also to determine whether it is the optimal choice to build large numbers of devices at a given location in the proposed architecture. Generally, such design problems occur in medium to long-term network designs [85].

Scenario #1:

In this scenario, out of the total of 20 IoT devices in the model, we take one end of the extremes and assume that only a single IoT device is active at any time instance and the rest of the IoT devices are in idle mode. As expected and shown in Figure 3, for low workload values such as 1000 MIPS, significant savings (up to 98%) can be achieved compared to the baseline solution, where the baseline solution is a scenario where processing is always carried out at the GP-DC. This significant saving is due to the low-power local computational resources available at the IoT layer; hence, the costly overhead of the network and high idle powers of DC servers are avoided. However, as the workload increases, due to resource limitation, the processing capacity available at the IoT layer can no longer host workloads, therefore we begin to see the intervention of the CF nodes as they are only a single hop away. As shown in Figure 4, the general trend in this scenario always favors the activation of additional servers attached to the CF node due to its low idle power consumption compared with the servers located in the upper layers of the fog infrastructure and the cloud DC. Nevertheless, the results still indicate promising power savings of at least 70%.

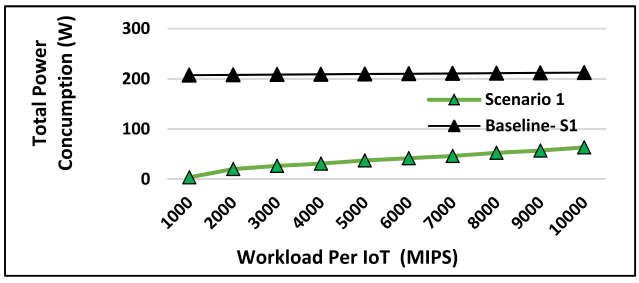


FIGURE 3. Total power consumption of Scenario #1 using the fog approach vs. the baseline, in the un-capacitated case.

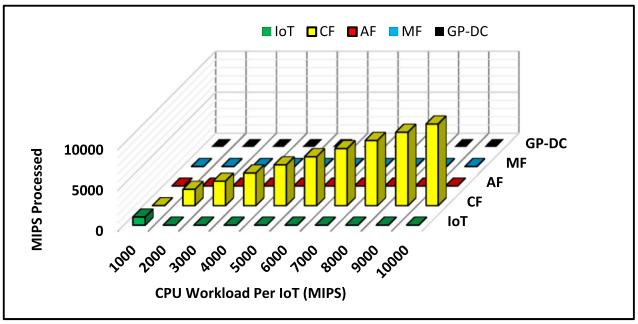


FIGURE 4. Workload distribution of Scenario #1, using fog, in the un-capacitated case.

Scenario #2:

In this scenario, the number of IoT source nodes has increased to five, residing in the same group and connected to the same CPE fog (CF) node. The trends in this scenario remain the same as Scenario #1, except for workload values of 5000 MIPS and beyond. As can be seen in Figure 6, the model decides to allocate all the workloads to the metro fog (MF) that is accessed via the metro network. Although the IoT devices are collocated in the same group and can be allocated to a single CF node, the results indicate that for relatively high workloads ($\geq$5000 MIPS) activating a large MF server with a higher idle power and other associated overheads such as networking and PUE is more optimal than activating many smaller CF servers with less processing efficiency (W/MIPS). Hence, this gives interesting insights about the potential large-scale deployment of such small servers at the edge of the network which may not be as energy efficient as larger fog nodes concentrated higher up in the network hierarchy. For lower ends of the workloads, CF servers produce savings of up to 69%, however, this diminishes as the workload intensity increases and this drops to 37% as can be seen in Figure 5, due to the activation of many CF servers to process the increased workloads. With the workloads allocated to the MF node, power savings of up to 46% can be achieved, compared to the baseline. As can be seen in Figure 6, the model never utilizes the access fog (AF) server despite its close proximity in terms of the number of hops, due to its high associated PUE coupled with low processing efficiency.

Scenario #3:

In this scenario, we aim to investigate the effect the geographical location of the IoT devices has on the optimal

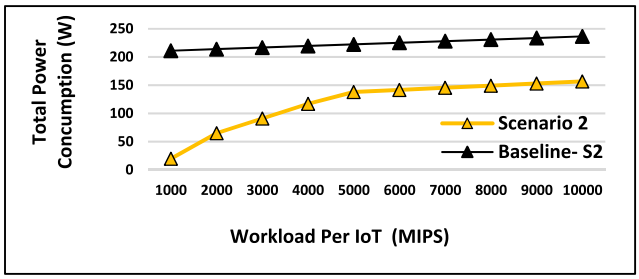


FIGURE 5. Total power consumption of Scenario #2 using the fog approach vs. the baseline, in the un-capacitated case.

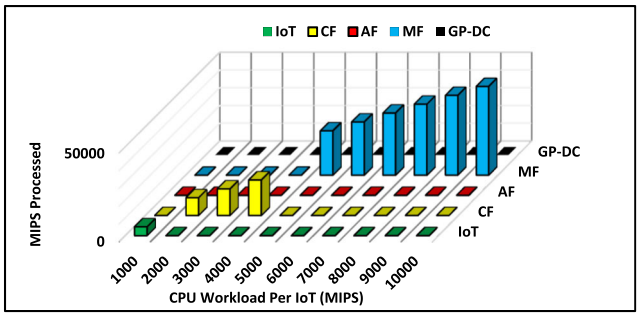


FIGURE 6. Workload distribution of Scenario #2, using fog, in the un-capacitated case.

allocation of the application services, hence, each request is connected to a separate network. Interestingly, the trends remain unchanged. The results here indicate that despite the activation of additional ONU devices due to the distribution of the source nodes in the network, allocating services to CF nodes is still the optimal choice as in Scenario #1. However, as was expected, for processing workloads higher than 5000 MIPS processing moves to the MF as in Scenario #2 due to its higher processing efficiency. The fog approach still produces promising power savings compared to the baseline scenario, as can be seen in Figure 7. When all of the workloads are hosted at the CF layer, power savings of up to 66% can be achieved whilst this drops down to 39% as processing is moved to the MF layer. The difference in the savings is due to the difference between the idle power consumption of the servers in those layers.

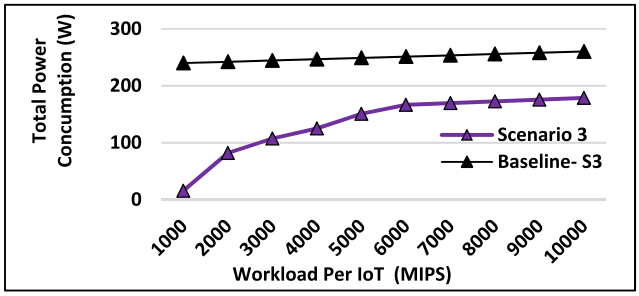


FIGURE 7. Total power consumption of Scenario #3 using the fog approach vs. the baseline, in the un-capacitated case.

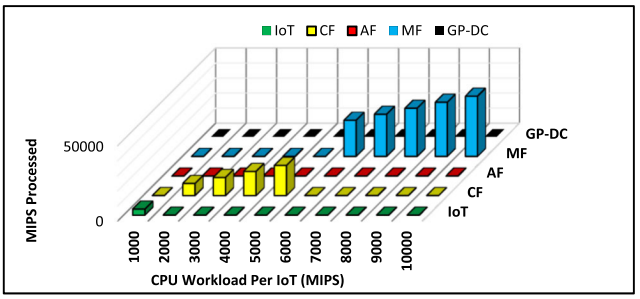


FIGURE 8. Workload distribution of Scenario #3, using fog, in the un-capacitated case.

Scenario #4:

We take the other end of the extremes and assume that all of the IoT nodes are generating requests for resources, simultaneously. With the increase in the number of IoT source nodes, the volume of workload also increases, hence trends are expected to change. As can be seen in Figure 9, the fog approach still yields total savings of up to 17% compared to the baseline. However, when the workload volume reaches a certain level, e.g. at 5000 MIPS, the model decides to allocate all of the workloads to the centralized cloud data center and bypasses the fog layers all together as serving high workload volumes requires activating multiple servers at the metro fog (MF) layer. This justifies networking overheads and higher idle power associated with activating a large server of improved processing efficiency and PUE at the cloud layer. At 6000 MIPS and 7000 MIPS the model switches back to the MF as at those particular workload levels, additional servers have to be activated at the cloud, thus the PUE and processing efficiency of the cloud servers do not justify the idle power consumption of multiple cloud servers in addition to the networking overhead. Generally, the trends indicate that the optimum processing location is dependent on the number of servers required to process the workload. The only time the CF server is utilized is at 4000 MIPS in combination with MF as it saves activating an additional MF server.

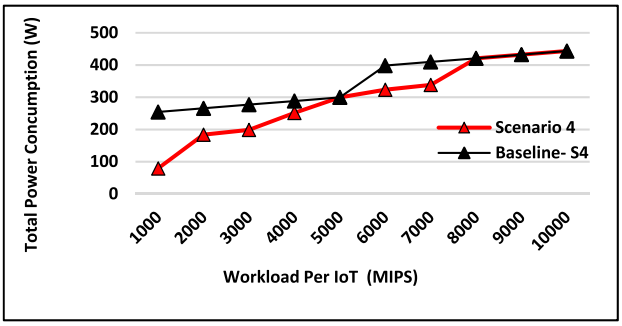


FIGURE 9. Total power consumption in Scenario #4 of fog vs. baseline, in the un-capacitated case.

2) CAPACITATED DESIGN WITH NON-SPLITTABLE SERVICE TASKS

Here, we consider the case where extra capacity cannot be added to the processing nodes in question, hence the

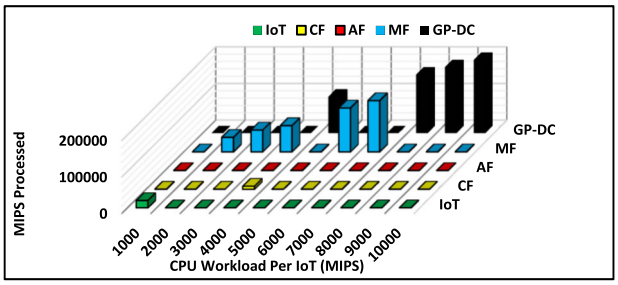


FIGURE 10. Workload distribution in Scenario #4, using fog, in the un-capacitated case.

problem is capacitated. Such design problems are faced in the short-term when the network is already designed, and the processing nodes have been put in place.

Scenario #1:

In the capacitated design problem, different trends are expected because the prospect of adding extra processing capacity is no longer the case. As can be seen in Figure 12, unlike the trends observed in the scenarios of the un-capacitated case, the AF server is chosen as the next best choice after the IoT local computation and CF capacities have run out. We have already observed that the AF server is never a good choice in the un-capacitated case, and this is primarily down to the high idle power of the servers used inside the AF and the associated PUE required for cooling. Although a bad choice in the longer run, the fog approach still yields savings of up to 46% with AF server as the chosen processing destination, as shown in Figure 11.

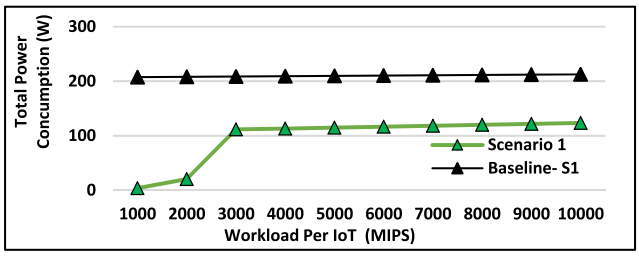


FIGURE 11. Total power consumption in Scenario #1 of fog vs. baseline, in the capacitated case.

