Check for updates

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 8593808, 17 pages
https://doi.org/10.1155/2019/8593808

WILEY | Hindawi

Research Article

Online Supervisory Control and Resource Management for Energy Harvesting BS Sites Empowered with Computation Capabilities

Thembelihle Dlamini,[1,2] **Ángel Fernández Gambín**,[1] **Daniele Munaretto**,[2] **and Michele Rossi**[1]

[1]*Department of Information Engineering, University of Padova, Padova, Italy*
[2]*Athonet, Bolzano Vicentino, Vicenza, Italy*

Correspondence should be addressed to Thembelihle Dlamini; dlamini@dei.unipd.it

Received 24 October 2018; Accepted 3 February 2019; Published 19 February 2019

Academic Editor: Gianluigi Ferrari

Copyright © 2019 Thembelihle Dlamini et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The convergence of communication and computing has led to the emergence of *multi-access edge computing (MEC)*, where computing resources (supported by virtual machines (VMs)) are distributed at the edge of the mobile network (MN), i.e., in base stations (BSs), with the aim of ensuring reliable and ultra-low latency services. Moreover, BSs equipped with *energy harvesting (EH)* systems can decrease the amount of energy drained from the power grid resulting into energetically self-sufficient MNs. The combination of these paradigms is considered here. Specifically, we propose an online optimization algorithm, called Energy Aware and Adaptive Management (ENAAM), based on foresighted control policies exploiting (short-term) traffic load and harvested energy forecasts, where BSs and VMs are dynamically switched on/off towards energy savings and Quality of Service (QoS) provisioning. Our numerical results reveal that ENAAM achieves energy savings with respect to the case where no energy management is applied, ranging from 57% to 69%. Moreover, the extension of ENAAM within a cluster of BSs provides a further gain ranging from 9% to 16% in energy savings with respect to the optimization performed in isolation for each BS.

1. Introduction

The full potential of 5G radio access technology can be realized through the use of distributed intelligence, whereby content, control, and computation are moved closer to mobile users, hereby referred to as the *network edge*. This evolution has led to the emergence of the multi-access edge computing (MEC) paradigm, which allows network functions to be virtualized and then deployed at the network edge to guarantee the low latency required by some applications. In this paper, we consider a hybrid edge computing architecture where computing servers are co-located with each base station (BS), and a centralized controller (a point within range to a set of BSs) is utilized to manage them, deciding upon the allocation of their computing and transmission resources. This type of architecture is in line with recent trends [1].

The convergence of communication and computing (MEC [2]) within the mobile space poses new challenges related to energy consumption, as BSs are densely deployed to maximize capacity and also empowered with computing capabilities to minimize latency. To cope with these challenges, previous studies have put forward BS sleep modes [3, 4], as BSs are dimensioned for the expected maximum capacity, yet traffic varies during the day. In addition, energy savings within the virtualized computing platform are of great importance, as virtualization can also lead to energy overheads. Therefore, a clear understanding and a precise modeling of the server energy usage can provide a fundamental basis for server operational optimizations. The experimental results in [5, 6] show that the locus of energy consumption for the Virtualized Network Function (VNF) components is the virtual machine (VM) instance where the VNF is instantiated and executed. Thus, for a given expected

traffic load, the energy consumption can be minimized by launching an optimal number of VMs, a technique referred to as VM *soft-scaling*, together with BS power saving methods, i.e., BS sleep modes.

Along these lines, we propose a controller-based network architecture for managing energy harvesting (EH) BSs empowered with computation capabilities where on/off switching strategies allow BSs and VMs to be dynamically switched on/off, depending on the traffic load and the harvested energy forecast, over a given look-ahead prediction horizon. To solve the energy consumption minimization problem in a distributed manner, the *controller* partitions the BSs into clusters based on their location; then, for each cluster, it minimizes a cost function capturing the individual communication site energy consumption and the users' Quality of Service (QoS). To manage the communication sites, the controller performs online supervisory control by forecasting the traffic load and the harvested energy using a Long Short-Term Memory (LSTM) neural network [7], which is utilized within a Limited Look-ahead Control (LLC) policy (a predictive control approach [8]) to obtain the system control actions that yield the desired trade-off between energy consumption and QoS. This work is an extension of [9], where we consider energy savings within a single off-grid BS scenario (i.e., BS powered by either wind or solar energy sources) taking into account the need for MEC in remote/rural areas. In this paper, however, a dense environment is considered, similar to an urban or semi-urban scenario, where each BS is powered by hybrid energy supplies (solar and power grid) and empowered with computation capabilities. Moreover, the optimization problem is extended for multiple BSs where energy management procedures are executed within a BS cluster in contrast with the single BS case of [9].

The rest of the paper is structured as follows. The related work is discussed in Section 2, and the system model is presented in Section 3. In Section 4, we detail the optimization problem and the proposed LLC-based online algorithm for a *single* communication site. The *multiple* BS communication site case is addressed in Section 5. Our contribution is evaluated in Section 6, and, lastly, concluding remarks are given in Section 7.

2. Related Work and Paper Contribution

Next, we first provide a literature review related to BS sleep modes techniques. Then, we review the mathematical tools that we use in this paper, followed by the literature review related to energy savings in virtualized computing platforms (i.e., works related to soft-scaling). Finally, we put forward our contributions and novelty of our work.

Sleep-Mode Strategies in Mobile Networks. Cellular networks are dimensioned to support traffic peaks; i.e., the number of BSs deployed in a given area should be able to provide the required QoS to the mobile subscribers during the highest load conditions. However, during off-peak periods the network may be underutilized, which leads to an inefficient use of network resources and to excessive energy consumption. For these reasons, sleep modes have been proposed to dynamically turn off some of the BSs when the traffic load is low. This has been extensively studied in the literature; here we highlight the main applied techniques that are related to this work.

Clustering algorithms have been proposed as a way of switching off BSs to reduce the energy consumption. In [12], centralized and distributed algorithms group BSs exhibiting similar traffic profiles over time. In [13], a dynamic switching on/off mechanism locally groups BSs into clusters based on location and traffic load. The optimization problem is formulated as a non-cooperative game aimed at minimizing the BS energy consumption and the time required to serve their traffic load. Simulation results show energy costs and load reductions, while also providing insights of when and how the cluster-based coordination is beneficial.

Reducing the energy consumption involves some trade-offs in the optimization problem. QoS has been widely used as a trade-off metric [14, 15]. The Quality of Experience (QoE) is included in [16], where a dynamic programming switching algorithm is put forward. Other parameters that have been considered are the coverage probability and the BS state stability parameter, i.e., the number of on/sleep state transitions. For instance, a set of BSs switching patterns engineered to provide full network coverage at all times, while avoiding channel outage, is presented in [17]. According to the BS state stability concept, a two-objective optimization problem is formulated in [18] and solved with two algorithms: (i) near optimal but not scalable and (ii) low complexity, based on particle swarm optimization. The QoE is also affected by the UE position due to channel propagation phenomena. To this respect, in [19] the selection of the BSs to be switched off is taken so as to minimize the impact on the UEs' QoE, according to the distance from the handed off BSs.

To support sleep modes, neighboring cells must be capable of serving the traffic from the switched off cells. To achieve this, proper *user association* strategies are required. A framework to characterize the performance (outage probability and spectral efficiency) of cellular systems with sleeping techniques and user association rules is proposed in [20]. In that paper, the authors devise a user association scheme where a user selects its serving BS considering the maximum expected channel access probability. This strategy is compared against the traditional maximum SINR-based user association approach and is found superior in terms of spectral efficiency when the traffic load is inhomogeneous. User association mechanisms that maximize energy efficiency in the presence of sleep modes are addressed in [21]. There, a downlink HetNet scenario is considered, where the energy efficiency is defined as the ratio between the network throughput and the total energy consumption. Since this leads to a rather complex integer optimization problem, the authors propose a quantum particle swarm optimization algorithm to obtain a suboptimal solution.

A marketing approach to foster the opportunistic utilization of the unexploited small cell (SC) BS capacity in dense heterogeneous networks (HetNets) is presented in [22]. There, an offloading mechanism is introduced, where the

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

operators lease the capacity of a SC network owned by a third party in order to switch off their BSs (macro-BSs) and maximize their energy efficiency, when the traffic demand is low. The allocation of the SC resources among a set of competing operators is mathematically formulated as an auction problem.

A comprehensive power management model employing a BS switching on/off mechanism, within a BS system powered by green energy, is presented in [23]. The model considers weather conditions, user mobility, different green energy harvesting rates, energy storage with self-discharge effect, and switching on/off frequency. The authors propose two algorithms: the first decides which BSs are to be active based on the minimum energy cost, i.e., the energy price per time period, while the second one determines the active BSs by first prioritizing the minimum power consumption of the system and then the energy cost. The relationship between installing a solar harvesting system to power a BS and the energy management under varying demand is investigated in [24]. The authors present a solar installation planning model by explicitly modeling solar panels, batteries, inverters, and charge controllers, as well as the cellular network demand and energy management. They found that the solar installation and the energy management of the base stations are so coupled that even the order in which these technologies are introduced can have a major impact on the network cost and performance.

The survey paper [25] presents taxonomy of existing energy sustainable paradigms and methods to address energy savings in network elements (i.e., BSs) equipped with EH capabilities. Here, the authors discuss the shortcomings of previous studies related to efficient energy management procedures, the lack of relevant discussion related to the integration of EH into future networks, and, lastly, energy self-sustainability in future networks. The current work is a technical contribution where we address some of the shortcomings that were identified in [25], also proposing the use of machine learning (ML) tools for pattern forecasting and adaptive control schemes for decision-making. In addition, this work is in line with the research topics which can be found in our review paper [26].

The majority of the works on BS switching off mechanism considered clusters of BSs from a *single* mobile operator perspective, where some functions of the BS can be switched off and then the remaining active BSs handle the upcoming traffic. A new approach is presented in [27] which exploits the coexistence of multiple BSs from different mobile operators in the same area. An intracell roaming-based infrastructure-sharing strategy is proposed, followed by a distributed game-theoretic switching off scheme that takes into account the conflicts and interaction among the different operators. Moreover, in [28], the authors investigate the energy and cost efficiency of multiple HetNets (i.e., each HetNet is composed of eNodeBs (eNBs) and SC BSs from one operator) that share their infrastructure and also are able to switch off part of it. Here, a form of roaming-based sharing is also adopted, whereby the operator can roam its traffic to a rival operator during a predefined period of time and area. An energy efficient optimization problem is formulated and solved using a cooperative greedy heuristic algorithm. Regarding the cost efficiency, the cooperation and cost sharing decisions among the operators are modeled using a Shapley Value based bankruptcy game.

Pattern Forecasting along with Foresighted Optimization. Control-theoretic and machine learning (ML) methods for resource management have been successfully applied to various problems, e.g., task scheduling, bandwidth allocation, and network management policies. In the paradigm of supervisory control for managing mobile networks (MNs), online forecasting using ML techniques and the LLC method can yield the desired system behavior when taking into account the environmental expectations, i.e., traffic load and energy to be harvested. Next, we briefly review the mathematical tools that we use in this paper, namely, the LLC method and LSTM neural network [7].

Control-theoretic algorithms and the LLC method have been used to obtain control actions that optimize the system behavior, by employing a forecasting mathematical model, over a limited look-ahead prediction horizon. LLC is conceptually similar to model predictive control (MPC) [29]. In [30], an online supervisory control scheme based on LLC policies is proposed. Here, after the occurrence of an event, the next control action is determined by estimating the system behavior a few steps into the future, using the currently available information as inputs. The control action exploration is performed using a search tree assuming that the controller knows all future possible states of the process over the prediction horizon. Moreover, in [8], an online control framework for resource management in switching hybrid systems is proposed, where the system's control inputs are finite. The relevant parameters of the operating environment, e.g., workload arrival, are estimated and then used by the system to forecast future behavior over a look-ahead horizon. From this, the controller optimizes the predicted system behavior following the specified QoS through the selection of the system controls.