Scenario #2:

In this scenario, we begin to observe the disappearance of the AF node as anticipated due to its lower processing efficiency and higher PUE compared with the MF node, as shown in Figure 14. The total power savings here drop down to 41% from 69% for workload volumes of 2000 MIPS in the un-capacitated case. This is mainly the difference between hosting the demands in the CF layer compared to the AF layer. As shown in Figure 13, still a significant amount of power saving is achieved compared to the baseline solution. Although the CF servers had enough capacity to host 9600 MIPS of the total 10,000 MIPS (2000 MIPS/IoT), the model is forced to consolidate processing at the AF layer due to the task splitting constraint forcing processing to take

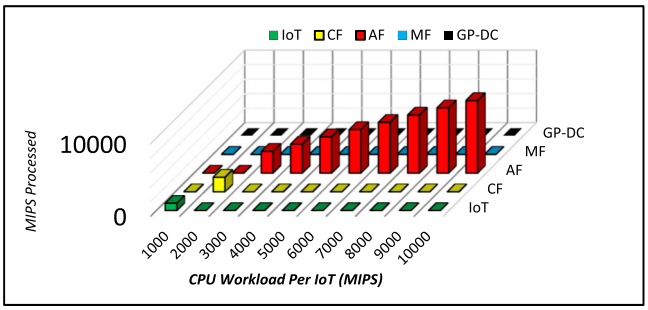


FIGURE 12. Workload distribution in Scenario #1, using fog, in the capacitated case.

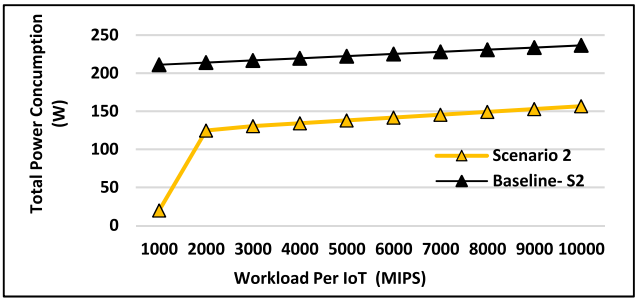


FIGURE 13. Total power consumption in Scenario #2 of fog vs. baseline, in the capacitated case.

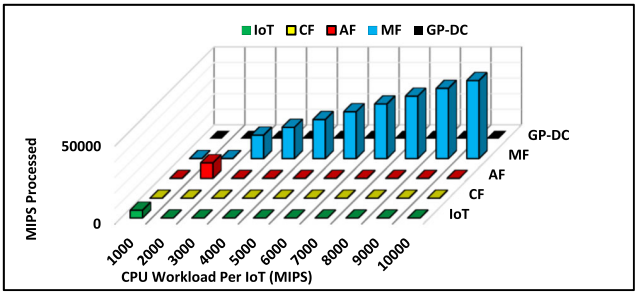


FIGURE 14. Workload distribution in Scenario #2, using fog, in the capacitated case.

place in a single location because the AF server would need to intervene anyway to process at least 400 MIPS thus packing a single AF server is the optimal choice in this case. This is consistent with previous observations in the un-capacitated case, for lower workload volumes (i.e. 2000 MIPS), the model tends to serve the demands in the lower layers of the fog such as the AF node primarily due to the level of workload since the processing efficiency of the MF server and its lower PUE does not justify the networking overhead for accessing the MF. However, as the workload increases (i.e. 3000 MIPS and higher), the processing efficiency coupled with the lower PUE of the MF server compensates for the networking overhead, hence the MF node is chosen as the optimal location to serve the demands as can be seen in Figure 14.

Scenario #3:

As shown in Figure 16 the trends in this scenario are relatively comparable to Scenario #2, except for the case at 2000 MIPS where instead of the AF server, the CF servers are

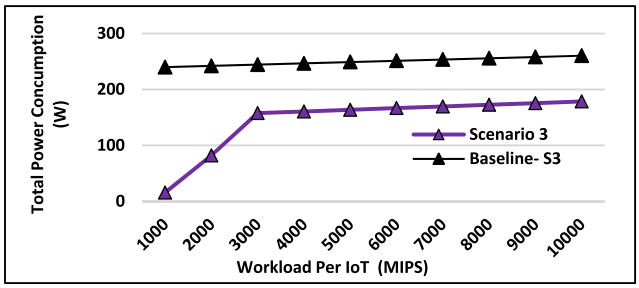


FIGURE 15. Total power consumption in Scenario #3 of fog vs. baseline, in the capacitated case.

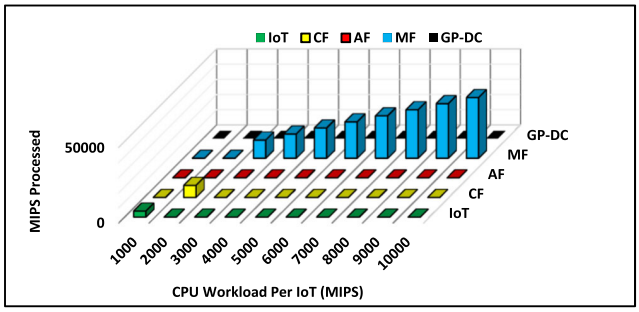


FIGURE 16. Workload distribution in Scenario #3, using fog, in the capacitated case.

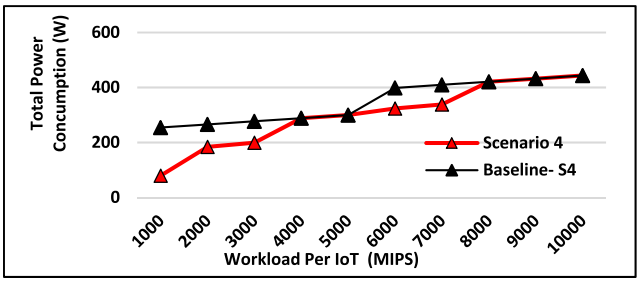


FIGURE 17. Total power consumption in Scenario #4 of fog vs. baseline, in the capacitated case.

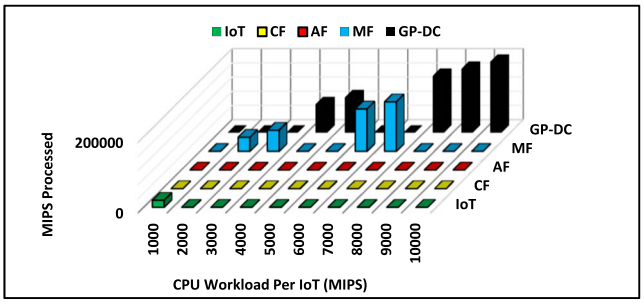


FIGURE 18. Workload distribution in Scenario #4, using fog, in the capacitated case.

utilized. This is mainly due to the geographical distribution of the IoT source nodes as in this scenario, each CF server has enough capacity to serve its source node and the number of source nodes happen to match the number of CF servers available, hence the high idle power and associated PUE of the higher fog layers like the AF and the MF can be avoided in this case, unlike Scenario #2 at 2000 MIPS. A total saving of up to 66% is achieved at 2000 MIPS and up to 55% saving at higher workloads is achieved, as shown in Figure 15.

Scenario #4:

In this scenario, we begin to observe the same trends that were found in Scenario #4 in the un-capacitated case except that the intervention of the cloud occurs earlier in this scenario, which is at 4000 MIPS. This result proves the consistency of the model since the extra capacity needed to host all the demands at 4000 MIPS requires multiple servers at the MF node, thus it becomes more efficient to migrate all services to the GP-DC to better pack the already activated servers as it is much more efficient and has a better PUE value. As can be seen in Figure 17, utilization of the MF servers is only beneficial at certain workload values, otherwise once a certain number of servers are required, the network overhead to get to the GP-DC justifies the activation of the MF server. Figure 18 shows that there are still substantial savings (about 17%) at 7000 MIPS despite the activation of multiple servers at the MF and its high PUE, compared to the GP-DC.

3) CAPACITATED DESIGN WITH SPLITTABLE SERVICE TASKS

Future IoT applications will consist of multiple components, coordinating and communicating over the network to achieve a common task, similar to application design in Service-Oriented Architectures (SOA) [1] smart surveillance cameras. Each IoT device holds a limited amount of computational resources. Given the scale of the IoT network, each device may be called on to provide a variety of services [86]. An example of such a use case is a smart video surveillance application deployed to detect red light violations through several components: namely motion detection, object tracking, data compression, etc. Service distribution in this context would consist of allocating simple tasks such as sensing and actuating in the cameras while the long-term, sophisticated algorithms are expected to be placed on higher-capacity servers such as those in the cloud [90]. In this direction, we evaluate a scenario in which, processing tasks can be split into multiple sub-tasks, hence multiple processing nodes can be utilized to process a single workload [88]. Figure 19 shows an illustrative example of task splitting with an IoT service consisting of four sub-tasks (S1-S4). A single sub-task (S1) is processed locally whilst the second sub-task (S2) is offloaded via the ONU device to another IoT in the same IoT group. Since the total IoT capacity has been fully utilized, the remaining sub-tasks (S3 and S4) are processed on the CPU attached to the ONU node. Note that this is merely an illustrative example, it does not reflect the optimal distribution of the sub-tasks in any way. The main goal is to determine in cases where IoT devices' available CPU capacity is not sufficient to process a given task, whether splitting among multiple processing nodes becomes beneficial in terms of total power reduction, given the added power associated with network overheads after splitting, especially

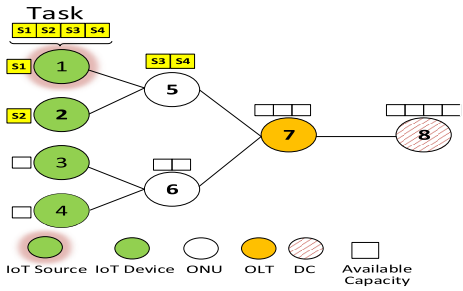


FIGURE 19. An illustrative example of task splitting.

when the network equipment consumes idle power. In our evaluations, task splitting is only said to have occurred if different sub-tasks belonging to the same source node are processed in geographically distributed servers, otherwise, if sub-tasks are all processed in the same processing node, then this is not classed as task splitting mainly due to the same network latency for the sub-tasks.

The MILP model remains unchanged except for a minimal modification to the processing location constraint in (22), to adapt to the variation introduced by task splitting. Previously, it was assumed that the parameter $K = 1$, in the current evaluations, K will adopt values from 1 – 5 to investigate whether task splitting in the short-term capacitated problem introduces additional savings on top of the fog approach. We also consider the same distribution of source nodes as previous sections to be able to make comparisons, where applicable.

Scenario #1:

Figure 20 shows, in the case where processing nodes are limited by capacity, with the increase in the number of service splits (i.e. K = 5), substantial savings can be made as opposed to the case with no service splits (K = 1). The savings are because the high capacity servers' idle power is avoided since application services can be processed locally between the IoT and the CF layers, despite the network overhead incurred in getting access to these devices.

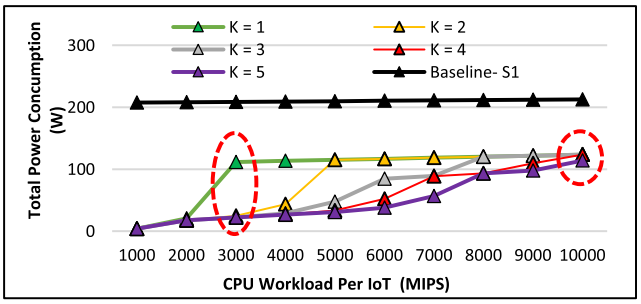


FIGURE 20. Total power consumption in Scenario #1 using fog, for a range of values of K, in the capacitated case.

The total savings achieved by the fog approach with non-splittable service tasks (K = 1) was up to 46% compared to the baseline, however, this figure increased to 88% when the value of K was changed to 2 (i.e. K = 2), as highlighted in Figure 20. However, when the workload volume increases to 10,000 MIPS, we begin to see a drastic dip in the power savings introduced by task splitting (i.e. K = 5). The power savings dropped from 88% to about 5% as highlighted in Figure 20. This is due to the number of activated CF servers that are needed to process the large workload and the added power due to networking overheads to access the CF servers that are located in different parts of the architecture shown in Figure 1. An important observation to be made here is that the model is forced to pack the CF servers to accommodate the extra workload that cannot be processed by IoT devices. This is mainly due to the splitting value (K = 5), for instance at 6000 MIPS, to keep processing at the IoT layer, this workload has to be split into 6 segments (i.e. K = 6) since each IoT device can only process at maximum 1000 MIPS. To comply with the splitting constraint, 4000 MIPS is processed by four IoT devices and the remaining 2000 MIPS is allocated to a CF server, hence the total number of splits is five as shown in Figure 20. From these results, we can, therefore, conclude that for a fog infrastructure owner offering IoT services such as the one presented here, the choice of task splitting among low-power IoT devices will introduce substantial power savings, but may not introduce further savings if these low-power devices are located in different parts of the network. Thus, the power savings obtained through relaxing the task splitting constraint (i.e. $K = |P|$) can guide IoT service providers in terms of cost reduction offered to the IoT service consumers.

Scenario #2:

In this scenario, the power savings introduced by task splitting is very limited, especially for workloads of 4000 MIPS

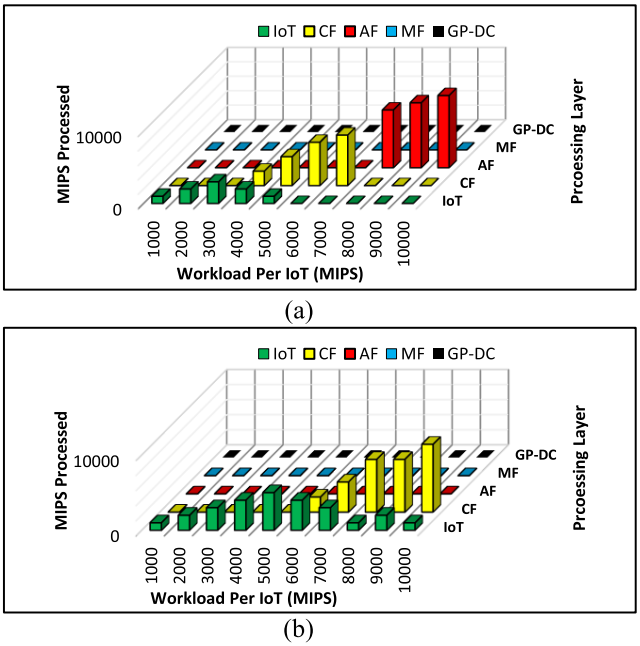


FIGURE 21. Workload distribution in Scenario #1 during (a) K = 3 and (b) K = 5 service splits, in the capacitate case.

and beyond, as shown in Figure 22. This is largely down to the processing capacity limitations placed on the CF servers and IoT devices. As in this particular distribution, all of the IoT nodes inside a single group have requests themselves, hence the idle processing resources of low-power nodes have to be accessed via the network depicted in Figure 1. In the lower end of the workloads (2000 MIPS – 3000 MIPS), the model utilizes task splitting and chooses to allocate the total workload to the low-power IoT nodes located in different parts of the network. Interestingly, the CF servers are avoided despite the ONU devices being activated anyway to gain access to the other parts of the network and the relatively lower idle power of the CF servers. This can be related mainly due to the superior processing efficiency of the MF servers compared to that of the CF servers, hence this gain compensates for the high idle power of the MF server coupled with the cost of the network and the external overhead of PUE. As the workload increases i.e. to 4000 MIPS, this means that the total workload of 20,000 MIPS can no longer be processed by the same number of IoT nodes in the previous workload. Therefore, another IoT group would need to be activated which increases the networking overhead, especially ONU devices' idle power consumption. Thus, any value of K>1 becomes irrelevant and trading off the idle power of a single MF server coupled with the PUE overhead saves more energy than utilizing the task splitting approach.

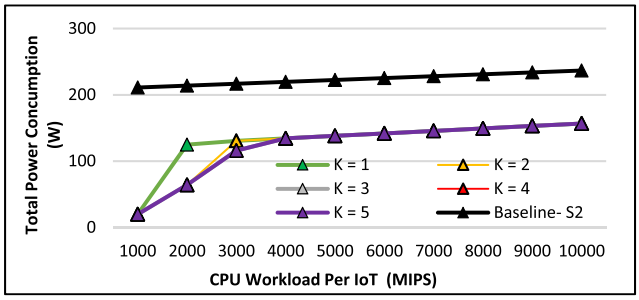


FIGURE 22. Total power consumption in Scenario #2 using fog, for a range of values of K, in the capacitated case.

Scenario #3:

The trends in this scenario remain relatively unchanged compared to Scenario #2 except for the fact that task splitting is utilized only due to the distribution of the IoT source nodes as they are based in separated groups, and the ONU devices would need to be turned ON anyway to get to the higher layers, hence the CF server attached to the ONUs are used due to their low idle power compared to that of the MF servers. This observation was established in previous scenarios of all the cases in Figure 23, at 4000 MIPS, where only the MF server was used, compared to 4000 MIPS in this scenario where the workload is processed between the IoT and CF nodes. In this scenario, a total saving of 56% was achieved with task splitting (K >3) as opposed to 33% with non-splittable tasks (K = 1), as highlighted in Figure 24.

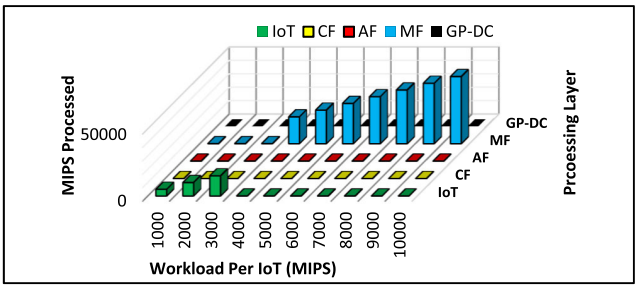


FIGURE 23. Workload distribution in Scenario #2 during K = 3 and K = 5.

As mentioned previously, this large saving is the difference between the idle power of the MF server and the smaller devices in the IoT and CF layers. However, we have already established that, when the workload is increased, the processing per instruction at the MF compensates for the idle power of its server, hence all workloads are processed at the MF layer as can be seen in Figure 25(a) and Figure 25(b) at 7000 MIPS.

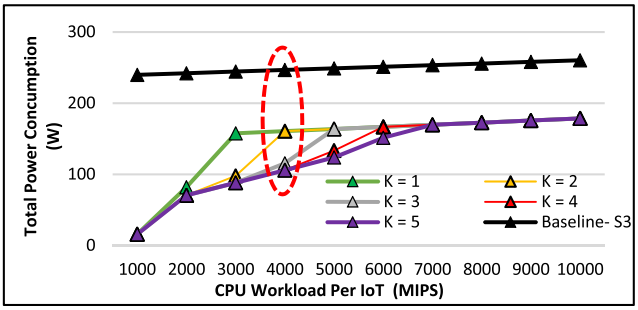
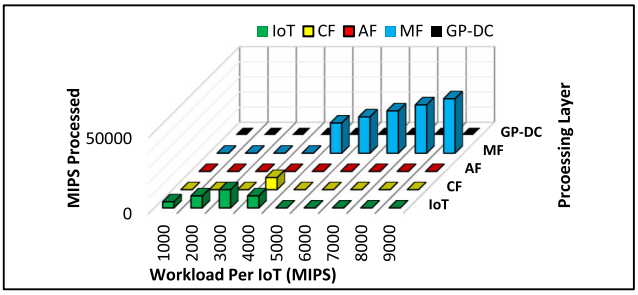
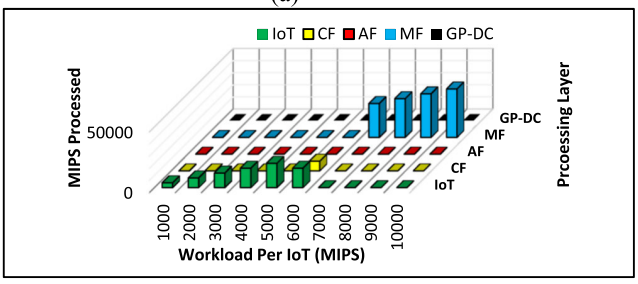


FIGURE 24. Total power consumption in Scenario #3 using fog, for a range of values of K, in the capacitated case.



(a)



(b)

FIGURE 25. Workload distribution in Scenario #3 during (a) K = 3 and (b) K = 5 service splits, in the capacitate case.

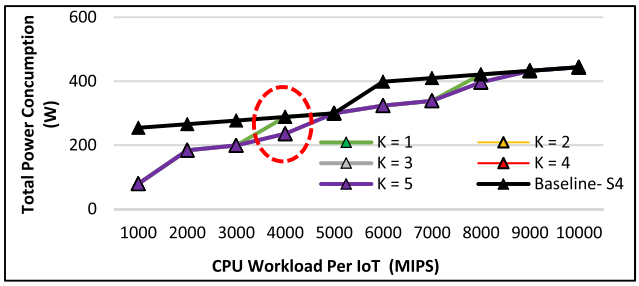


FIGURE 26. Total power consumption in Scenario #4 using fog, for a range of values of K, in the capacitated case.

Scenario #4:

For large workloads, task splitting is predominantly irrelevant, except in rare circumstances such as the case at 4000 MIPS at K>1, a total of 80,000 MIPS is demanded by the source nodes and if all of this was to be processed at the MF layer, it would require two servers, hence, in this case, the MF server is fully packed and the remaining workload (6560 MIPS) is processed on the source nodes' local CPUs. Thus, as shown in Figure 25, task splitting at K>1 introduces total savings of up to 18% compared to 0% with no task splitting (K = 1) as the solution was the same as the baseline in this instance. Similar to the observations obtained in the un-capacitated case, as shown in both cases K = 3 and K = 5 in Figure 27, the MF and the cloud are largely the best choices, respectively, when the workload is too high and the AF is avoided despite its close proximity to the source nodes. This is mainly due to the energy efficiency of the servers available to these processing layers and the relatively improved PUE compared to the AF.

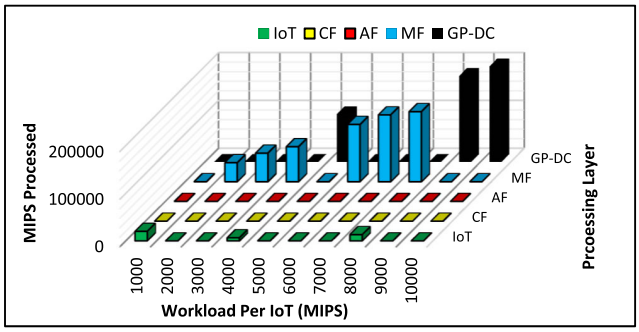


FIGURE 27. Workload distribution in Scenario #4 during both K = 3 and K = 5 service splits, in the capacitate case.

4) IMPACT OF SP-DCS GIVEN NON-SPLITTABLE SERVICE TASKS

Given the high energy efficiency of SP-DC servers, it is worth investigating its impact on improving the energy efficiency of the proposed fog model and whether the distributed processing approach is still producing savings in the presence of such highly energy-efficient servers at the centralized cloud, in both the capacitated and un-capacitated design problems without the prospect of enhanced server packing through task splitting. The results indicated that all the trends in Scenario #1, Scenario #2 and Scenario #3, remained unchanged. However, as expected, the trends changed in Scenario #4 due to the high level of workloads and the need for the cloud to intervene even before considering an SP-DCs, if anything, the deployment of SP-DC should incentivize centralized processing more than ever. Thus, the impact of the SP-DC is observed at and beyond 4000 MIPS, as shown in Figure 29. Interestingly, as shown in Figure 28, the SP-DC yields total savings of up to 50% compared to processing in GP-DC (baseline), whilst the maximum saving obtained in the optimized scenario with GP-DCs shown in Figure 9 was up to 30%, in both capacitated and un-capacitated design problems. These results demonstrate that hosting services of high computational workload on mini DCs in the fog layers that are associated with high PUEs and are less efficient in terms of processing per instruction brings no benefits when a highly efficient centralized DC is available at the core network.