To model time series datasets, the LSTM network is used as it is able to handle the long-term dependencies due to its inherent capability of storing past information and then recalling it. In [31], a distributed LSTM online method based on the particle filtering algorithm is presented with an aim of investigating the performance of online training of LSTM architectures in a distributed network of nodes. An LSTM based model for variable length data regression is proposed and then put into a nonlinear state-space form to train the model in an online fashion. Then, financial and real life datasets are used for performance evaluation, and it is observed that the distributed online approach yields the same results that are obtained in the centralized case, when considering the mean square errors as the performance measure. Moreover, an LSTM forecasting method is utilized in [9] within an LLC-based algorithm to obtain the system control actions yielding the desired trade-off between energy consumption and QoS, for a remote site powered by only green energy.

Energy Savings in Virtualized Platforms through Soft-Scaling. With the advent of virtualization, it is expected that the Network Function Virtualization (NFV) framework can

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

exploit the benefits of virtualization technologies to significantly reduce the energy consumption of large scale network infrastructures. In virtualized computing environments, the locus of energy consumption for components is due to the VMs running in the server(s). Thus, energy saving studies within the virtualized computing environment have involved the scaling down of the number of computing nodes/servers (autoscaling [32]), VM migration [33] (movement of a VM from one host to another), and soft resource scaling [34] (shortening of the access time to physical resources), all hereby referred to as VM *soft-scaling*, i.e., the reduction of computing resources per time instance.

Algorithms for the dynamic on/off switching of servers have been proposed as a way of minimizing energy consumption in computing platforms. In [32], at the beginning of each time slot computing resources are provisioned depending on the expected server workloads via a reinforcement learning-based resource management algorithm, which learns on-the-fly the optimal policy for dynamic workload offloading and the autoscaling of servers. Then in [9], computing resources (VMs) are provisioned based on a LLC policy after forecasting the future workloads and harvested energy. In [33], the Central Processing Unit (CPU) utilization thresholds are used to identify overutilized servers. Hence, migration policies, enabled by the live VM migration method [35], are applied for moving the VMs between physical nodes (servers). The VMs are only moved to hosts that will accept them without incurring high energy cost, i.e., without any increase in the CPU utilization. Subsequently, the idle servers are switched off.

Power management is also of interest in virtualized computing platforms, i.e., data centers using virtualization technologies. In [34], a power management approach called *VirtualPower* is presented. The algorithm exploits hardware power scaling, i.e., the dynamic power management strategies using Dynamic Voltage and Frequency Scaling (DVFS) [36, 37], and software-based methods, i.e., scaling the allocation of physical resources to VMs using the hypervisor scheduler, for controlling the power consumption of underlying platforms. Due to the low power management benefits obtained from hardware scaling, a *soft resource scaling* mechanism is proposed whereby the scheduler shortens the maximum resource usage time for each VM, i.e., the time slice allocated for using the underlying physical resources.

Novelty of this Work. Here, we consider the aforementioned scenario, where each BS is equipped with EH hardware (a solar panel for EH and an Energy Buffer (EB) for energy storage) and a MEC server co-located with the BS for computation purposes, under the management enabled by the *controller*.

Motivated by the potential capabilities of EH and MEC and the presence of the controller,

(1) we introduce the use of virtualization with the aim of investigating how VMs can be soft-scaled based on the forecasted server workloads, as VMs are the source of energy consumption in computing environments;

(2) we put forward the edge controller-based architecture for small cell BSs management, as one of the future trends for small cells [1] in 5G MNs;

(3) we reconsider the BS sleeping control mechanism under the new MEC paradigm, which has not been sufficiently covered in the literature. In addition, we use a clustering method for enabling energy savings within the MN;

(4) we estimate the short-term future traffic load and harvested energy in BSs, by using LSTM neural network [38];

(5) we develop an online supervisory control algorithm for the radio access (edge) network management based on a predictive method, specifically the LLC method, along with clustering and energy management procedures. The main goal is to enable energy savings (ES) strategies within the access network, BS sleep modes, and VM soft-scaling, following the energy efficiency requirements of a virtualized infrastructure from [39]. The proposed management algorithm is called Energy Aware and Adaptive Management (ENAAM) and is hosted in the edge controller. The ENAAM algorithm considers the future BS traffic load, onsite green energy in the EB, and then provisions access network resources, per communication site, based on the learned information; i.e., energy saving decisions are made in a forward-looking fashion.

The proposed optimization strategy leads to a considerable reduction in the energy consumed by the edge computing and communication facilities, promoting self-sustainability within the mobile network through the use of green energy. This is achieved under the controller guidance, which makes use of forecasting, clustering, control theory, and heuristics.

3. System Model

As a major deployment of MEC and in line with current trends for future mobile networks as suggested by prominent network operators (e.g., Huawei Technologies [1]), the considered scenario is illustrated in Figure 1. It consists of a densely deployed MN featuring N BSs and colocated cache-enabled MEC servers. Each MEC server hosts M VMs. Each communication site, i.e., the BS and the colocated MEC server, is empowered with EH capabilities through a solar panel and an EB that enables energy storage. Energy supply from the power grid is also available. Moreover, the Energy Manager (EM) is an entity responsible for selecting the appropriate energy source and for monitoring the energy level of the EB. All BSs communicate with a centralized entity called the *edge controller*, which is responsible for managing the access network apparatuses. The energy level information is reported periodically to the edge controller through the pull file transfer mode procedure (e.g., File Transfer Protocol [40]). Moreover, we consider a discrete-time model, whereby time is discretized as $t = 1, 2, \ldots$, and each time slot t has a

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

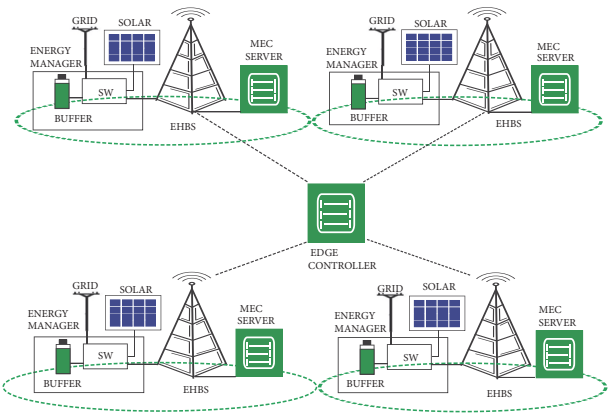


Figure 1: Edge network topology. The electromechanical switch (SW) selects the appropriate source of energy.

fixed duration τ. The list of symbols that are used in the paper is reported in Table 1.

3.1. Traffic Load and Energy Consumption. Mobile traffic volume exhibits temporal and spatial diversity and also follows a diurnal behavior [41]. Therefore, traffic volume at individual BSs can be estimated using historical mobile traffic datasets. In this paper, real MN traffic load traces obtained from the Big Data Challenge organized by Telecom Italia Mobile (TIM) [10] are used to emulate the computational load (in fact, the dataset is not a true representative of future applications that require processing at the edge but contains data that is exchanged with the purpose of communication. We nevertheless use it due to the difficulties in finding open datasets containing computing requests). Specifically, the used data was collected in the city of Milan during the month of November 2013, and it is the result of users interaction within the TIM MN, based on Call Detail Record (CDR) files for a day considering four BS sites representing the traffic load profiles. A CDR file consists of SMS, Calls, and Internet records with timestamps. To understand the behavior of the mobile data, we have applied the X-means clustering algorithm [42] to classify the load profiles into several categories. In our numerical results, each BS $n = 1, 2, \ldots, N$ is assigned a load profile $L_n(t)$, which is picked at random as one of the four clusters (each cluster represents a typical BS load profile) in Figure 2. $L_n(t)$ consists of computation workloads $\Gamma_n(t)$ ([MB]) and standard workloads $\Gamma'_n(t)$ ([MB]). According to [43], we assume that 80% of $L_n(t)$ is *delay sensitive* and, as such, requires processing at the edge, i.e., $\Gamma_n(t) = 0.8L_n(t)$, whereas the remaining 20% pertains to standard flows, *delay tolerant* traffic, i.e., $\Gamma'_n(t) = L_n(t) - \Gamma_n(t)$.

The total energy consumption ([J]) for the communication site n at time slot t is formulated as follows, inspired by [9, 44–47]

$$\theta_{\text{tot},n}(t) = \theta_{\text{BS},n}(t) + \theta_{\text{MEC},n}(t) + \theta_{\text{TX},n}(t), \quad (1)$$

where $\theta_{\text{BS},n}(t)$ is the BS energy consumption term, $\theta_{\text{MEC},n}(t)$ is the MEC server consumption term due to computation activities, and $\theta_{\text{TX},n}(t)$ represents the data transmission energy consumption between the BS and the MEC server.

Table 1: Notation: list of symbols used in the analysis.

Symbol	Description
	Input Parameters
N	number of BSs, indexed by n
M	maximum number of VMs hosted by each MEC server
τ	time slot duration
$L_n(t)$	BS n traffic load profile in time slot t, n is the BS index
$\Gamma_n(t)$	workload handled by the MEC server at BS n in time slot t
$\Gamma'_n(t)$	standard (non MEC) traffic at time t
θ_0	BS load independent energy consumption or operation energy
$f_{\max}$	maximum processing rate for VM m
$\mathcal{F}$	a finite set of available processing rates for VM m
$\theta_m^{\text{ov}}(t)$	energy overheads incurred when turning on/off VMs
$\theta_{\text{idle},m}(t)$	static energy consumed by VM m in the idle state
$\theta_{\max,m}(t)$	maximum energy consumed by VM m at maximum processing rate
$\gamma_m(t)$	workload fraction to be computed by the m-th VM
$\gamma^{\max}$	maximum computation load per-VM
Δ	maximum per-slot and per-VM allowed processing time
$\theta_{\text{idle}}(t)$	energy consumption of network interfaces in idle mode
$\theta_{\text{data}}(t)$	energy cost of exchanging one unit of data between the server and the BS
$\beta_{\max}$	maximum energy buffer capacity
$\beta_{\text{up}}, \beta_{\text{low}}$	upper and lower energy buffer thresholds
	Variables
$\theta_{\text{tot},n}(t)$	total energy consumption for the communication site n
$\theta_{\text{BS},n}(t)$	BS n energy cost at t
$\theta_{\text{MEC},n}(t)$	server consumption due to computation activities
$\theta_{\text{TX},n}(t)$	data transmission energy consumption between the BS and the MEC server
$\zeta_n(t)$	BS n switching status indicator at t
$M(t)$	number of VMs to be active in time slot t
$\theta_{\text{load}}(t)$	total wireless transmission power
$f_m(t)$	instantaneous processing rate
$\theta_m^{\text{op}}(t)$	energy consumption of VM m operation
$\alpha_m(t)$	load dependent factor
$\mu_m(t)$	the expected processing time
$B_n(t)$	the total amount of load that is served by the BS site
$\beta_n(t)$	energy buffer level in slot t
$H_n(t)$	harvested energy profile in slot t
$Q_n(t)$	purchased grid energy in slot t

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

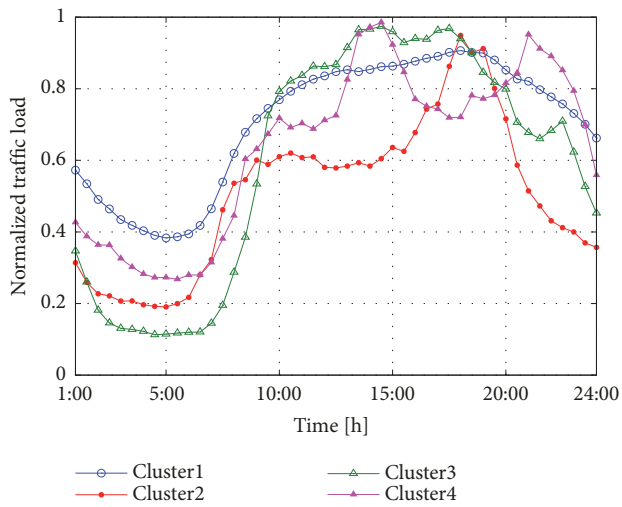


FIGURE 2: Example traces for normalized BS traffic loads. The data from [10] has been split into four representative clusters.