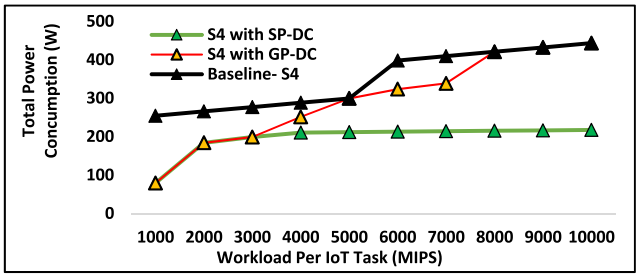


FIGURE 28. Total power consumption in Scenario #4 with/ without SP-DC.

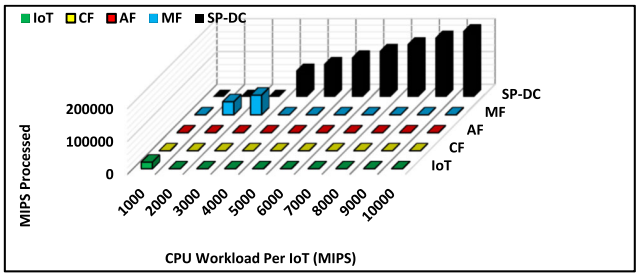


FIGURE 29. Workload distribution using fog, in Scenario #4 with SP-DC, in all cases.

VI. INTER-SERVICE PROCESSING OVERHEAD DUE TO SYNCHRONIZATION TRAFFIC

In the previous section, we evaluated the performance of the distributed processing approach for the capacitated problem given splittable services without considering any splitting overhead. In this section, we consider a case in which service splits incur extra processing overhead, i.e. the processing overhead is needed for synchronization purposes between the sub-tasks of a given application to complete a processing task. Note that we only consider the processing overhead since

the communication traffic proportional power consumption is almost negligible in terms of its influence on decision making as network equipment's' idle power is 60% - 90% of the maximum power. For the processing overhead, we have assumed a fraction of the maximum power consumption at a serving node is added on top to form the total power consumption. As shown in the illustrative example in Figure 30, a given service is split unto three serving nodes (S1- S3), hence synchronization links annotated by green dashed lines are established among the serving nodes. Therefore, each serving node for the service in question consumes an extra processing overhead

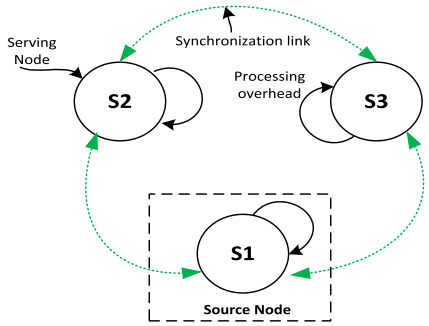


FIGURE 30. Illustrative example of splitting overhead.

Before introducing the MILP model, the additional parameters and variables are defined as follows:

Application Parameters:

ψ_d PUE of processing device $d \in P$.

Variables:

$\lambda^s_{d1,d2}$ $\lambda^s_{d1d2} = 1$, if there is synchronization traffic of service $s \in \mathbb{S}$ between processing node $d1 \in \mathbb{P}$ and $d2 \in \mathbb{P} : d2 \neq d1$, otherwise $\lambda^s_{d1,d2} = 0$.

$\lambda_{d1,d2}$ Synchronization traffic between processing nodes $d1 \in \mathbb{P}$ and $d2 \in \mathbb{P} : d2 \neq d1$.

$\lambda^{d1,d2}_{m,n}$ Synchronization traffic between processing node $d1 \in \mathbb{P}$ and $d2 \in \mathbb{P} : d2 \neq d1$, traversing link m, n, where $m \in \mathbb{N}$ and $n \in \mathbb{N}_m$.

$\lambda^{(sync)}_i$ Synchronization traffic on node $i \in \mathbb{N}$.

$\rho^s_{d1,d2}$ Processing overhead resulting from splitting the service request of source node $s \in \mathbb{S}$, between processing nodes $d1 \in \mathbb{P}$ and $d2 \in P : d2 \neq d1$.

The total power consumption equations remain intact except for an additional equation which accounts for synchronization processing overhead and this is defined as follows:

Processing Power Consumption Overhead Due to Synchronization:

$$\sum_{s\in\mathbb{S}} \sum_{d1\in\mathbb{P}} \sum_{\substack{d2\in\mathbb{P}:\\ d2\neq d1}} \rho^s_{d1,d2}.E^{(i)}_{d2}.\psi_{d2} \tag{41}$$

Additional Constraint:

$$\sum_{n\in\mathbb{N}_m} \lambda^{d1,d2}_{m,n} - \sum_{n\in\mathbb{N}_m} \lambda^{d1,d2}_{n,m} = \begin{cases} \lambda_{d1,d2}, & m = d1 \\ -\lambda_{d1,d2}, & m = d2 \\ 0, & otherwise \end{cases}$$
$$\forall d1 \in \mathbb{P}, d2 \in \mathbb{P}, m \in \mathbb{N} : d1 \neq d2. \tag{42}$$

Constraint (42) conserves synchronization traffic from the source node to the destination node. It ensures that the total incoming traffic at a node is equal to the total outgoing traffic of that node; unless the node in question is either the source node or the destination node.

$$\lambda^s_{d1,d2} \leq \Omega^{s,d1} \quad \forall s \in \mathbb{S}, d1 \in \mathbb{P}, d2 \in \mathbb{P} : d1 \neq d2 \tag{43}$$
$$\lambda^s_{d1,d2} \leq \Omega^{s,d2} \forall s \in \mathbb{S}, d1 \in P, d2 \in \mathbb{P} : d1 \neq d2 \tag{44}$$
$$\lambda^s_{d1,d2} \geq \left(\Omega^{s,d1} + \Omega^{s,d2}\right) - 1$$
$$\forall s \in \mathbb{S}, d1 \in \mathbb{P}, d2 \in \mathbb{P} : d1 \neq d2 \tag{45}$$

Constraints (43) to (45) are used in the linearization of the product of binary variables $\Omega^{s,d1}$ and $\Omega^{s,d2}$, where $s \in \mathbb{S}$ $d1 \in \mathbb{P}$ and $d2 \in \mathbb{P} : d2 \neq d1$.

$$\rho^s_{d1,d2} = \lambda^s_{d1,d2}.CPU_s.\phi$$
$$\forall s \in \mathbb{S}, d1 \in \mathbb{P}, d2 \in \mathbb{P} : d1 \neq d2 \tag{46}$$

Constraint (46) ensures that the total synchronization processing overhead resulting from splitting the service request of source node $s \in \mathbb{S}$, between processing nodes $d1 \in \mathbb{P}$ and $d2 \in \mathbb{P} : d2 \neq d1$ is realized.

$$\lambda^{(sync)}_i = \sum_{d1\in\mathbb{P}} \sum_{\substack{d2\in\mathbb{P}:\\ d2\neq d1}} \lambda^{d1,d2}_{m,n} + \sum_{d1\in\mathbb{P}} \sum_{\substack{n\in\mathbb{N}_m:\\ d1\neq m, m\in\mathbb{P}}} \lambda^{d1,m}_{n,m}$$
$$\forall m \in \mathbb{N} \tag{47}$$

Constraint (47) ensures that egress and ingress synchronization traffic on node $i \in N$ is accounted for.

$$\lambda_{d1,d2} = \sum_{s\in\mathbb{S}} \left(\lambda^s_{d1,d2}.BW_s\right)$$
$$\forall d1 \in \mathbb{P}, d2 \in \mathbb{P} : d1 \neq d2 \tag{48}$$

Constraint (48) ensures that, the total communication demand between $d1 \in \mathbb{P}$ and $d2 \in \mathbb{P}$, where $d2 \neq d1$ is achieved.

A. POWER CONSUMPTION EVALUATION

The power consumption evaluations within this section are based on the assumption that each sub-task requires 1% or 10% of the total processing workload allocated to a serving node. Note that the number of task splits has not been constrained. The evaluation considers Scenario #1, Scenario #2 and Scenario #3 in the capacitated case, since the results in the previous section showed the largest number of splits occurred in these scenarios. Therefore, it is of interest to investigate the extent to which the processing overhead impacts the decision in terms of consolidating workloads in fewer processing nodes.

Scenario #1:

In previous evaluations, the task splitting case in the capacitated setting produced substantial savings during low workloads such as Scenario #1. After having considered the processing overhead due to synchronization, the results in Figure 32(a), indicate that for a small overhead such as 1%, task splitting is predominantly favorable and the only time services are consolidated in fewer servers is when the workload is very high i.e. ≥ 9000 MIPS. In such cases, although the model could have made use of splitting, instead it decided to consolidate the services in the AF server with a much higher idle power, associated PUE and networking overhead in return for much better processing efficiency compared to IoTs and CF nodes. Moreover, when processing overhead is increased to 10%, even for very small workloads such as 2000 MIPS, the placement decision is already impacted and the services are consolidated on the CF nodes, as can be seen in Figure 32(b). Also in Figure 32(a), at 5000 MIPS, the original solution with no overhead (S1| No_OH) decided to split the total workload among the same IoT group because there was enough capacity offered by the IoT devices, which resulted in 5 task splits. The current solution has done the same number of splits but due to the extra processing overhead imposed by synchronization, the aggregate capacity of the IoTs in the same group has been fully utilized, hence the CF node is activated as it saves more power than activating the higher layer fog nodes. The general trend shows that even at very small overhead (e.g. 1%), task splitting, in the long run, is not an efficient choice as can be seen in Figure 32(a) with the increase in workload, the services are placed higher and higher up the network hierarchy so that service requests can be consolidated on fewer servers. As shown in Figure 31, the total power consumption is increased by up to 42%, considering 10% overhead compared to the optimization scenario with no overhead (No_OH). In contrast, for a very high workload such as 10,000 MIPS, the power overhead was up to 5%. This is because, in the original solution (No_OH), all of the CF nodes across the network topology in Figure 1 were utilized, which makes the total power consumption comparable to that of the current solutions with overheads due to the network.

Scenario #2:

In this scenario, the number of IoT source nodes has increased, hence the total workload has also increased. As can be seen in Figure 34(a), for workloads of 2000 MIPS, the number of service splits have dropped from 10 with no processing overhead to 9, considering the 1% of processing overhead. This confirms the previous observations in Scenario #1 that despite some overhead, for very low demands, task splitting does still introduce power savings, which is up to 67% considering 1% overhead compared to processing in the cloud, whilst this power saving drops to 42% when the overhead is increased to 10% as can be seen in Figure 33. This is mainly due to the difference between local processing on IoT devices and CF servers that are based in different parts of the network. Consistent with previous

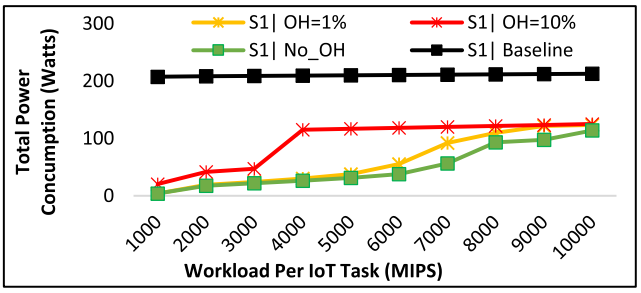
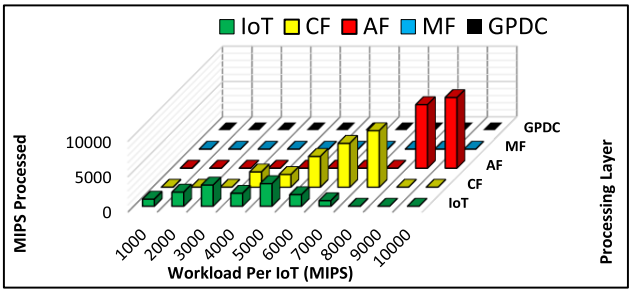
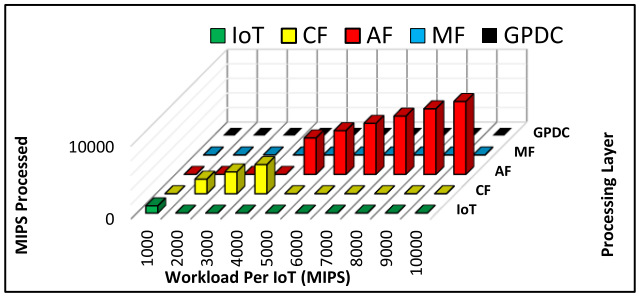


FIGURE 31. Power consumption of fog with/out overheads compared with the baseline solution in Scenario #1.