BS Energy Consumption. $\theta_{\mathrm{BS},n}(t) = \zeta_n(t)\theta_0 + \theta_{\mathrm{load}}(t)$, where $\zeta_n(t) \in \{\varepsilon, 1\}$ is the BS switching status indicator (1 for *active mode* and ε for *power saving mode*) and θ_0 is a constant value (load independent), representing the operation energy which includes baseband processing, radio frequency power expenditures, etc. The constant $\varepsilon \in (0, 1)$ accounts for the fact that the baseband energy consumption can be scaled down as well whenever there is no or little channel activity, into a power saving mode. $\theta_{\mathrm{load}}(t)$ represents the total wireless transmission (load dependent) power to meet the target transmission rate from the BS to the served user(s) and to guarantee low latency at the edge. Since we assume a noise-limited channel and the guarantee of low latency requirements at the edge, $\theta_{\mathrm{load}}(t)$ is obtained by using the transmission model in [44] (see (5) in this reference). Here, we neglect the imbalance of traffic volumes in uplink and downlink, and also we do not account for the switching energy cost for the BS mode transition [46] due to the fact that future BS functions will be virtualized [48].

MEC Server Energy Consumption. It depends on the number of VMs running in time slot t, named, $M(t) \leq M$, and on the CPU frequency that is allotted to each virtual machine. Specifically, VMs are instantiated on top of the physical CPU cores, and each VM is given a share of the host server CPU, memory, and network input/output interfaces. The CPU is the main consumer of energy in the server [33] due to the VM-to-CPU share mapping. Hence, in this work we focus on the CPU utilization only. With $f_m(t) \in [0, f_{\max}]$ we mean the instantaneous processing rate [49], expressed in bits per second that are computed, and $f_{\max}$ is the maximum processing rate for VM m. In this paper, $f_m(t)$ is set within a finite set $\mathcal{F} = \{f_0, f_1, \ldots, f_{\max}\}$ where $f_0 = 0$ represents zero speed of the VM (e.g., deep sleep or shutdown). At any given time t, the total energy consumption of a virtualized server, with $M(t)$ running VMs, is

$$\theta_{\mathrm{MEC},n}(t) = \sum_{m=1}^{M(t)} \left(\theta_m^{\mathrm{op}}(t) + \theta_m^{\mathrm{ov}}(t)\right), \quad (2)$$

where $\theta_m^{\mathrm{op}}(t)$ is the energy consumption of VM m operation and $\theta_m^{\mathrm{ov}}(t) \geq 0$ is the energy cost incurred through the turning on/off the VM; i.e., $\theta_m^{\mathrm{ov}}(t) > 0$ only when VM m is switched on/off and it is zero otherwise. $\theta_m^{\mathrm{op}}(t)$ is obtained using the linear relationship between the CPU utilization contributed by VM m and the energy consumption, from [49, 50] (see (4) in the second reference):

$$\theta_m^{\mathrm{op}}(t) = \theta_{\mathrm{idle},m}(t) + \alpha_m(t)\left(\theta_{\max,m}(t) - \theta_{\mathrm{idle},m}(t)\right), \quad (3)$$

where $\theta_{\mathrm{idle},m}(t)$ represents the *static* energy drained by VM m in the idle state, and $\theta_{\max,m}(t)$ is the *maximum* energy it drains. The quantity, $\alpha_m(t)(\theta_{\max,m}(t) - \theta_{\mathrm{idle},m}(t))$, represents the *dynamic* energy component, where $\alpha_m(t) = (f_m(t)/f_{\max})^2$ [8] is a load dependent factor. Note that $\alpha_m(t)$ and $f_m(t)$ are deterministically related as $f_{\max}$ is a constant. $\theta_m^{\mathrm{ov}}(t)$ is obtained from [50] (see (5) in this reference) as a constant and is typically limited to a few hundreds of mJ per MHz^2.

Conventionally, for each BS site, the hypervisor, i.e., the software that provides the environment in which the VMs operate, is in charge of allocating $f_m(t)$ and the workload fraction to be computed by the m-th VM, named $\gamma_m(t)$. In our setup, we have $\sum_{m=1}^{M(t)} \gamma_m(t) \leq \Gamma_n(t)$, where equality is achieved when the workload is fully served by $M(t)$ VMs. We also note that, in practical application scenarios, the maximum per-VM computation load to be computed is generally limited up to an assigned value, named $\gamma^{\max}$. Motivated by the energy efficient requirements from [39], i.e., the hypervisor's ability to accept and implement policies from a management entity, in this paper, the *edge controller* usage is pursued. Here, the edge controller determines the $f_m(t)$ value that will yield the desired or expected processing time, $\mu_m(t) = \gamma_m(t)/f_m(t)$, considering the workload $\gamma_m(t)$ allotted to VM m. $\mu_m(t)$ must be less than or equal to the maximum per-slot and per-VM processing time (in seconds), named, Δ; i.e., $\mu_m(t) \leq \Delta$. Note that Δ is also the server's response time, i.e., the maximum time allowed for processing the total computation load.

We remark that, as a result of the allocation procedure that is developed in this paper, for any BS site n, the processing rates $f_m(t)$ shall be found, similar to [50] (see Remark 1 from this reference). Then, the total amount of load that is served by the BS site may be set as follows: $B_n(t) = \sum_{m=1}^{M(t)} \gamma_m(t) \leq \Gamma_n(t)$. The objective of the considered optimization is to find the operating mode for the BS (either "on" or "power saving"), the number of VMs $M(t)$ that are to be allocated and, for each of them, the processing rate $f_m(t)$. In doing so, (1) the amount of delay sensitive load that is not served at the edge, $\Gamma_n(t) - \sum_{m=1}^{M(t)} \gamma_m(t)$, shall be minimized, while exploiting as much as possible the energy harvested from the solar panels, so that the mobile network will be energetically self-sufficient and (2) the load is computed in a time shorter than or equal to Δ. The details of the proposed optimization algorithm are provided in Section 4.

Data Transmission Energy Consumption. We assume that the intercommunication between the BS and the MEC server is bidirectional and symmetric. Hence, under steady-state operating conditions, for the communication site n, $\theta_{\mathrm{TX},n}(t)$ is obtained as $\theta_{\mathrm{TX},n}(t) = \theta_{\mathrm{idle}}(t) + \theta_{\mathrm{data}}(t)B_n(t)$ by using the

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

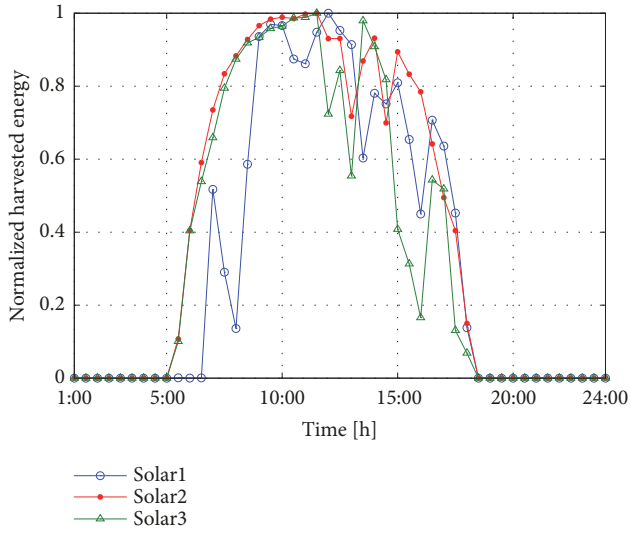


FIGURE 3: Example traces for harvested solar energy from [11].

VM migration hint from [51], where $\theta_{\text{idle}}(t)$ (fixed value in J) is the energy drained by the network interfaces in idle mode over a time slot t, θ_{data} (fixed value in J/byte) is the cost of exchanging one byte of data between the MEC server and the BS per time slot t, and $B_n(t)$ is the amount of data exchanged. These parameters, $\theta_{\text{idle}}(t)$ and $\theta_{\text{data}}(t)$, are obtained from [51]. Note that $B_n(t)$ also corresponds to the amount of data to be processed at the MEC server in bytes.

3.2. Energy Patterns and Storage. The energy buffer is characterized by its maximum energy storage capacity $\beta_{\max}$. At the *beginning* of each time slot t, the EM provides the energy level report to the edge controller through the local MEC server; thus the EB level $\beta_n(t)$ is known, enabling the provision of the required computation resources, i.e., the VMs. The energy level report/file from the EM to the MEC server is transferred using the pull mode procedure (e.g., File Transfer Protocol) [40].

In this work, the amount of harvested energy $H_n(t)$ in time slot t in the communication site n is obtained from open source solar traces [11] (see Figure 3). The dataset is the result of daily environmental records. In our numerical results, $H_n(t)$ represents a daily solar radiation record for three different areas. From the three solar profiles, each communication site energy profile is picked at random to represent the daily energy harvested and then scaled to fit the EB capacity $\beta_{\max}$ of 490 kJ. Thus, the available EB level $\beta_n(t+1)$ at the beginning of time slot $t+1$ is calculated as follows:

$$\beta_n(t+1) = \beta_n(t) + H_n(t) - \theta_{\text{tot},n}(t) + Q_n(t), \quad (4)$$

where $\beta_n(t)$ is the energy level in the battery at the beginning of time slot t, $\theta_{\text{tot},n}(t)$ is the energy consumption of the communication site over time slot t (see (1)), and $Q_n(t) \geq 0$ is the amount of energy purchased from the power grid. We remark that $\beta_n(t)$ is updated at the beginning of time slot t whereas $H_n(t)$ and $\theta_{\text{tot},n}(t)$ are only known at the end of it.

For decision-making in the edge controller, the received EB level reports are compared with the following thresholds: β_{low} and β_{up}, respectively termed the lower and the upper energy threshold with $0 < \beta_{\text{low}} < \beta_{\text{up}} < \beta_{\max}$. β_{up} corresponds to the desired energy buffer level at the BS and β_{low} is the lowest EB level that any BS should ever reach. If $\beta_n(t) < \beta_{\text{low}}$, then BS n is said to be *energy deficient* and our optimization in the following section makes sure that $\beta_n(t)$ never falls below β_{low} due to its transmission and computing activities within a time slot. Instead, if for any time slot we have $\beta_n(t) < \beta_{\text{up}}$, then the following amount of energy $Q_n(t) = \beta_{\text{up}} - \beta_n(t)$ is purchased from the energy grid to compensate for the deviation from the desired EB level (due to previous BS activity).

4. Optimization for a Single Communication Site

In this section, we formulate an optimization problem to obtain *energy savings* through short-term traffic load and harvested energy predictions, along with energy management procedures for a *single* communication site. The optimization problem is defined in Section 4.1, and the communication site management procedures are presented in Section 4.2.

4.1. Problem Formulation. At the beginning of each time slot t, the edge controller receives the energy level report $\beta_n(t)$ from each EM (via the MEC application responsible for energy profiles in the MEC server), using the pull mode file transfer. Here, we aim to minimize the overall energy consumption in the communication site over time, i.e., the consumption related to the BS transmission activity and the MEC server, by applying BS power saving modes and VM soft-scaling, i.e., tuning the number of active virtual machines. To achieve this, we first consider the optimization for a single communication site. We define two cost functions as follows:

(F1) $\theta_{\text{tot},n}(t)$, which weighs the energy consumption due to transmission (BS) and computation (MEC server);

(F2) a quadratic term $(\Gamma_n(t) - B_n(t))^2$, which accounts for the QoS cost.

In fact, (F1) tends to push the system towards self-sustainability solutions; i.e., $\zeta_n(t) \longrightarrow \varepsilon$. Instead, (F2) favors solutions where the delay sensitive load is entirely processed by the local MEC server; i.e., $B_n(t) \longrightarrow \Gamma_n(t)$. A weight $\eta \in [0,1]$ is utilized to balance the two objectives (F1) and (F2). The corresponding (weighted) cost function is defined as

$$J(\zeta, \alpha, t) \stackrel{\Delta}{=} \overline{\eta}\theta_{\text{tot},n}(\zeta_n(t), \{\alpha_m(t)\}, t) + \eta(\Gamma_n(t) - B_n(t))^2, \quad (5)$$

where $\overline{\eta} \stackrel{\Delta}{=} 1 - \eta$; with $\{\alpha_m(t)\}$ we mean the sequence of factors $\alpha_1(1), \alpha_2(1), \ldots, \alpha_{M(t)}(1)$. Hence, letting 1 be the current time

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

slot and T be the time horizon, the following optimization problem is formulated over time slots $1, \ldots, T$:

$$\begin{aligned} \textbf{P1}: \min_{\boldsymbol{\zeta}, \boldsymbol{\alpha}} \quad & \sum_{t=1}^{T} J(\zeta, \alpha, t) \\ \text{subject to:} \quad & \text{C1: } \zeta_n(t) \in \{\varepsilon, 1\}, \\ & \text{C2: } b \leq M(t) \leq M, \\ & \text{C3: } \beta_n(t) \geq \beta_{\text{low}}, \\ & \text{C4: } 0 \leq f_m(t) \leq f_{\max}, \\ & \text{C5: } 0 \leq \gamma_m(t) \leq \gamma^{\max}, \\ & \text{C6: } \mu_m(t) \leq \Delta, \quad t = 1, \ldots, T, \end{aligned} \tag{6}$$

where $m = 1, \ldots, M(t)$ (VM index) and vectors $\boldsymbol{\zeta}$ (BS switching status in time slots $1, \ldots, T$) and $\boldsymbol{\alpha}$ (load dependent factor) contain the *control actions* for the considered time horizon, per communication site; i.e., $\boldsymbol{\zeta} = [\zeta(1), \zeta(2), \ldots, \zeta(T)]$ and $\boldsymbol{\alpha} = [\{\alpha_m(1)\}, \{\alpha_m(2)\}, \ldots, \{\alpha_m(T)\}]$. Constraint C1 specifies the BS operation status (either *power saving* or *active*), C2 forces the required number of VMs, $M(t)$, to be always greater than or equal to a minimum number $b \geq 1$: the purpose of this is to be always able to handle mission critical communications. C3 makes sure that the EB level is always above or equal to a preset threshold β_{low}, to guarantee *energy self-sustainability* over time. Note that this constraint may imply that in certain time slots the BS is to be switched off, although the workload may be nonnegligible. When managing a single BS site (the formulation in this section), this implies that the load will not be served, but this fact may be compensated for when multiple communication sites are jointly managed, e.g., handing off the workload to another, energy richer, and BS. This is dealt with in Section 5. Furthermore, C4 and C5 bound the maximum processing rate and workloads of each running VM m, with $m = 1, \ldots, M(t)$, respectively. Constraint C6 represents a hard-limit on the corresponding per-slot and per-VM processing time.

To solve P1 in (6), we leverage the use of LLC [8, 30] and heuristics, obtaining the controls $\varsigma(t) \stackrel{\Delta}{=} (\zeta(t), \{\alpha(t)\})$ for $t = 1, \ldots, T$. Note that (6) can iteratively be solved at any time slot $t \geq 1$, by just redefining the time horizon as $t' = t, t+1, \ldots, t+T-1$.

4.2. Communication Site Management. In this subsection, a traffic load and energy harvesting prediction method and an online management algorithm are proposed to solve the previously stated problem P1. In Section 4.2.1, we discuss the prediction of the future (short-term) traffic load and harvested energy processes, and then in Section 4.2.2, we solve P1 by first constructing the state-space behavior of the control system, where online control key concepts are introduced. Finally, the algorithm for managing the single communication site is presented in Section 4.2.3.

Modeling steps
Step 1: load and normalize the dataset
Step 2: split dataset into training and testing
Step 3: reshape input to be [samples, time steps, features]
Step 4: create and fit the LSTM network
Step 5: make predictions
Step 6: calculate performance measure

Box 1: LSTM prediction model steps.

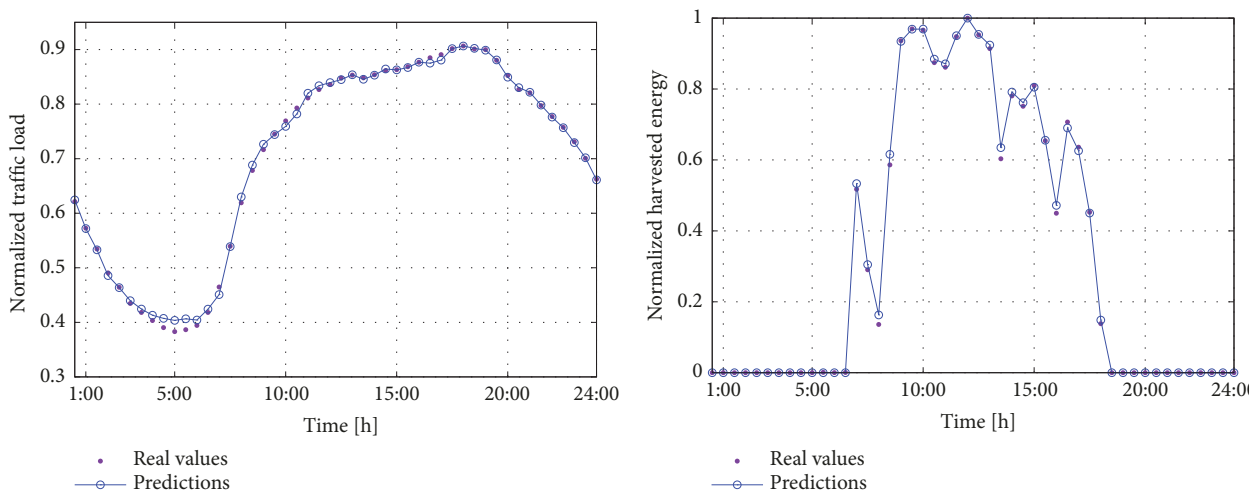
4.2.1. Traffic Load and Energy Forecasting. ML techniques constitute a promising solution for network management and energy savings in cellular networks [52, 53]. In this work, given a time slot duration of $\tau = 30$ min, we perform time series prediction; i.e., we obtain the $T = 3$ estimates of $\widehat{L}_n(t)$ and $\widehat{H}_n(t)$, by using an LSTM network developed in Python using Keras deep learning libraries (Sequential, Dense, LSTM) where the network has a visible layer with one input, one hidden layer of four LSTM blocks or neurons, and an output layer that makes a single value prediction. This type of recurrent neural network uses backpropagation through time for learning and memory blocks for regression [7]. The dataset is split as 67% for training and 33% for testing. The network is trained using 100 epochs (2,600 individual training trials) with batch size of one. As for the performance measure of the model, we use the Root Mean Square Error (RMSE). The prediction steps are outlined in Box 1. Figures 4(a) and 4(b) show the prediction results that will be discussed in Section 6.

4.2.2. Edge System Dynamics. We denote the system state vector at time t by $\mathbf{x}(t) = (M(t), \beta_n(t))$, which contains the number of active VMs, $M(t)$, and the EB level, $\beta_n(t)$, for the BS site n. $\boldsymbol{\varsigma}(t) = (\zeta(t), \{\alpha_m(t)\})$ is the input vector, i.e., the control action that drives the system behavior at time t. The system evolution is described through a discrete-time state-space equation, adopting the LLC principles [8, 30]:

$$\mathbf{x}(t+1) = \Phi(\mathbf{x}(t), \boldsymbol{\varsigma}(t)), \tag{7}$$

where $\Phi(\cdot)$ is a behavior model that captures the relationship between $(\mathbf{x}(t), \boldsymbol{\varsigma}(t))$, and the next state $\mathbf{x}(t+1)$. Note that this relationship accounts for (1) the amount of energy drained $\theta_{\text{tot},n}(t)$ that harvested $H_n(t)$ and that purchased from the power grid $Q_n(t)$, which together lead to the next buffer level $\beta_n(t+1)$ through (4) and (2) to the traffic load $L_n(t)$, from which we compute the server workloads $\Gamma_n(t)$ that leads to $M(t)$ and to the control $\boldsymbol{\varsigma}(t)$. The network management algorithm in the edge controller, the ENAAM algorithm, finds the best control action vector for the communication site, following a *model predictive control approach*. Specifically, for each time slot t, problem (6) is solved, obtaining control actions for the whole time horizon $t, t+1, \ldots, t+T-1$. The control action that is applied at time t is $\boldsymbol{\varsigma}^*(t)$, which is the first one in the retrieved control sequence. This control amounts to setting the BS radio mode according to $\zeta^*(t)$, i.e., either active or power saving, and the number of instantiated VMs,

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) One-step ahead predictive mean value for $L(t)$

(b) One-step ahead predictive mean value for $H(t)$

FIGURE 4: One-step online forecasting for both $L(t)$ and $H(t)$ patterns.

$M^*(t)$, along with their obtained $\{\alpha_m^*(t)\}$ values (see Remarks 1 and 2 below). This is repeated for the following time slots $t+1, t+2, \ldots$.

Remark 1 (role of prediction). State $\mathbf{x}(t)$ and control $\boldsymbol{\varsigma}(t)$ are, respectively, measured and applied at the beginning of time slot t, whereas the offered load $L_n(t)$ and the harvested energy $H_n(t)$ are accumulated during the time slot and their value becomes known only by the end of it. This means that, being at the beginning of time slot t, the system state at the next time slot $t+1$ can only be estimated, which we formally write as

$$\widehat{\mathbf{x}}(t+1) = \Phi(\mathbf{x}(t), \boldsymbol{\varsigma}(t)), \tag{8}$$

and the same applies to the subsequent time slots in the optimization horizon $t+2, t+3, \ldots, t+T-1$. For these estimations we use the forecast values of load $\widehat{L}_n(t)$ and harvested energy $\widehat{H}_n(t)$, from the LSTM forecasting module.