(a)



(b)

FIGURE 32. Workload distribution in Scenario #1 during (a) 1% processing overhead, (b) 10% processing overhead.

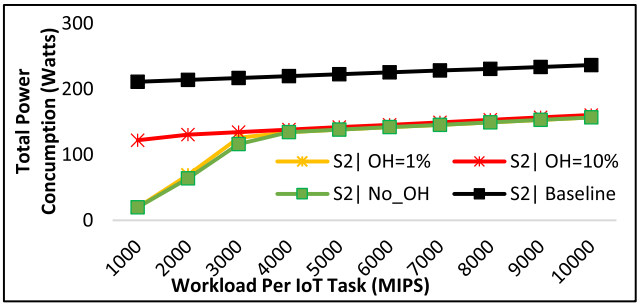


FIGURE 33. Power consumption of fog with/out overheads compared with the baseline solution in Scenario #2.

observations, the MF becomes the dominant choice due to its processing efficiency as this was the case before the synchronization overhead, if anything, synchronization overhead will provide even further incentives to utilize the MF servers.

Scenario #3:

In this scenario, due to the distribution of the IoT source nodes, for low workload volumes and an overhead of 1%,

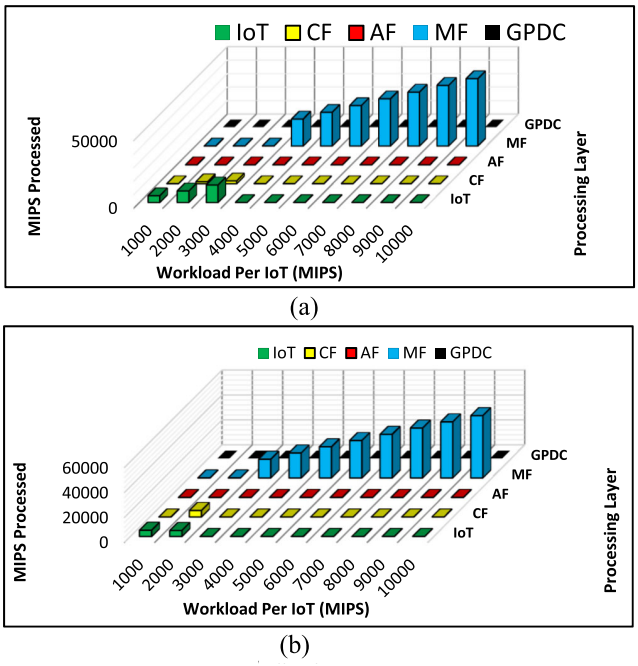


FIGURE 34. Workload distribution in Scenario #2 during (a) 1% CPU overhead, (b) 10% processing overhead.

task splitting is still favorable although the number of splits has decreased compared to the case with no synchronization overheads due to the intervention of the CF servers, as can be observed in Figure 36(a). In order for an IoT source node to access the resources of another IoT device to process its request, an ONU device must be activated, hence utilizing the CF servers with larger capacity would be a better packing option as it will drop the number of service splits, consequently consolidating the workloads in fewer servers to avoid the extra power cost due to the overheads. It has already been established that, when source nodes have low workloads and there are enough idle processing resources on the IoT devices, task splitting can always produce significant savings as shown in Figure 35 at 2000 MIPS regardless of the synchronization overhead. However, when the overhead is low (i.e. 1%) and there are enough resources available on the lower fog layers (IoT and CF), the model chooses to always perform task splitting, regardless of the networking overhead incurred to access the lower fog devices in different parts of the network. Moreover, when the processing overhead is increased to 10%, task splitting is no longer the favorable choice, predominantly as can be seen in Figure 36(b). The total power savings drop from up to 63% with no processing overhead to up to 28% with 10% overhead as can be seen in Figure 35.

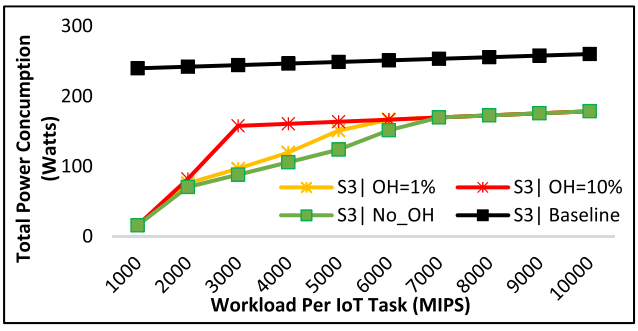


FIGURE 35. Power consumption of fog with/out overheads compared with the baseline solution in Scenario #3.

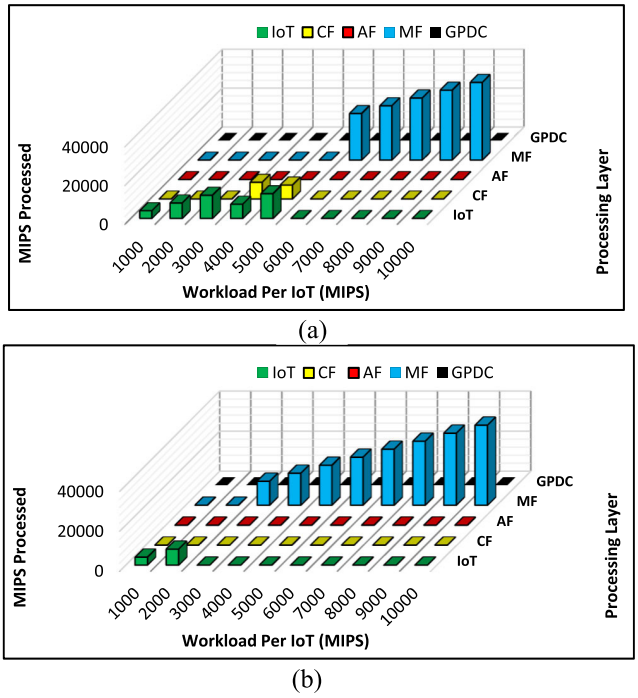


FIGURE 36. Workload distribution in Scenario #3 during (a) 1% processing overhead, (b) 10% processing overhead.

VII. ENERGY EFFICIENT DISTRIBUTED PROCESSING (EEDP) HEURISTIC

The service placement problem over the distributed processing architecture depicted in Figure 1 is computationally costly to solve as it is a non-deterministic polynomial (NP)-hard problem. Generally, due to the NP-hard nature of placement problems, it is impossible to devise heuristic algorithms that can find the optimal solution within polynomial time [89]. The simplest case of our models, i.e. the service placement with non-splittable tasks with $P = 29$ placement locations (20 IoTs, 4 CFs, 2 AFs, 1 MF, 2 DCs) would require evaluating a number of possible solutions that can be in the order of $\sum_{\rho=1}^{P} \frac{P!}{P-\rho}$, where ρ is the number of processing servers. Evidently, this kind of exhaustive method is something that must be totally avoided as it is not feasible to implement such models in real-time for large scale networks. Therefore, heuristic methods can be developed to mitigate this issue as they can provide simple and fast operations in real-time with performances that may approach that of the optimal MILP solution. In this direction, the optimal results of the

MILP model for smaller networks can serve as a benchmarking tool to measure the performance of the developed heuristic.

We have tailored the operation of the EEDP heuristic algorithm towards approximating the results of the MILP model for the case of splittable tasks in a capacitated network, in Section V(3). This is because this scenario was the most energy efficient among the different cases investigated. The flowchart of the EEDP heuristic is shown in Figure 37. It is important to note that all the assumptions of the model under consideration are upheld. Note that we have relaxed the splitting constraint in (22) for both the MILP model and the EEDP heuristic by setting the value of K to $|P|$. Based on the insights from the results of Section V(3), due to the high idle power of processing servers, the model tends to consolidate the placement in larger processing nodes when the capacity of the IoT and CF layer is violated during high workloads. Thus, the heuristic starts by summing the CPU demands to ensure that servers with high idle power are efficiently packed. The algorithm then checks whether or not the total demand (D) can be hosted by the IoT source (SRC) nodes themselves, if the answer to this is "yes", then the total power consumption resulting from processing D locally at source nodes will be calculated. This is because if local computation suffices, there is no need to search the set of all IoTs and hence this reduces the complexity of the heuristic. Then, the sum of the demands in D is allocated to the remaining processing layers sequentially and the total power consumption (network and processing) is calculated. The network power consumption is calculated by routing the traffic via the minimum hop routing algorithm [56] to the relevant processing node(s). The EEDP algorithm then compares the total power consumption of all the processing layers and terminates by choosing the allocation that results in the minimum power consumption for processing the sum of the CPU demands in D. It is worthy of mentioning that we have only considered the idle power of the networking equipment as this accounts for up to 90% of their maximum power as has been shown in Table 2, however, the algorithm can easily be updated to account for the proportional power consumption of networking equipment should that be necessary.

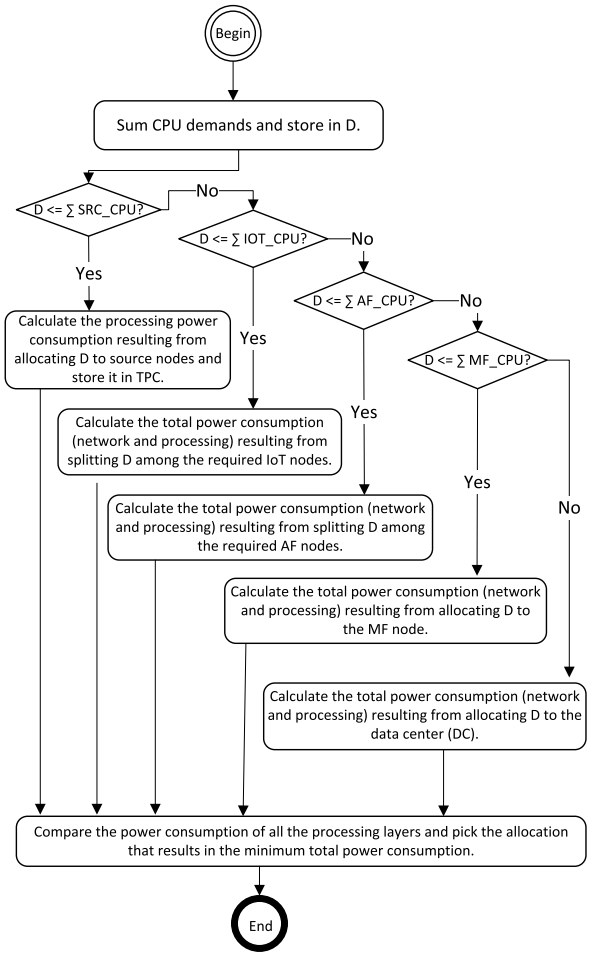


FIGURE 37. Heuristic flowchart.