Remark 2 (VM number and workload allocation). A remark on the provisioned VMs per time slot per-MEC server, $M(t)$, is in order. Specifically, the number of active VMs (i.e., the VM computing cluster) depends on the predicted load, $\widehat{L}_n(t+1)$, where the expected server workload is $\widehat{\Gamma}_n(t+1) = 0.8\widehat{L}_n(t+1)$. Each VM can compute an amount of up to $\gamma^{\max}$. Then, an estimate of the number of virtual machines that shall be active in time slot t to serve the predicted server workloads is here obtained as follows: $M(t) = \lceil(\widehat{\Gamma}_n(t+1)/\gamma^{\max})\rceil$, where $\lceil\cdot\rceil$ returns the nearest upper integer. We heuristically split the workload among virtual machines by allocating a workload $\gamma_m(t) = \gamma^{\max}$ to the first $M(t)-1$ VMs, $m = 1, \ldots, M(t)-1$, and the remaining workload $\gamma_m(t) = \widehat{L}_n(t+1) - (M(t)-1)\gamma^{\max}$ to the last one $m = M(t)$.

Controller Decision-Making. The controller is obtained by estimating the relevant parameters of the operating environment, i.e., the BS load $\widehat{L}_n(t)$ and the harvested energy $\widehat{H}_n(t)$, and subsequently using them to forecast the future system behavior through (8) over a look-ahead time horizon of T time slots. The control actions are picked by minimizing $J(\zeta, \alpha, t)$ (see (5)). At the beginning of each time slot t the following process is iterated:

(1) Future system states, $\widehat{\mathbf{x}}(t+k)$, for a prediction horizon of $k = 1, \ldots, T$ steps are estimated using (8). These predictions depend on past inputs and outputs up to time t, on the estimated load $\widehat{L}_n(\cdot)$ and energy harvesting $\widehat{H}_n(\cdot)$ processes, and on the control $\boldsymbol{\varsigma}(t+k)$, with $k = 0, \ldots, T-1$.

(2) The sequence of controls $\{\boldsymbol{\varsigma}(t+k)\}_{k=0}^{T-1}$ is obtained for each step of the prediction horizon by optimizing the weighted cost function $J(\cdot)$ (see (5)).

(3) The control $\boldsymbol{\varsigma}^*(t)$ corresponding to the first control action in the sequence with the minimum total cost is the applied control for time t and the other controls $\boldsymbol{\varsigma}^*(t+k)$ with $k = 1, \ldots, T-1$ are discarded.

(4) At the beginning of the next time slot $t+1$, the system state $\mathbf{x}(t+1)$ becomes known and the previous steps are repeated.

4.2.3. The ENAAM Algorithm. Let t be the current time. $\widehat{L}_n(t+k-1)$ is the forecast load in slot $t+k-1$, with $k = 1, \ldots, T$, i.e., over the prediction horizon. For the control to be feasible, we need $\underline{\Gamma}_n(t) \le B_n(t) \le \widehat{\Gamma}_n(t+k-1)$, where $\underline{\Gamma}_n(t)$ is the smallest Γ such that $\texttt{round}\,(\widehat{\underline{\Gamma}}_n(t+1)/\gamma^{\max}) = b$. For the buffer state, we heuristically set $\zeta(t+k-1) = \varepsilon$ if either $\beta_n(t+k-1) < \beta_{\text{low}}$ or $L_n(t+k-1) < L_{\text{low}}$, and $\zeta(t+k-1) = 1$; otherwise β_{low} and L_{low} are preset low thresholds for the EB and the BS load, respectively. For slot $t+k-1$, the feasibility

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Input: $\mathbf{x}(t)$ (current state)
Output: $\boldsymbol{\varsigma}^*(t) = (\zeta^*(t), \{\alpha_m^*(t)\})$
01: Initialization of variables
$\mathcal{S}(t) = \{\mathbf{x}(t)\}$, $\text{Cost}(\mathbf{x}(t)) = 0$
02: **for** $k = 1, \ldots, T$ **do**
(i) forecast the load $\widehat{L}_n(t+k-1)$
(ii) forecast the harvested energy $\widehat{\text{H}}_n(t+k-1)$
(iii) $\mathcal{S}(t+k) = \emptyset$
03: **for all** $\mathbf{x} \in \mathcal{S}(t+k-1)$ **do**
04: **for all** $\boldsymbol{\varsigma} = (\zeta, \{\alpha_m(t)\}) \in \mathcal{A}(t+k-1)$ **do**
05: $\widehat{\mathbf{x}}(t+k) = \boldsymbol{\Phi}(\mathbf{x}(t+k-1), \boldsymbol{\varsigma})$
06: $\text{Cost}(\widehat{\mathbf{x}}(t+k)) = J(\zeta, \alpha, t+k-1)$
$+\text{Cost}(\mathbf{x}(t+k-1), \boldsymbol{\varsigma})$
07: $\mathcal{S}(t+k) = \mathcal{S}(t+k) \cup \{\widehat{\mathbf{x}}(t+k)\}$
end for
end for
end for
08: **Find** $\widehat{\mathbf{x}}_{\min} = \text{argmin}_{\widehat{\mathbf{x}} \in \mathcal{S}(t+T)} \text{Cost}(\widehat{\mathbf{x}})$
09: $\boldsymbol{\varsigma}^*(t)$:= control leading from $\mathbf{x}(t)$ to $\widehat{\mathbf{x}}_{\min}$
10: **Return** $\boldsymbol{\varsigma}^*(t)$

Algorithm 1: ENAAM.

set $\mathcal{A}(t+k-1)$ contains the control pairs $(\zeta(t), \{\alpha_m(t)\})$ that obey these relations.

The algorithm is specified in Algorithm 1 as it uses the technique in [8]: the search starts (line 01) from the system state at time t, $\mathbf{x}(t)$, and continues in a breadth-first fashion, building a tree of all possible future states up to the prediction depth T. A cost is initialized to zero (line 01) and is accumulated as the algorithm travels through the tree (line 06), accounting for predictions, past outputs, and controls. The set of states reached at every prediction depth $t+k$ is referred to as $\mathcal{S}(t+k)$. For every prediction depth $t+k$, the search continues from the set of states $\mathcal{S}(t+k-1)$ reached at the previous step $t+k-1$ (line 03), exploring all feasible controls (line 04), obtaining the next system state from (8) (line 05), updating the accumulated cost as the result of the previous accumulated cost, plus the cost associated with the current step (line 06), and updating the set of states reached at step $t+k$ (line 07). When the exploration finishes, the initial action (at time t) that leads to the best final accumulated cost, at time $t+T-1$, is selected as the optimal control $\boldsymbol{\varsigma}^*(t)$ (lines 08, 09, 10). Finally, for line 04, we note that Γ_n belongs to the continuous set $[\underline{\Gamma}_n, \widehat{L}_n(t+k-1)]$. To implement this search, we quantized this interval into a number of equally spaced points, obtaining a search over a finite set of controls.

ENAAM Complexity. The computation complexity of the algorithm is $O(N_x N_\varsigma T)$, where $N_x \triangleq |\mathbf{x}(t)|$ and $N_\varsigma \triangleq |\boldsymbol{\varsigma}(t)|$, respectively, represent the number of system states and the number of feasible actions at time t. Note that state and action space are, respectively, quantized into $N_x = M \times N_\beta$ and $N_\varsigma = 2 \times M \times N_\alpha$ levels, where M is the number of virtual machines, N_β is the number of quantization levels for the energy buffer, and N_α is the number of quantization levels for the load variable $\alpha_m(t)$. Such quantization facilitates the search in Algorithm 1. Note that exhaustive search would entail a complexity of $O((N_x N_\varsigma)^T)$.

5. Multiple Communication Sites

In this section, we extend the work from Section 4 by considering the energy savings for *multiple* communication sites. We formulate an optimization problem to obtain energy savings through short-term traffic load and harvested energy predictions and clustering, along with energy management procedures for the clustered BS sites. The problem formulation for multiple communication sites is described in Section 5.1; then cluster formation is discussed in Section 5.2, and the edge management procedure for each cluster, enabled by the edge controller, is presented in Section 5.3.

5.1. Problem Formulation. Our objective is to improve the overall energy savings of the network by clustering BSs based on their location (or distance measures) similarity and then optimizing the energy savings within each cluster by employing the single optimization case described in Section 4. From an energy efficiency perspective, in a cluster of BS nodes, one BS (or more) might have a preference of switching off, by first offloading its (their) traffic load to its (their) neighboring BS that have enough spare capacity for handling extra traffic load and then switching off. The whole offloaded traffic load from the BS, denoted by BS n, is allocated to the neighboring cluster member (active BS) in which orthogonal resource allocation helps mitigate intracluster interference, such that the selected neighboring BS, denoted by BS n', is allocated the incremental load, denoted by $L_{nn'}(t) \triangleq L_n(t)$. Whenever a BS is switched off, it should maintain service to its users via a reassociation process in order to offload the users to the neighboring active BS having extra resources for handling upcoming extra traffic load. The reassociation process involves notifying the connected users to try and connect to neighboring BSs with extra resources.

In the view of the above, we consider that all BSs are grouped into sets of clusters $\mathcal{O} = \{O_1, \ldots, O_{|\mathcal{O}|}\}$. Here, a given cluster $O_i \in \mathcal{O}$, with $i = 1, \ldots, |\mathcal{O}|$, consists of a set of BSs that coordinate with the controller. The clustering mechanism is discussed in Section 5.2. For each cluster $O_i \in \mathcal{O}$, we aim to minimize the energy consumption, i.e., the consumption due to BS transmission and the running VMs in the servers, using BS power saving modes and VM soft-scaling per active cluster member. To do so, we define a cost function which captures the individual communication site energy consumption and its QoS. The (weighted) cost for each cluster member, BS $n \in O_i$, is redefined as follows:

$$J_n(\zeta, \alpha, t) \triangleq \overline{\eta}\theta_{\text{tot},n}\left(\zeta_n(t), \{\alpha_m(t)\}_n, t\right) + \eta\left(\Lambda_n(t) - B_n(t)\right)^2, \tag{9}$$

where $\zeta_n(t)$ is the activity status of BS n (either *power saving* or *active*) and $\{\alpha_m(t)\}_n$ is the set of factors for the allocated VMs at BS n. Moreover, $\Lambda_n(t) \longleftarrow L_n(t)$ if BS n only handles its own traffic, whereas $\Lambda_n(t) \longleftarrow L_n(t) + \Delta L_n(t)$, in case one

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

(or multiple) BSs are switched off in time slot t and its (their) traffic is redirected (handed off) to BS n. The computation of $\Delta L_n(t)$ is addressed in Section 5.3. The per cluster cost $\Upsilon_{O_i}(\boldsymbol{\zeta}_i, \boldsymbol{\alpha}_i, t)$ is the aggregated cost of all cluster members, $\Upsilon_{O_i}(\boldsymbol{\zeta}_i, \boldsymbol{\alpha}_i, t) = \sum_{\forall n \in O_i} J_n(\zeta, \alpha, t)$. Hence, over time horizon, $t = 1, \ldots, T$, the following optimization problem is defined:

$$\begin{aligned} \textbf{P2}: \min_{\mathscr{C}} \quad & \sum_{\forall O_i \in \mathcal{O}} \Upsilon_{O_i}(\boldsymbol{\zeta}_i, \boldsymbol{\alpha}_i, t) \\ \text{subject to:} \quad & \text{C1} - C6\text{: from Eq. (6)}, \\ & \text{C7: } |O_i| \geq 1, \quad \forall O_i \in \mathcal{O}, \\ & \text{C8: } O_i \cap O_j = \emptyset, \\ & \qquad \forall O_i, O_j \in \mathcal{O},\ O_i \neq O_j, \end{aligned} \tag{10}$$

where $\mathscr{C} \triangleq \{\boldsymbol{\zeta}_i, \boldsymbol{\alpha}_i\}$ is the collection of variables to be reconfigured for all the BS clusters (the whole MN), for all time slots $t = 1, \ldots, T$. As for the constraints, C7 and C8 ensure that each BS is part of only one cluster. Solving **P2** in (10) involves BS clustering, the forecasting method from Section 4.2.1, a heuristic rule for the selection of which BSs have to be switched off, and the ENAAM algorithm from Section 4.2.3. Once **P2** is solved, the control action to be applied at time t, per cluster O_i, corresponds to the elements in $\{\boldsymbol{\zeta}_i, \boldsymbol{\alpha}_i\}$ that are associated with the first time slot 1 in the optimization horizon. As above, (10) can iteratively be solved at any time slot $t \geq 1$, by just redefining the time horizon as $t' = t, t+1, \ldots, t+T-1$.

5.2. Cluster Formation. Clustering algorithms have been proposed as a way of enabling energy saving mechanisms in BSs, where groups of inactive BSs or BSs with low loads are switched off. With the advent of EH BSs, the BSs with $\beta_n(t) < \beta_{\text{low}}$ can be switched off, while still guaranteeing the QoS through the other active BSs. That is, within each formed cluster, the controller tries to minimize the cost function, which captures the tradeoff between the energy efficiency and the QoS of each cluster member. The key step in clustering is to identify similarities or distance measures between BSs in order to group BSs with similar characteristics. In this paper, we use the location of the BSs as it defines the relative neighborhood (the distance measures) with the other BSs. Using the location of the BSs and the distance between the BSs, we obtain a distance-based similarity matrix $\mathbf{W}^d$. In addition, we assume that the network topology is static during the clustering algorithm execution.

In Section 5.2.1 we detail the clustering measure that we use to obtain the similarities between BSs based on location, followed by the distance-based clustering algorithm in Section 5.2.2.

5.2.1. Relative Neighborhood Based on BS Adjacency and Gaussian Similarity. Similar to [13], we model the MN as a graph $G = (\mathcal{N}, E)$, where $\mathcal{N}$ represents the set of BSs, while the set E contains the edges between any two BSs. There is an edge $(n, n') \in E$ if and only if n and n' can mutually receive each other's transmission. In this case, we say that n and n' are neighbors. We use a parameter $r_{nn'}$ to characterize the presence of a link between nodes, where $r_{nn'} \in \{0, 1\}$. Let y_n be the coordinates of BS $n \in \mathcal{N}$ in the Euclidean space. The relative neighborhood of BS n is defined by the nearness of the BSs in its e_d-radio propagation space (or neighborhood):

$$\mathcal{Z}_n = \left\{ n' \text{s.t.} \left\| y_n - y_{n'} \right\| \leq e_d \right\}. \tag{11}$$

If $n' \in \mathcal{Z}_n$ we say that BSs n and n' are neighbors, and we set $r_{nn'} = 1$; otherwise $r_{nn'} = 0$. The links between the vertices in $\mathcal{N}$ are weighted based on their similarities. Based on the distance between BS n and n', we can classify the BSs based on their location using the Gaussian similarity measure [13] (a classification kernel function used in machine learning), which is defined as

$$w^d_{nn'} = \begin{cases} \exp\left(\dfrac{-\left\| y_n - y_{n'} \right\|^2}{2\sigma_d^2} \right) & \text{if } \left\| y_n - y_{n'} \right\| \leq e_d, \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where $2\sigma_d^2$ adjusts the impact of the neighborhood size. In (12), we assume that the BSs located far from each other have low similarities, compared to those that are close to each other, as those that are close are more likely to cooperate with each other. The distance-based similarity matrix $\mathbf{W}^d$ is formed using $w^d_{nn'}$ as the (n, n')-th entry.

5.2.2. Distance-Based Clustering. The BS clustering is performed after obtaining the similarity matrix $\mathbf{W}^d$ of the MN graph $G = (\mathcal{N}, E)$. Given the matrix $\mathbf{W}^d$, we employ a centralized clustering method, specifically the K-means [54], as the matrix provides the full location knowledge. K-means partitions the set of nodes into clusters in which each node belongs to the cluster with the nearest mean distance. In addition, the value of K, i.e., the number of clusters ($|O_i|$), is known prior and is a design parameter. This algorithm requires knowledge of all the BS locations; thus, it is categorized as a centralized method. In our case, this process does not incur any computation delay as the edge controller is assumed to have high computation capabilities.

5.3. Edge Network Management. Our aim is to implement and validate an LLC framework for dynamic resource provisioning in multiple communication sites with the goal of achieving energy savings within the access network through BS sleep modes and VM soft-scaling. Given the formation of clusters, load, and energy forecasting, our next goal is to develop a mechanism for solving **P2** (see (10)) where each cluster of BSs adjusts its transmission parameters and its computing cluster entities based on the forecast information. In order to minimize the per cluster cost function, we introduce the notion of *network impact* in Section 5.3.1, whereas we describe the edge management procedure in Section 5.3.2.

5.3.1. Network Impact. The dynamic BS switching off strategies may have an impact on the network due to the traffic

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

load that is offloaded to the neighboring BSs. To avoid this, the BS to be switched off must be carefully identified within a BS cluster. To determine whether a particular BS can be switched off or not, we follow the work done in [55]. As an example, we consider one cluster O_i, together with its cluster members $n \in O_i$, then from it we choose one BS, BS n, where BS n neighbors set is denoted by $\mathcal{N}_n$. Note that the BS $n' \in \mathcal{N}_n$ is the BS to which the traffic load will be offloaded to after turning off BS n. Also, BS n can only be switched off if there exists a neighboring BS n' that satisfies the following feasibility constraint [55]:

$$L_{n'}(t) + L_{nn'}(t) \leq 1, \quad n' \in \mathcal{N}_n, \tag{13}$$

where $L_{n'}(t)$ is the original BS n' traffic load and $L_{nn'}(t)$ is the incremental traffic load from BS n (the switched off BS) to BS n' (the neighboring BS). We recall that the load $L_{n'}(t)$ is normalized with respect to the maximum load that a BS can sustain, so the inequality in (13) means that it is feasible for BS n' to take the extra load from BS n. To quantify how the incremental system load affects the overall network load due to the switching off process, we introduce the notion of *network impact*. For every BS n within cluster O_i, $i = 1, \ldots, K$, its *network impact* due to the offloaded system load onto one of the neighboring BSs is defined as follows:

$$I_n(t) = \max_{n' \in \mathcal{N}_n} \left[L_{n'}(t) + L_{nn'}(t) \right], \quad \forall n \in O_i. \tag{14}$$

Here, the maximum network impact value $I_n(t)$ over the neighboring BSs is considered as a measure for each BS towards switching off and generating extra traffic loads for its neighboring BSs. In this work, considering cluster O_i, we switch off the BS n^* that has the least network impact; i.e.,

$$n^* = \arg\min_{n \in O_i} I_n(t). \tag{15}$$

The BS that takes the load from n^* is selected as the BS n' that minimizes $L_{n'}(t) + L_{n^*n'}(t)$ over the set of active BSs that are on within the cluster O_i. For BS n', we then set $L_{n'}(t) \longleftarrow L_{n'}(t) + L_{n^*n'}(t)$. This procedure is sequentially repeated for all the cluster members until there is no active BS whose neighbors satisfy the feasibility condition of (13). Note that here, we focus only on which BS to switch off, as for the BS turning on state, we assume that the *commitment time* (time configured so that the BS automatically wakes up without external triggers) is a system parameter that is preconfigured when the BS is switched off.

5.3.2. Edge Management Procedure. Here, we propose a distributed edge network management procedure that makes use of the ENAAM algorithm (see Section 4.2.3). The decision-making criterion only depends on the BS information and on its neighboring BSs; thus, the BS switching off decision can be localized within each cluster. To decide which BSs shall be switched off, we follow a sequential decision process. While this is heuristic, it allows coping with the high complexity associated with an optimal (all BSs are jointly assessed) allocation approach. The edge management procedure is as follows.

For each BS cluster O_i, with $i = 1, \ldots, K$, we have the following:

(1) Initialize an allocation variable $\Delta L_n(t) = 0$ for all BSs $n \in O_i$. Compute $I_n(t)$, using (14), for all BSs n and obtain the BS with the least *network impact* $n^*(t)$, using (15). Switch off BS $n^*(t)$ and assign its load to the neighboring BS $n' \in O_i$ that minimizes $L_{n'}(t) + \Delta L_{n'}(t) + L_{n^*n'}(t)$. Update the extra allocation for BS n' as $\Delta L_{n'}(t) \longleftarrow \Delta L_{n'}(t) + L_{n^*n'}(t)$. Recompute $I_n(t)$ for all the BSs that are still on and identify the next BS that can be switched off, i.e., the one with the *least network impact*. This procedure is repeated until none of the BSs in the cluster verifies Eq. (13). At this point, we have identified all the BSs n^* that shall be switched off in O_i.

(2) For each active BS $n' \in O_i$, the ENAAM algorithm is executed using $L_{n'}(t) + \Delta L_{n'}(t)$, where $\Delta L_{n'}(t) = 0$ if BS n' does not take extra load, whereas it is greater than zero otherwise. Note that, $\Delta L_{n'}(t)$ corresponds to the total traffic that is handed over to BS n', possibly from multiple nearby BSs.

Edge Network Management Complexity. The algorithm is independently executed for each cluster and the corresponding time complexity is obtained as follows. Considering the action *Step (1)*, from above, the time complexity associated with the computation of the BS having the least network impact is linear with the size of the cluster $|O_i|$. Once that is computed, the complexity associated with updating the load allocation for the active BSs is $|O_i| - 1$, which leads to a total complexity of $|O_i|(|O_i| - 1) = O(|O_i|^2)$. Moreover, such process is iterated for each BS that is switched off. In the worst case, where all the BSs but one are switched off, the final complexity of step 1 is $O(|O_i|^3)$. As for *Step (2)*, from above, the computation complexity depends on the ENAAM algorithm, which is independently executed by each *active* BS. Thus, in the worst case (no BSs are switched off), the total aggregated complexity is as follows: $O(|O_i| N_x N_\varsigma T)$, which is linear in all variables, namely, number of cluster members, number of BS states, number of actions, and time horizon T.

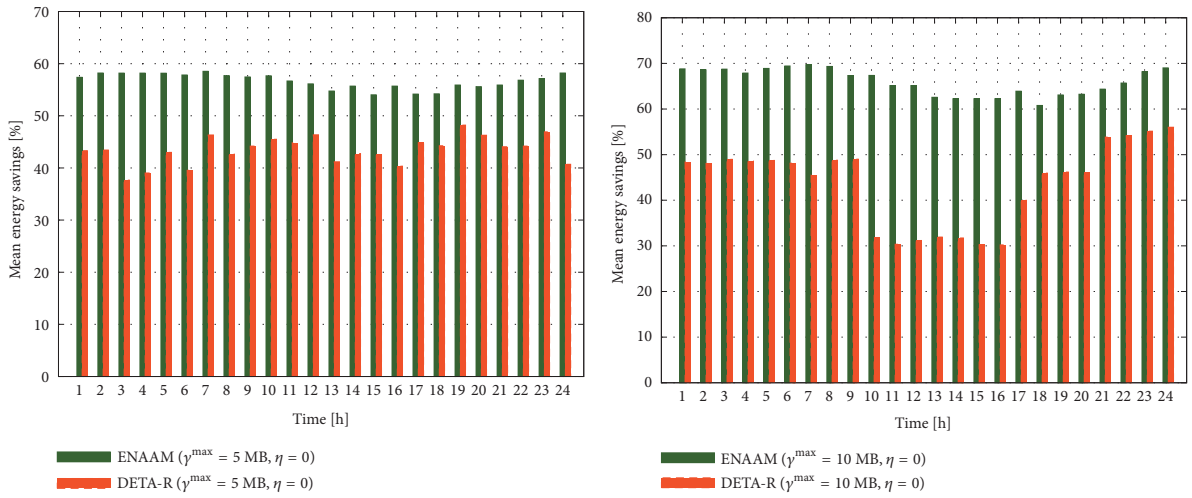
6. Performance Evaluation

In this section, we show some selected numerical results for the scenario of Section 3. The parameters that were used for the simulations are listed in Table 2.