Figure 38 shows the performance of the EEDP heuristic compared with that of the MILP models in terms of the average power savings across all the evaluated workloads (1000 MIPS – 10,000 MIPS) in all four scenarios. Note that due to the complexity of the MILP model, we limited the value of K to 5 in all of our evaluations in this work, however, we have removed this constraint in this section for both the EEDP heuristic and the MILP model by setting the value of K to the number of available processing nodes across all layers (i.e. $K = |P|$), as complexity is not a limitation anymore. As a result, in S #1 there are further savings of 6% when comparing $K = |P|$ with $K = 5$; and the MILP and the EEDP heuristic are in agreement. The saving is because there are enough IoT resources within the same network and hence the heuristic can split the total demand D among as many low-power IoT nodes as possible. It is important to note that splitting among many IoT nodes will not produce further savings if the IoT nodes are located in different parts of the network as shown by the results of the second MILP that has $K = |P|$.

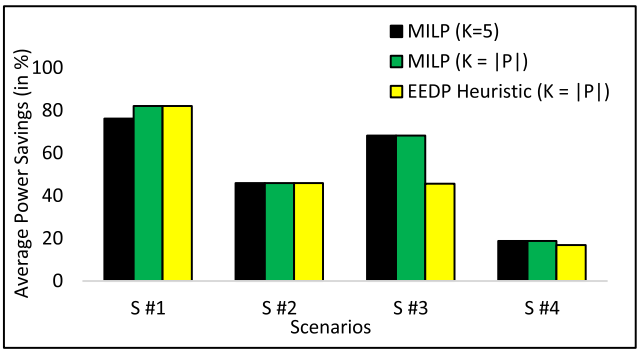


FIGURE 38. Average power consumption of the MILP model in Section V(3) compared with that of the EEDP heuristic.

In Scenario #2 (S #2), the heuristic produces comparable savings to that of the second MILP model. Note here, the prospect of unlimited service splits does not produce

further savings unlike S #1 because the aggregate capacity of the low-power IoT nodes within the same network is no longer able to process the total demand in D, hence activating servers with higher capacity and idle power is more energy efficient than splitting the demand among multiple low-power IoT nodes that are located in different parts of the network.

The heuristic performs poorly in Scenario #3 (S #3) due to the distribution of the source nodes. The gap in the average savings compared to the optimal solution (both MILP models) is approximately 23%. This is because the heuristic is not able to differentiate between the CF servers in terms of the number of network hops between the source nodes and the connected CF nodes. In a distribution such as S #3, the optimal choice is to allocate each demand to the CF server with the minimum number of network hops as is the case in the MILP model, however since we are dealing with the sum of the demands (D), this is not possible in the heuristic. Also, due to the exhaustive nature of the MILP model, different processing combinations that involve multiple processing layers are considered, however, this increases the complexity of the heuristic drastically and therefore we do not consider this aspect, hence there will be a gap in savings compared to the MILP model. As shown in Figure 38, S #4, the MILP models produce better savings as they can offload the extra workload to the low-power IoT nodes to prevent the activation of an additional server at the MS layer, this is observed in Figure 27 during workloads of 4000 MIPS and 8000 MIPS.

The EEDP heuristic is examined by considering the same scenarios as the MILP model in Section V(3) as well as the network topology shown in Figure 1. The EEDP heuristic took 0.038 seconds to evaluate the worst-case scenario which is S #4 using a computer running on an Intel i-7 CPU with 16 GB of RAM. The MILP took 20 minutes to evaluate the same scenario.

VIII. CONCLUSION AND FUTURE WORKS

This article has investigated the energy efficiency of distributed processing based on the concept of fog computing for resource-intensive visual-based applications in the context of IoT. A practical PON access network has been used to aggregate the IoT traffic towards the cloud DC, via the metro and core networking layers. We developed several MILP models to help us produce a comparative study of the performance of the fog approach in different circumstances, which call for different solutions. The service placement problem was investigated, casting the problem into two main areas of network designs; namely, the capacitated problem that is mainly concerned with solutions applicable to a short-term basis as opposed to the un-capacitated counterpart that applies to the longer-term basis. In the latter case where resources are un-capacitated, we studied the performance of the fog approach, given that IoT service tasks are non-splittable. For low workloads, results showed significant power savings of up to 90% compared to the baseline solution, which is due to the difference between local computation on low-power IoT nodes and high-end servers inside cloud DCs. However, during relatively higher workloads, the power savings dropped to 30% due to the activation of higher capacity fog nodes in the metro layer and/or high-end servers inside cloud DCs. For the capacitated problem, the limited processing resources available at the IoT and CF layers motivated us to study the prospect of improved server packing through the concept of task splitting. Results showed, for the maximum number of splits, substantial power savings of up to 88% was achieved compared to 46% with non-splittable services. However, during high workload volumes, task splitting introduced no savings, and services were consolidated in fewer servers located at the MF and cloud layers.

We further extended our studies to include the impact of the processing overhead that is needed for the inter-service communication among sub-tasks of a given application service. The results showed that the extra processing overhead between IoT sub-tasks has a great influence on the total number of task splits. The impact was evident in Scenario #1, wherein the case with no overheads, the maximum number of splits occurred. Whereas with the lowest overhead which was 1%, the maximum number of splits reduced from 3 to 2 for the lowest ends of the workloads and for the higher workloads this reduced from 5 to 0 as consolidating processing in the AF node was more energy efficient than splitting the services among the lower layer devices such as the IoTs and CFs. The results demonstrated that deploying SP-DCs in the cloud did not perform better than the fog except in Scenario #4 where the workload levels and the number of requests were high. It was observed that SP-DCs yielded total savings of up to 50% compared to processing in GP-DC, whilst the maximum power savings obtained was up to 30% using the fog approach. This is a promising performance from the SP-DC and these results demonstrate that for scenarios where the workload is extremely high, deploying mini DCs in the fog layers that are associated with high PUEs and are less efficient in terms of processing per instruction, hosting services on them brings no benefits when a highly efficient centralized DC is available at the core network. As for the un-capacitated problem, it was found that generally for high workloads, building too many small fog nodes (CFs) was not a good option in the long run, due to their limited resources and processing efficiency (W/MIPS), hence, allocating services to the metro fog and /or the cloud DC saved more power due to their processing efficiency. We also built a simple heuristic with a significant reduction in complexity to mimic the behavior of the MILP model in Section V(3).

Motivated by these promising results, a number of research directions for future work include developing optimization models that account for latency metrics explicitly in the objective function or the constraints since latency plays an important role in mission-critical applications such as real-time surveillance. Further areas of future work include differentiating between CPU efficiencies by linking the number of instructions required by a low-power node as opposed to a powerful node due to differences in their CPU architectures. Future work can also consider uncertainties in the

TABLE 8. Analytic verification of the optimal choice in Scenario #4 at 5000 MIPS.

Checkpoint 1							
Potential Solution 1 *Capacitated problem, Scenario #4, all processing done at MF with 5000 MIPS per IoT source node and 2 idle servers.*							
Layer	**IoT**	**CF**	**AF**	**MF**	**Core**	**DC**	**Total Power (W)**
Network Power (W)	6.8	36	3.6	17.8	-	-	**64.2**
Processing Power (W)	-	-	-	$8 + (2 \times 79.8) + 72.4 = \mathbf{240}$	-	-	**240**
Total Power (W)	**304.2**						
Optimal Solution *Capacitated problem, Scenario #4, all processing done at DC with 5000 MIPS per IoT source node and 1 idle servers.*							
Network Power	-	-	-	2.2	$11.6 + 5.1 + 3.4 + 51 = \mathbf{71.1}$	-	$64.2 + 71.1 = \mathbf{135.3}$
Processing Power	-	-	-	$15.1 + 87.4 + 53.9$	-	-	**156.4**
Total Power	**291.7**						
MILP Results	≅ **299** ➔ see Figure 17 and workload 5000 MIPS.						

TABLE 9. Analytic verification of the optimal choice in Scenario #2 at 2000 MIPS.

Checkpoint 2							
Potential Solution *Capacitated problem, Scenario #2, all processing done at AF layer with 2000 MIPS per IoT source node and 1 idle servers.*							
Layer	**IoT**	**CF**	**AF**	**MF**	**Core**	**DC**	**Total Power (W)**
Network Power (W)	-	-	-	17.8	-	-	$13.4 + 17.8 = \mathbf{31.2}$
Processing Power (W)	-	-	-	8 + 80 + 7.2	-	-	**95.2**
Total Power (W)	**126.4**						
Optimal Solution *Capacitated problem, Scenario #2, all processing done at AF layer with 2000 MIPS per IoT source node and 1 idle servers.*							
Network Power	1.7	9	2.7	-	-	-	**13.4**
Processing Power	-	-	-	8.6 + 85.5 + 15.5	-	-	**109.6**
Total Power	**123**						
MILP Results	≅ **125** ➔ see Figure 13 and workload 2000 MIPS						

TABLE 10. Analytic verification of the optimal choice in Scenario #4 at 1000 MIPS.

Checkpoint 3							
Optimal Solution *Capacitated problem, Scenario #4, all processing done at IoT layer with 1000 MIPS per IoT source node, 20 IoT devices in total.*							
Network Power (W)	-	-	-	-	-	-	-
Processing Power (W)	$10 + (20{,}000\ MIPS \times 0.034)$	-	-	-	-	-	79.2
Total Power (W)	**79.2**						
MILP Results	= **79.2** ➔ refer to Figure 13 and workload 1000 MIPS						

fog/ IoT environment i.e. a random number of workloads, random size of workloads, etc. through modeling techniques of stochastic linear programming optimization. Finally, it is worth looking into duty cycling approaches (deep sleep and/or shallow sleep) that can result in additional levels of power consumption savings of network equipment and

processing servers as it is clear that shutting nodes down completely poses latency challenges during the startup phase.

APPENDIX

We consider boundary cases whose optimum service placement solutions are known in advance and hence serves as verification of the MILP results using simple analytic closed-form expressions. The MILP model in section IV forms the basis of the works in the subsequent sections, hence it is crucial to validate the results of this model. The verification process is composed of several checkpoints, in which the proposed MILP model is validated against various carefully selected scenarios (boundary cases) consisting of the case where all the services are processed by (i) the general-purpose DC (GP-DC), (ii) the access fog (AF), (iii) the metro fog (MF) and (iv) finally the IoT devices. Although not exhaustive, however, the aforementioned checkpoints ensure that all the network and server elements between the IoT and the cloud are verified in terms of their networking and processing power consumption in one or more of the Selected boundary cases / verification scenarios. The total power consumption is verified by referring to the appropriate results figures in the evaluation section accordingly. The verifications are summarized in Table 8 – Table 10. In each table, reference is made to the scenario that is being validated. The column headings represent the power consumption at each layer of the evaluated architecture while the row headings consist of the network, processing, and total power consumption. It is important to note that, the proportional power consumption of the network is ignored (since it's negligible compared to the idle power consumption of the networking equipment (and the servers processing power consumption), hence the analytic calculations only account for idle network powers and processing power consumption, for the sake of simplicity. Also, for the equations in the processing row, the first value is the network power consumption of the processing node, the second is the server idle power consumption and the third is the proportional power consumption per instruction. All figures are calculated after PUE overhead.

ACKNOWLEDGMENT

Barzan A. Yosuf would like to thank EPSRC for providing his Doctoral Training Award Scholarship. All data are provided in full in the results section of this article.

REFERENCES

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

[2] M. H. Asghar, A. Negi, and N. Mohammadzadeh, "Principle application and vision in Internet of Things (IoT)," in *Proc. Int. Conf. Comput., Commun. Automat. (ICCCA)*, Jul. 2015, pp. 427–431, doi: 10.1109/CCAA.2015.7148413.

[3] O. Vermesan and P. Friess, *Internet of Things—From Research and Innovation to Market Deployment*. River Publishers, 2014.

[4] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018, doi: 10.1109/access.2017.2778504.

[5] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016, doi: 10.1109/JIOT.2016.2584538.