6.1. Simulation Setup. We consider multiple BSs, each one colocated with a MEC server and a coverage radius of 40 m. In addition, we use a virtualized server with specifications from [56] for a VMware ESXi 5.1-ProLiant DL380 Gen8. Our time slot duration τ is set to 30 min and the time horizon is set to $T = 3$ time slots. The simulations are carried out by exploiting the Python programming language.

6.2. Numerical Results. Pattern Forecasting. We show real and predicted values for the traffic load and harvested energy over time in Figures 4(a) and 4(b), where we track the

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) Mean energy savings for $\eta = 0$ and $\gamma^{max} = 5$ MB

(b) Mean energy savings for $\eta = 0$ and $\gamma^{max} = 10$ MB

Figure 5: Mean energy savings for the single BS case.

Table 2: System parameters.

Parameter	Value
Total BSs, N	24
Max. number of VMs, M	27
Min. number of VMs, b	1
Time slot duration, τ	30 min
Operating power, θ_0	10.6 W
Energy overheads for switching VM, $\theta_m^{ov}(t)$	0.05 J/MHz^2
Max. computation workload per VM, γ^{max}	{5, 10} MB
Max. allowed processing time, Δ	0.8 s
Energy cons. of network interfaces, $\theta_{idle}(t)$	3 J
Cost of exchanging one unit of data, $\theta_{data}(t)$	6 J/byte
Processing rate set, $\mathcal{F}$	{0, 4, 8, 12, 16, 20}
Static energy consumed by VM, $\theta_{idle,m}(t)$	4 J
Max. energy cons. by VM at f_{max}, $\theta_{max,m}(t)$	10 J
Energy storage capacity, β_{max}	490 kJ
Lower energy threshold, β_{low}	30% of β_{max}
Upper energy threshold, β_{up}	70% of β_{max}
Low traffic threshold, L_{low}	4 MB

one-step predictive mean value at each step of the online forecasting routine. Then, Table 3 shows the average RMSE of the normalized harvested energy and traffic load processes, for different time horizon values, $T \in \{1, 2, 3\}$. Note that the predictions for $H(t)$ are more accurate than those of $L(t)$ (confirmed by comparing the average RMSE), due to differences in the used dataset granularity. However, the measured accuracy is deemed good enough for the proposed optimization.

Single Communication Site. Figures 5(a) and 5(b) are computed with $\eta = 0$ using Cluster 1 and Solar 1 as traffic load and harvested energy profiles for each BS (see Figures 2 and 3). Moreover, $\gamma^{max} = 5$ MB and 10 MB, respectively.

Table 3: Average prediction error (RMSE) for harvested energy and traffic load processes, both normalized in [0, 1].

	$T = 1$	$T = 2$	$T = 3$
$L(t)$	0.037	0.042	0.048
$H(t)$	0.011	0.016	0.021

They show the mean energy savings achieved over time when on-demand and energy aware edge resource provisioning are enabled (i.e., BS sleep modes and VM soft-scaling), in comparison with the case where they are not applied. Our edge network management algorithm (ENAAM) is benchmarked with another one that heuristically selects the amount of traffic that is to be processed locally, $B_n(t) \leq \Gamma_n(t)$, depending on the expected load behavior. It is named Dynamic and Energy-Traffic-Aware algorithm with Random behavior (DETA-R). Both ENAAM and DETA-R are aware of the predictions in future time slots (see Section 4.2.1); however, DETA-R provisions edge resources using a heuristic scheme. DETA-R heuristic works as follows: if the expected load difference is $\widehat{L}(t+1) - \widehat{L}(t) > 0$, then the normalize workload to be processed by BS n in the current time slot t, $B_n(t)$, is randomly selected in the range [0.6, 1]; otherwise, it is picked evenly at random in the range (0, 0.6).

Average results for the ENAAM scheme show energy savings of 69% ($\gamma^{max} = 10$ MB) and 57% ($\gamma^{max} = 5$ MB), while DETA-R achieves 49% ($\gamma^{max} = 10$ MB) and 43% ($\gamma^{max} = 5$ MB) on average, where these savings are with respect to the case where *no energy management* is performed; i.e., the network is dimensioned for maximum expected capacity (maximum value of $\theta_{tot,n}(t)$, with $M = 27$ VMs, $\forall t$). The results show that the maximum load allocated to each VM, γ^{max}, has an impact towards energy savings. An increase in energy savings is observed when $\gamma^{max} = 10$ MB due to the fact that the number of VMs demanded per time slot is reduced, when compared to the allocation of $\gamma^{max} = 5$ MB.

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

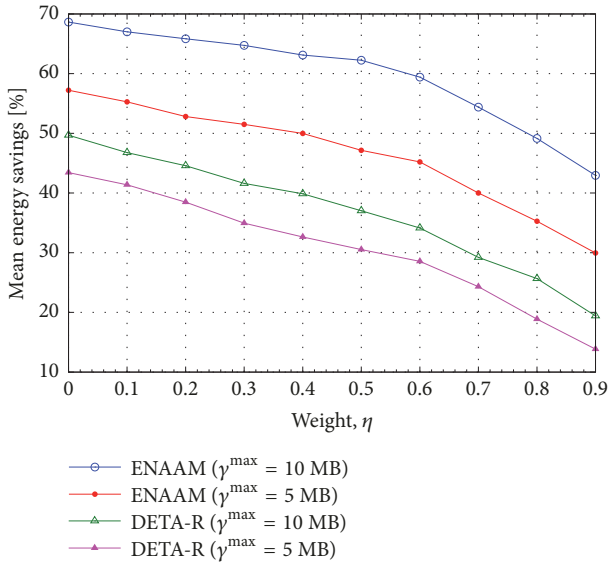


FIGURE 6: Energy savings versus weight η (single BS case).

The ESs evolution with respect to η is presented in Figure 6, taking into account the load allocated to each VM, $\gamma^{\max}$. The results were obtained using Cluster 1 and Solar 1 as traffic load and harvested energy profiles (*see* Figures 2 and 3). As expected, a drop in energy savings is observed when QoS is prioritized, i.e., $\eta \longrightarrow 1$, as in this case the BS energy consumption is no longer considered. It can be observed that ENAAM achieves a 50% (or above) from $\eta = [0, 0.4]$ when $\gamma^{\max} = 5$ MB and from $\eta = [0, 0.7]$ when $\gamma^{\max} = 10$ MB. This shows that the higher the load allocated to each VM, the lesser the energy that is drained, as few VMs are running. DETA-R operates at below 50% for all η and $\gamma^{\max}$ values.

Multiple Communication Sites. Figures 7(a) and 7(b) present the mean energy savings achieved with respect to the cluster size and the weight η, using all the traffic load and harvested energy profiles from Figures 2 and 3. Each BS randomly picks its own traffic load and harvested energy profile at the beginning of the optimization process. Here, to select the BS to be switched off, we use the management procedure of Section 5.3. As for DETA-R, a BS is randomly selected to evolve its operating mode to power saving mode and offload its load to a nearby BS (in this case, the least loaded neighboring BS is selected), without taking into account its network impact measure.

Figure 7(a) shows the average energy savings obtained when clustering is adopted, i.e., here, the cluster size is increased from $|O_i| = 1$ to 10 and $\eta = 0$. The obtained energy savings are with respect to the case where all BSs are dimensioned for maximum expected capacity (maximum value of $\theta_{\text{tot},n}(t)$, with $M = 27$ VMs, $\forall t, \forall n \in O_i$). It should be noted that the energy savings increase as the size of the cluster grows, thanks to the load balancing among active BSs, which cannot be implemented in the single communication site scenario (i.e., when BSs are independently managed).

Then, Figure 7(b) shows the average energy savings with respect to η, when the cluster size is set to an intermediate case ($|O_i| = 6$). Again, here the energy savings are obtained with respect to the case where all the BSs are dimensioned for maximum capacity. As expected, there is a drop in the energy savings achieved as the value of η increases, as QoS is prioritized. It can be observed that ENAAM achieves a value of 50% or above when $\eta = [0, 0.8]$ (at $\gamma^{\max} = 10$ MB) and when $\eta = [0, 0.6]$ (at $\gamma^{\max} = 5$ MB). DETA-R achieves value above 50% or above when $\eta = [0, 0.4]$ (at $\gamma^{\max} = 10$) and $\eta = [0, 0.1]$ (at $\gamma^{\max} = 5$ MB).

Comparing Figures 6 and 7(b), an average gain of 9% on the energy savings is observed when clustering is applied, by considering the mean energy savings with respect η achieved with ENAAM for both cases. From Figure 7(a) we see that this gain can be as high as 16% for ENAAM with $\gamma^{\max} = 5$ MB (red curve) and bigger for the DETA-R approach. These results support the notion that performing a clustering-based optimization is beneficial thanks to the additional cooperation within each neighborhood of BSs. This cooperation allows switching off more BSs through load balancing, increasing the energy savings while still controlling the users' QoS.

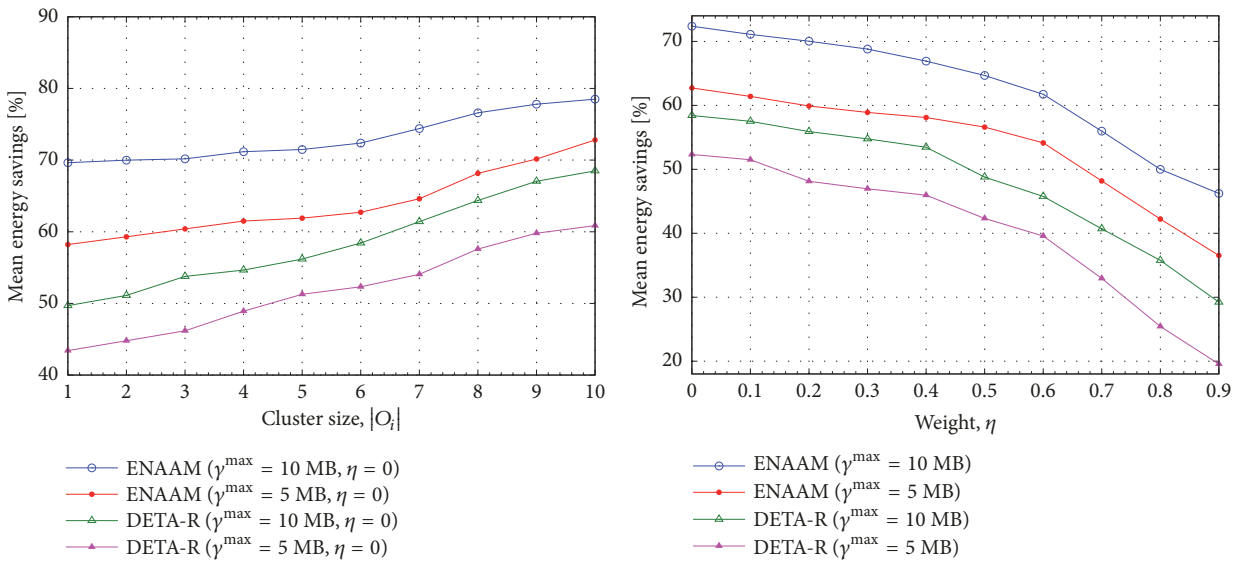
7. Conclusions

In this paper, we have envisioned an edge network where a group of BSs are managed by a controller, for ease of BS organization and management, and also a mobile network where the edge apparatuses are powered by hybrid supplies, i.e., using green energy in order to promote energy self-sustainability and the power grid as a backup. Within the edge, each BS is endowed with computation capabilities to guarantee low latency to mobile users, offloading their workloads locally. The combination of energy saving methods, namely, BS sleep modes and VM soft-scaling, for single and multiple BS sites helps to reduce the mobile network's energy consumption. An edge energy management algorithm based on forecasting, clustering, control theory and heuristics, is proposed with the objective of saving energy within the access network, possibly making the BS system self-sustainable. Numerical results, obtained with real-world energy and traffic load traces, demonstrate that the proposed algorithm achieves energy savings between 57% and 69%, on average, for the single communication site case, and a gain ranging from 9% to 16% on energy savings is observed when clustering is applied, with respect to the allocated maximum per-VM loads of 5 MB and 10 MB. The energy saving results are obtained with respect to the case where no energy management techniques are applied, either in one BS or single cluster.

Data Availability

In this paper, we have used open source datasets for the mobile network (MN) traffic load and the harvested solar energy. The details are as follows: (1) the real MN traffic load traces used to support the findings of this study were obtained from the Big Data Challenge organized by Telecom Italia Mobile (TIM) and the data repository has been cited in this

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) Energy savings versus cluster size

(b) Energy savings versus η for $|O_i| = 6$

FIGURE 7: Energy savings for the multiple BSs case.

article. (2) The real solar energy traces used to support the findings of this study have also been cited in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement no. 675891 (SCAVENGE).

References

[1] "Five Trends to Small Cells 2020," Tech. Rep., Huawei Technologies, Helsinki, Finland, 2016.

[2] M. Patel, Y. Hu, P. Hédé et al., "Mobile edge computing introductory technical white paper," Tech. Rep., ETSI, Sophia-Antipolis, France, 2014.

[3] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 56–61, 2011.

[4] J. Erman and K. Ramakrishnan, "Understanding the super-sized traffic of the super bowl," in *Proceedings of the 2013 conference on Internet measurement conference*, Barcelona, Spain, October 2013.

[5] R. Morabito, "Power consumption of virtualization technologies: An empirical investigation," in *Proceedings of the in IEEE International Conference on Utility and Cloud Computing (UCC)*, Limassol, Cyprus, Dec 2015.

[6] Y. Jin, Y. Wen, and Q. Chen, "Energy efficiency and server virtualization in data centers: An empirical investigation," in *proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM Workshops)*, Orlando, FL, USA, Mar 2012.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Mass, USA, 2016.

[8] S. Abdelwahed, N. Kandasamy, and S. Neema, "Online control for self-management in computing systems," in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Ontario, Canada, 2004.

[9] T. Dlamini, A. F. Gambin, D. Munaretto, and M. Rossi, "Online resource management in energy harvesting bs sites through prediction and soft-scaling of computing resources," in *Proceedings of the 2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications (PIMRC)*, Bologna, Italy, September 2018.

[10] "Open Big Data Challenge," https://dandelion.eu/datamine/open-big-data/.

[11] "Solar Radiation Measurement Data," https://energydata.info/dataset/armenia-solar-radiation-measurement-data-2017.

[12] H. Zhang, J. Cai, and X. Li, "Energy-efficient base station control with dynamic clustering in cellular network," in *Proceedings of the IEEE International Conference on Communications and Networking (CHINACOM)*, Guilin, China, August 2013.

[13] S. Samarakoon, M. Bennis, W. Saad, and M. Latva-Aho, "Dynamic clustering and on/off strategies for wireless small cell networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 2164–2178, 2016.

[14] S. Cai, L. Xiao, H. Yang, J. Wang, and S. Zhou, "A cross-layer optimization of the joint macro-and picocell deployment with sleep mode for green communications," in *Proceedings of the 22nd Wireless and Optical Communications Conference, WOCC 2013*, Chongqing, China, May 2013.

[15] Y. Zhu, Z. Zeng, T. Zhang, and D. Liu, "A QoS-aware adaptive access point sleeping in relay cellular networks for energy efficiency," in *Proceedings of the IEEE Vehicular Technology Conference (VTC Spring)*, Seoul, Korea, May 2014.

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[16] Y. Yuan and P. Gong, "A QoE-orientated base station sleeping strategy for multi-services in cellular networks," in *Proceedings of the International Conference on Wireless Communications and Signal Processing, (WCSP 2015)*, Nanjing, China, October 2015.

[17] F. Han, Z. Safar, and K. J. R. Liu, "Energy-efficient base-station cooperative operation with guaranteed QoS," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3505–3517, 2013.

[18] C. Liu, Y. Wan, L. Tian, Y. Zhou, and J. Shi, "Base station sleeping control with energy-stability tradeoff in centralized radio access networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, USA, December 2015.

[19] A. Bousia, A. Antonopoulos, L. Alonso, and C. Verikoukis, "'Green' distance-aware base station sleeping algorithm in LTE-Advanced," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 1347–1351, Ottawa, Canada, June 2012.

[20] H. Tabassum, U. Siddique, E. Hossain, and M. J. Hossain, "Downlink performance of cellular systems with base station sleeping, user association, and scheduling," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5752–5767, 2014.

[21] Y. Zhu, Z. Zeng, T. Zhang, L. An, and L. Xiao, "An energy efficient user association scheme based on cell sleeping in LTE heterogeneous networks," in *Proceedings of the 2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 75–79, Sydney, Australia, September 2014.

[22] A. Bousia, E. Kartsakli, A. Antonopoulos, L. Alonso, and C. Verikoukis, "Multiobjective auction-based switching-off scheme in heterogeneous networks: to bid or not to bid?" *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9168–9180, 2016.

[23] B. Z. Dongsheng Han and Z. Chen, "Sleep mechanism of base station based on minimum energy cost," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 4202748, 13 pages, 2018.

[24] M. D'Amours, A. Girard, and B. Sanso, "Planning solar in energy-managed cellular networks," *IEEE Access*, vol. 6, pp. 65212–65226, 2018.

[25] N. Piovesan, A. Fernandez Gambin, M. Miozzo, M. Rossi, and P. Dini, "Energy sustainable paradigms and methods for future mobile networks: A survey," *Elsevier - Computer Communications*, vol. 119, pp. 101–117, 2018.

[26] D. Thembelihle, M. Rossi, and D. Munaretto, "Softwarization of mobile network functions towards agile and energy efficient 5G architectures: a survey," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 8618364, 21 pages, 2017.

[27] A. Antonopoulos, E. Kartsakli, A. Bousia, L. Alonso, and C. Verikoukis, "Energy-efficient infrastructure sharing in multi-operator mobile networks," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 242–249, 2015.

[28] M. Oikonomakou, A. Antonopoulos, L. Alonso, and C. Verikoukis, "Evaluating cost allocation imposed by cooperative switching off in multi-operator shared HetNets," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11352–11365, 2017.

[29] J. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2002.

[30] S.-L. Chung, S. Lafortune, and F. Lin, "Limited lookahead policies in supervisory control of discrete event systems," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 37, no. 12, pp. 1921–1935, 1992.

[31] T. Ergen and S. S. Kozat, "Online training of LSTM networks in distributed systems for variable length data sequences," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 5159–5165, 2017.

[32] J. Xu and S. Ren, "Online learning for offloading and autoscaling in renewable-powered mobile edge computing," in *Proceedings of the 59th IEEE Global Communications Conference, GLOBECOM 2016*, Washington, DC, USA, December 2016.

[33] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[34] R. Nathuji and K. Schwan, "VirtualPower: coordinated power management in virtualized enterprise systems," in *Proceedings of the 21st ACM SIGOPS symposium on operating systems principles, SOSP'07*, pp. 265–278, Washington, DC, USA, October 2007.

[35] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in *Proceedings of the annual conference on usenix annual technical conference*, Berkeley, calif, USA, Apr 2005.

[36] C. Jeffrey, A. Darrell, T. Prachi, V. Amin, and D. Ronald, "Managing energy and server resources in hosting centers," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, Alberta, Canada, Oct 2001.

[37] J. Lorch and A. J. Smith, "PACE: A new approach to dynamic voltage scaling," *IEEE Transactions on Computers*, vol. 53, no. 7, pp. 856–869, 2004.

[38] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, OTexts, Melbourne, Australia, 2013.

[39] Network., Network Functions Virtualisation (NFV): Hypervisor Domain, ETSI, Sophia-Antipolis, France, Jan 2015.

[40] "3GPP TS 32.2.297, charging data rececord (CDR) file format and transfer," Tech. Rep., ETSI, Sophia-Antipolis, France, 2016.

[41] C. Peng, S.-B. Lee, S. Lu, H. Luo, and H. Li, "Traffic-driven power saving in operational 3G cellular networks," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom '11)*, pp. 121–132, Las Vegas, Nev, USA, September 2011.

[42] D. Pelleg and A. W. Moore, "X-means: Extending K-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, San Francisco, calif, USA, Jun 2000.

[43] F. B. Abdesslem and A. Lindgren, "Large scale characterisation of YouTube requests in a cellular network," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–9, Sydney, Australia, June 2014.

[44] L. Chen, S. Zhou, and J. Xu, "Energy efficient mobile edge computing in dense cellular networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May 2017.

[45] P.-S. Yu, J. Lee, T. Q. S. Quek, and Y.-W. P. Hong, "Traffic offloading in heterogeneous networks with energy harvesting personal cells-network throughput and energy efficiency," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1146–1161, 2016.

[46] J. Wu, Y. Bao, G. Miao, S. Zhou, and Z. Niu, "Base-station sleeping control and power matching for energy-delay tradeoffs with bursty traffic," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3657–3675, 2016.

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[47] L. Haikun, X. Cheng-Zhong, J. Hai, G. Jiayu, and L. Xiaofei, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of the 20th international symposium on high performance distributed computing*, California, calif, USA, Jun 2011.

[48] "Virtualization for small cells: Overview," Tech. Rep., Small Cell Forum, Draycott, England, 2015.

[49] K. Li, "Performance analysis of power-aware task scheduling algorithms on multiprocessor computers with dynamic voltage and speed," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 11, pp. 1484–1497, 2008.

[50] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Transactions on Cloud Computing*, 2016.

[51] C. Canali, L. Chiaraviglio, R. Lancellotti, and M. Shojafar, "Joint minimization of the energy costs from computing, data transmission, and migrations in cloud data centers," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 580–595, 2018.

[52] M. Chen, W. Saad, and C. Yin, "Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks," *IEEE Wireless Communications*, 2017, https://arxiv.org/abs/1710.02913.

[53] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[54] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*, Springer, 2010.

[55] E. Oh, K. Son, and B. Krishnamachari, "Dynamic base station switching-on/off strategies for green cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2126–2136, 2013.

[56] "Standard Performance Evaluation Corporation," Tech. Rep., SPEC, Virginia, USA, https://www.spec.org/virt_sc2013/results/res2013q2/.

6302, 2019, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2019/8593808 by Universitaetsbibl Augsburg, Wiley Online Library on [10/12/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License