[6] B. J. Baliga, R. W. A. Ayre, K. Hinton, R. S. Tucker, and F. Ieee, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[7] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.

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

[9] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018, doi: 10.1109/JIOT.2017.2767608.

[10] O. Skarlat, S. Schulte, M. Borkowski, and P. Leitner, "Resource provisioning for IoT services in the fog," in *Proc. IEEE 9th Int. Conf. Service-Oriented Comput. Appl. (SOCA)*, Nov. 2016, pp. 32–39, doi: 10.1109/SOCA.2016.10.

[11] *Which IoT Applications Work Best With Fog Computing? | Network World*. Accessed: Oct. 25, 2019, [Online]. Available: https://www.networkworld.com/article/3147085/which-iot-applications-work-best-with-fog-computing.html

[12] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3909–3914, doi: 10.1109/ICC.2015.7248934.

[13] J. Ding, R. Yu, Y. Zhang, S. Gjessing, and D. H. K. Tsang, "Service provider competition and cooperation in cloud-based software defined wireless networks," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 134–140, Nov. 2015, doi: 10.1109/MCOM.2015.7321982.

[14] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *J. Netw. Comput. Appl.*, vol. 98, pp. 27–42, Nov. 2017, doi: 10.1016/j.jnca.2017.09.002.

[15] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016, doi: 10.1109/JSAC.2016.2545559.

[16] V. Angelakis, I. Avgouleas, N. Pappas, E. Fitzgerald, and D. Yuan, "Allocation of heterogeneous resources of an IoT device to flexible services," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 691–700, Oct. 2016, doi: 10.1109/JIOT.2016.2535163.

[17] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 14–20, Aug. 2017, doi: 10.1109/MCOM.2017.1600885.

[18] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A resilient Internet of Things architecture for smart cities," *Ann. Telecommun.*, vol. 72, nos. 1–2, pp. 19–30, Feb. 2017, doi: 10.1007/s12243-016-0530-y.

[19] F. Idzikowski, L. Chiaraviglio, A. Cianfrani, J. Lopez Vizcaino, M. Polverini, and Y. Ye, "A survey on energy-aware design and operation of core networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1453–1499, 2nd Quart., 2016, doi: 10.1109/COMST.2015.2507789.

[20] V. B. C. Souza, W. Ramirez, X. Masip-Bruin, E. Marin-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 0–4, doi: 10.1109/ICC.2016.7511465.

[21] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, "IoT-cloud service optimization in next generation smart environments," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4077–4090, Dec. 2016.

[22] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in fog-cloud computing paradigm," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 1222–1228, doi: 10.23919/INM.2017.7987464.

[23] E. Sturzinger, M. Tornatore, and B. Mukherjee, "Application-aware resource provisioning in a heterogeneous Internet of Things," in *Proc. Int. Conf. Opt. Netw. Design Modeling (ONDM)*, May 2017, pp. 1–6.

[24] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Resource provisioning for IoT application services in smart cities," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–9.

[25] I. Al Ridhawi, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, "A profitable and energy-efficient cooperative fog solution for IoT services," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3578–3586, May 2020, doi: 10.1109/TII.2019.2922699.

[26] N. Kaur and S. K. Sood, "An energy-efficient architecture for the Internet of Things (IoT)," *IEEE Syst. J.*, vol. 11, no. 2, pp. 796–805, Jun. 2017.
[27] M. Taheri and N. Ansari, "A feasible solution to provide cloud computing over optical networks," *IEEE Netw.*, vol. 27, no. 6, pp. 31–35, Nov. 2013, doi: 10.1109/MNET.2013.6678924.
[28] S. H. S. Newaz, W. S. B. H. Suhaili, G. M. Lee, M. R. Uddin, A. F. Y. Mohammed, and J. K. Choi, "Towards realizing the importance of placing fog computing facilities at the central office of a PON," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2017, pp. 152–157, doi: 10.23919/ICACT.2017.7890075.
[29] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan. 2018, doi: 10.1109/TCC.2015.2485206.
[30] S. Mondal, G. Das, and E. Wong, "A novel cost optimization framework for multi-cloudlet environment over optical access networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6, doi: 10.1109/GLOCOM.2017.8254251.
[31] H. Guo, J. Liu, and H. Qin, "Collaborative mobile edge computation offloading for IoT over fiber-wireless networks," *IEEE Netw.*, vol. 32, no. 1, pp. 66–71, Jan. 2018, doi: 10.1109/MNET.2018.1700139.
[32] M. Noreikis, Y. Xiao, and A. Yla-Jaaski, "QoS-oriented capacity planning for edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6, doi: 10.1109/ICC.2017.7997387.
[33] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019, doi: 10.1016/j.sysarc.2019.02.009.
[34] A. M. Al-Salim, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient big data networks: Impact of volume and variety," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 458–474, Mar. 2018, doi: 10.1109/TNSM.2017.2787624.
[35] A. M. Al-Salim, T. E. H. El-Gorashi, A. Q. Lawey, and J. M. H. Elmirghani, "Greening big data networks: Velocity impact," *IET Optoelectron.*, vol. 12, no. 3, pp. 126–135, Jun. 2018, doi: 10.1049/iet-opt.2016.0165.
[36] B. G. Bathula, M. Alresheedi, and J. M. H. Elmirghani, "Energy efficient architectures for optical networks," in *Proc. LCS*, London, U.K., 2009.
[37] B. G. Bathula and J. M. H. Elmirghani, "Energy efficient optical burst switched (OBS) networks," in *Proc. IEEE Globecom Workshops*, Nov./Dec. 2009, pp. 1–6, doi: 10.1109/GLOCOMW.2009.5360734.
[38] T. E. H. El-Gorashi, X. Dong, and J. M. H. Elmirghani, "Green optical orthogonal frequency-division multiplexing networks," *IET Optoelectron.*, vol. 8, no. 3, pp. 137–148, Jun. 2014, doi: 10.1049/iet-opt.2013.0046.
[39] H. M. Mohammad Ali, T. E. H. El-Gorashi, A. Q. Lawey, and J. M. H. Elmirghani, "Future energy efficient data centers with disaggregated servers," *J. Lightw. Technol.*, vol. 35, no. 24, pp. 5361–5380, Dec. 15, 2017, doi: 10.1109/JLT.2017.2767574.
[40] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "BitTorrent content distribution in optical networks," *J. Lightw. Technol.*, vol. 32, no. 21, pp. 3607–3623, Nov. 1, 2014, doi: 10.1109/JLT.2014.2351074.
[41] N. I. Osman, T. El-Gorashi, L. Krug, and J. M. H. Elmirghani, "Energy-efficient future high-definition TV," *J. Lightw. Technol.*, vol. 32, no. 13, pp. 2364–2381, Jul. 1, 2014, doi: 10.1109/JLT.2014.2324634.
[42] M. Musa, T. Elgorashi, and J. Elmirghani, "Energy efficient survivable IP-over-WDM networks with network coding," *J. Opt. Commun.*, vol. 9, no. 3, pp. 207–217, 2017, doi: 10.1364/JOCN.9.000207.
[43] A. N. Al-Quzweeni, A. Q. Lawey, T. E. H. Elgorashi, and J. M. H. Elmirghani, "Optimized energy aware 5G network function virtualization," *IEEE Access*, vol. 7, pp. 44939–44958, 2019, doi: 10.1109/ACCESS.2019.2907798.
[44] M. S. Hadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Patient-centric cellular networks optimization using big data analytics," *IEEE Access*, vol. 7, pp. 49279–49296, 2019, doi: 10.1109/ACCESS.2019.2910224.
[45] Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient IoT virtualization framework with peer to peer networking and processing," *IEEE Access*, vol. 7, pp. 50697–50709, 2019, doi: 10.1109/ACCESS.2019.2911117.
[46] H. Q. Al-Shammari, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Service embedding in IoT networks," *IEEE Access*, vol. 8, pp. 2948–2962, 2020, doi: 10.1109/ACCESS.2019.2962271.
[47] B. A. Yosuf, M. Musa, T. Elgorashi, and J. M. H. Elmirghani, "Impact of distributed processing on power consumption for IoT based surveillance applications," in *Proc. 21st Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2019, pp. 1–5, doi: 10.1109/ICTON.2019.8840023.
[48] B. Yosuf, M. Musa, T. Elgorashi, A. Q. Lawey, and J. M. H. Elmirghani, "Energy efficient service distribution in Internet of Things," in *Proc. 20th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2018, pp. 1–4, doi: 10.1109/ICTON.2018.8473659.
[49] B. A. Yosuf, *Energy Efficient Distributed Processing for IoT*. Leeds, U.K.: Univ. Leeds, 2019.
[50] R. Dhall and H. Agrawal, "An improved energy efficient duty cycling algorithm for IoT based precision agriculture," *Procedia Comput. Sci.*, vol. 141, pp. 135–142, 2018, doi: 10.1016/j.procs.2018.10.159.
[51] F. Safara, A. Souri, T. Baker, I. Al Ridhawi, and M. Aloqaily, "PriNergy: A priority-based energy-efficient routing method for IoT systems," *J. Supercomput.*, pp. 1–18, Jan. 2020, doi: 10.1007/s11227-020-03147-8.
[52] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018, doi: 10.1109/JIOT.2018.2805263.
[53] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the Internet of Things," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 17–24, doi: 10.1109/IEEE.EDGE.2017.12.
[54] F. Jalali, S. Khodadustan, C. Gray, K. Hinton, and F. Suits, "Greening IoT with fog: A survey," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 25–31, doi: 10.1109/IEEE.EDGE.2017.13.
[55] P. Chołda and P. Jaglarz, "Optimization/simulation-based risk mitigation in resilient green communication networks," *J. Netw. Comput. Appl.*, vol. 59, pp. 134–157, Jan. 2016, doi: 10.1016/J.JNCA.2015.07.009.
[56] G. Shen and R. S. Tucker, "Energy-minimized design for IP over WDM networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 1, no. 1, pp. 176–186, Jun. 2009.
[57] Thepihut. *USB Wifi Adapter for the Raspberry Pi*. Accessed: Mar. 21, 2019. [Online]. Available: https://thepihut.com/products/raspberry-pi-zero-w
[58] RasPi.TV. *How Much Power Does Pi Zero W Use?*. Accessed: Mar. 21, 2019. [Online]. Available: http://raspi.tv/2017/how-much-power-does-pi-zero-w-use
[59] *Cisco ME 4600 Series Optical Network Terminal Data Sheet-Cisco*. Accessed: Mar. 16, 2018. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/switches/me-4600-series-multiservice-optical-access-platform/datasheet-c78-730446.html
[60] *Cisco ME 4600 Series Optical Line Terminal Data Sheet-Cisco*. Accessed: Nov. 20, 2019. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/switches/me-4600-series-multiservice-optical-access-platform/datasheet-c78-730445.html
[61] *Cisco Network Convergence System 5500 Series Modular Chassis Data Sheet-Cisco*. Accessed: Oct. 26, 2019. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/routers/network-convergence-system-5500-series/datasheet-c78-736270.html
[62] *Cisco Nexus 9300-FX Series Switches Data Sheet-Cisco*. Accessed: Oct. 26, 2019. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/datasheet-c78-742284.html
[63] C. Gray, R. Ayre, K. Hinton, and R. S. Tucker, "Power consumption of IoT access network technologies," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 2818–2823, doi: 10.1109/ICCW.2015.7247606.
[64] C. Delgado, J. R. Gállego, M. Canales, J. Ortín, S. Bousnina, and M. Cesana, "On optimal resource allocation in virtual sensor networks," *Ad Hoc Netw.*, vol. 50, pp. 23–40, Nov. 2016, doi: 10.1016/j.adhoc.2016.04.004.
[65] M. Kašpar. *Bandwidth Calculator*. Accessed: Mar. 20, 2019. [Online]. Available: https://www.cctvcalculator.net/en/calculations/bandwidth-calculator/
[66] *Tensor Cores in NVIDIA Volta Architecture | NVIDIA*. Accessed: Oct. 16, 2019. [Online]. Available: https://www.nvidia.com/en-gb/data-center/tensorcore/
[67] *Intel Xeon Processor E5-2680 (20M Cache, 2.70 GHz, 8.00 GT/s Intel QPI) Product Specifications*. Accessed: Oct. 26, 2019. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/64583/intel-xeon-processor-e5-2680-20m-cache-2-70-ghz-8-00-gt-s-intel-qpi.html
[68] *Intel Xeon Processor X5675 (12M Cache, 3.06 GHz, 6.40 GT/s Intel QPI) Product Specifications*. Accessed: Oct. 26, 2019. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/52577/intel-xeon-processor-x5675-12m-cache-3-06-ghz-6-40-gt-s-intel-qpi.html

[69] *Intel Xeon Processor E5-2420 (15M Cache, 1.90 GHz, 7.20 GT/s Intel QPI) Product Specifications*. Accessed: Oct. 26, 2019. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/64617/intel-xeon-processor-e5-2420-15m-cache-1-90-ghz-7-20-gt-s-intel-qpi.html
[70] *FAQs-Raspberry Pi Documentation*. Accessed: Oct. 26, 2019. [Online]. Available: https://www.raspberrypi.org/documentation/faqs/
[71] *Raspberry Pi 3: Specs, Benchmarks & Testing-The MagPi Magazine*. Accessed: Oct. 26, 2019. [Online]. Available: https://magpi.raspberrypi.org/articles/raspberry-pi-3-specs-benchmarks
[72] *Raspberry Pi Zero W (Wireless) | The Pi Hut*. Accessed: Oct. 26, 2019. [Online]. Available: https://thepihut.com/products/raspberry-pi-zero-w
[73] (2010). *8-Port 10/100 Switch*. Accessed: Oct. 16, 2019. [Online]. Available: https://dlink-me.com/pdf/DES-1008A.pdf.
[74] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: Eliminating server idle power," in *Proc. Int. Conf. Architectural Support Program. Lang. Operating Syst.*, 2009, pp. 205–216.
[75] *Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper-Cisco*. Accessed: Oct. 26, 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html
[76] (2016). *Cisco Industrial Benchmark*. Accessed: Mar. 16, 2018. [Online]. Available: https://www.cisco.com/c/dam/global/da_dk/assets/docs/presentations/vBootcamp_Performance_Benchmark.pdf
[77] M. Y. Arslan, I. Singh, S. Singh, H. V. Madhyastha, K. Sundaresan, and S. V. Krishnamurthy, "Computing while charging: Building a distributed computing infrastructure using smartphones," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2012, pp. 193–204, doi: 10.1145/2413176.2413199.
[78] A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Distributed energy efficient clouds over core networks," *J. Lightw. Technol.*, vol. 32, no. 7, pp. 1261–1281, Apr. 1, 2014, doi: 10.1109/JLT.2014.2301450.
[79] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *J. Lightw. Technol.*, vol. 33, no. 9, pp. 1828–1849, May 1, 2015, doi: 10.1109/JLT.2014.2380777.
[80] *America-Google Maps*. Accessed: Dec. 24, 2019. [Online]. Available: https://www.google.com/maps/search/america/@42.593777,-113.8013893,5z
[81] J. M. H. Elmirghani, "GreenTouch GreenMeter core network energy-efficiency improvement measures and optimization," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 2, pp. A250–A269, Feb. 2018, doi: 10.1364/JOCN.10.00A250.
[82] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-defined fog network architecture for IoT," *Wireless Pers. Commun.*, vol. 92, no. 1, pp. 181–196, Jan. 2017, doi: 10.1007/s11277-016-3845-0.
[83] '*Skynet', China's Massive Video Surveillance Network | Abacus*. Accessed: Oct. 25, 2019. [Online]. Available: https://www.abacusnews.com/who-what/skynet-chinas-massive-video-surveillance-network/article/2166938
[84] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, "Minimization of network power consumption with redundancy elimination," *Comput. Commun.*, vol. 59, pp. 98–105, Mar. 2015, doi: 10.1016/j.comcom.2014.12.002.
[85] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Francisco, CA, USA: Morgan Kaufmann, 2004.
[86] T. Bahreini and D. Grosu, "Efficient placement of multi-component applications in edge computing systems," in *Proc. 2nd ACM/IEEE Symp. Edge Comput.*, Oct. 2017, pp. 1–11, doi: 10.1145/3132211.3134454.
[87] M. Ashouri, P. Davidsson, and R. Spalazzese, "Cloud, edge, or both? Towards decision support for designing IoT applications," in *Proc. 5th Int. Conf. Internet Things: Syst., Manage. Secur.*, Oct. 2018, pp. 155–162, doi: 10.1109/IoTSMS.2018.8554827.
[88] K. Benson, "Enabling resilience in the Internet of Things," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 230–232, doi: 10.1109/PERCOMW.2015.7134032.
[89] H. A. Alharbi, T. E. H. Elgorashi, and J. M. H. Elmirghani, "Energy efficient virtual machines placement over cloud-fog network architecture," *IEEE Access*, vol. 8, pp. 94697–94718, 2020, doi: 10.1109/ACCESS.2020.2995393.
[90] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United states data center energy usage report," Lawrence Berkeley Natl. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-1005775, Jun. 2016, pp. 1–66.

BARZAN A. YOSUF (Member, IEEE) received the B.Eng. degree (Hons.) in computer systems engineering and the M.Sc. degree in embedded systems engineering from the University of Huddersfield, U.K., in 2012 and 2013, respectively, and the Ph.D. degree in energy efficient distributed processing for the IoT with the School of Electronic and Electrical Engineering, University of Leeds, U.K., in 2019. He is currently a Research Fellow in network architecture with the University of Leeds and a member of the IEEE Energy Efficient ICT Working Group, which focuses on developing improved energy efficient standards in areas, such as big data processing over core networks, network virtualization, and VM placement in the presence of access fog and core clouds. His current research interests include the Internet of Things (IoT), cloud fog architectures, energy efficiency, and optimization.

MOHAMED MUSA (Member, IEEE) received the B.Sc. degree (Hons.) in electrical and electronic engineering from the University of Khartoum, Khartoum, Sudan, in 2009, and the M.Sc. degree (Hons.) in broadband wireless and optical communication and the Ph.D. degree in energy efficient network coding in optical networks from the University of Leeds, Leeds, U.K., in 2011 and 2016, respectively. His current research interests include ICT energy optimization, network coding, and energy efficient routing protocols in optical networks. He chairs working group for the IEEE P1928.1 Standard and the Secretary of the IEEE Green ICT Standards Committee.

TAISIR ELGORASHI received the B.S. degree (Hons.) in electrical and electronic engineering from the University of Khartoum, Khartoum, Sudan, in 2004, the M.Sc. degree (Hons.) in photonic and communication systems from the University of Wales, Swansea, U.K., in 2005, and the Ph.D. degree in optical networking from the University of Leeds, Leeds, U.K., in 2010. She is a currently Lecturer in optical networks with the School of Electrical and Electronic Engineering, University of Leeds. She held a postdoctoral research position at the University of Leeds, from 2010 to 2014, where she focused on the energy efficiency of optical networks investigating the use of renewable energy in core networks, green IP over WDM networks with data centers, energy efficient physical topology design, the energy efficiency of content distribution networks, distributed cloud computing, network virtualization, and big data. She was a BT Research Fellow and developed energy efficient hybrid wireless optical broadband access networks and explored the dynamics of TV viewing behavior and program popularity, in 2012. The energy efficiency techniques developed during the postdoctoral research contributed three out of eight carefully chosen core network energy efficiency improvement measures recommended by the GreenTouch consortium for every operator network worldwide. Her work led to several invited talks at GreenTouch, Bell Labs, the Optical Network Design and Modeling Conference, the Optical Fiber Communications Conference, the International Conference on Computer Communications, and the EU Future Internet Assembly in collaboration with Alcatel-Lucent and Huawei.

JAAFAR ELMIRGHANI (Senior Member, IEEE) received the Ph.D. degree in synchronization of optical systems and optical receiver design from the University of Huddersfield, U.K., in 1994, and the D.Sc. degree in communication systems and networks from the University of Leeds, U.K., in 2014. He was the Chair of optical communications with the University of Wales, Swansea, from 2000 to 2007. He joined the University of Leeds, in 2007. He has founded, developed, and directed the Institute of Advanced Telecommunications and the Technium Digital (TD), a technology incubator/spin-off hub. He is currently the Director of the School of Electronic and Electrical Engineering, Institute of Communication and Power Networks, University of Leeds. He has provided outstanding leadership in a number of large research projects at IAT and TD. He has coauthored *Photonic Switching Technology: Systems and Networks* (Wiley) and has published over 500 articles. His research interests include optical systems and networks. He is a Fellow of IET and the Institute of Physics. He received the IEEE Communications Society Hal Sobol Award, the IEEE ComSoc Chapter Achievement Award for excellence in chapter activities, in 2005, the University of Wales Swansea Outstanding Research Achievement Award, in 2006, the IEEE Communications Society Signal Processing and Communication Electronics Outstanding Service Award, in 2009, the Best Paper Award at the IEEE ICC 2013, the 2015 IEEE ComSoc Transmission, Access, and Optical Systems Outstanding Service Award in recognition of the leadership and contributions to the area of green communications, the GreenTouch 1000x Award, in 2015, for pioneering research contributions to the field of energy efficiency in telecommunications, the 2016 IET Optoelectronics Premium Award, and shared with six GreenTouch innovators the 2016 Edison Award in the Collective Disruption Category for their work on the GreenMeter, an international competition, clear evidence of his seminal contributions to green communications, which have a lasting impact on the environment (green) and society. He has been awarded in excess of the £30 million in grants to date from EPSRC, the EU, and industry and has held prestigious fellowships funded by the Royal Society and BT. He was the Co-Chair of the GreenTouch Wired, Core and Access Networks Working Group, an Adviser of the Commonwealth Scholarship Commission, and a member of the Royal Society International Joint Projects Panel and the Engineering and Physical Sciences Research Council (EPSRC) College. He was the Founding Chair of the Advanced Signal Processing for Communication Symposium, which started at the IEEE GLOBECOM 1999 and has continued since at every ICC and GLOBECOM. He was also the Founding Chair of the IEEE ICC/GLOBECOM Optical Symposium at GLOBECOM 2000, the Future Photonic Network Technologies, Architectures, and Protocols Symposium. He chaired this symposium, which continues to date under different names. He was also the Founding Chair of the Green Track at the ICC/GLOBECOM at GLOBECOM 2011 and is the Co-Chair of the IEEE Sustainable ICT Initiative within the IEEE Technical Activities Board (TAB) Future Directions Committee (FDC) and the IEEE Communications Society, a pan IEEE Societies Initiative responsible for Green and Sustainable ICT activities across IEEE, since 2012. He has been on the Technical Program Committee of 38 IEEE ICC/GLOBECOM conferences, from 1995 to 2019, including 18 times as the Symposium Chair. He was the Chairman of the IEEE ComSoc Transmission, Access, and Optical Systems Technical Committee and the IEEE ComSoc Signal Processing and Communications Electronics Technical Committee. He was an Editor of the IEEE Communications Surveys and Tutorials, the IEEE Journal on Selected Areas in Communications Series on Green Communications and Networking, and *IEEE Communications Magazine*. He is an Editor of *IET Optoelectronics* and the *Journal of Optical Communications and Networking*. He was the Principal Investigator (PI) of the £6m EPSRC INTelligent Energy awaRe NETworks (INTERNET) Programme Grant, from 2010 to 2016, and is the PI of the £6.6m EPSRC Terabit Bidirectional Multi-User Optical Wireless System (TOWS) for 6G LiFi Programme Grant (for the period of 2019–2024). He was an IEEE ComSoc Distinguished Lecturer, from 2013 to 2016.