
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Gebrehiwot, Misikir; Lassila, Pasi; Aalto, Samuli

Dynamic load balancing in 5G HetNets for optimal performance-energy tradeoff

Published in:
IFIP Networking 2018

DOI:
[10.23919/IFIPNetworking.2018.8697028](https://doi.org/10.23919/IFIPNetworking.2018.8697028)

Published: 01/01/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Gebrehiwot, M., Lassila, P., & Aalto, S. (2018). Dynamic load balancing in 5G HetNets for optimal performance-energy tradeoff. In *IFIP Networking 2018* IEEE. <https://doi.org/10.23919/IFIPNetworking.2018.8697028>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Dynamic load balancing in 5G HetNets for optimal performance-energy tradeoff

Misikir Eyob Gebrehiwot, Pasi Lassila and Samuli Aalto
Department of Communications and Networking
Aalto University, Finland
Email: {misikir.gebrehiwot, pasi.lassila, samuli.aalto}@aalto.fi

Abstract—We consider optimal energy-aware load balancing of elastic downlink data traffic inside a macrocell with multiple small cells within its coverage area. The model for the problem corresponds to a system of parallel M/M/1-PS queues, where the macrocell is represented by a multiclass M/M/1-PS queue and each small cell is an energy-aware M/M/1-PS queue with additional states for the idle timer and the so-called setup delay. We apply the theory of MDPs to develop a near-optimal state-dependent policy, both for a weighted sum of the performance and energy as well as for the constrained formulation, where energy is minimized subject to a constraint on the performance. Specifically, we utilize the first step of the well-known policy iteration method under which the routing decision for each arrival requires evaluating the marginal future cost of adding the arrival in the small cell or the macrocell. As our main contribution, we derive the associated value functions and the explicit form of the near-optimal FPI policy. The performance of the policy is illustrated through numerical examples.

I. INTRODUCTION

Heterogeneous networks (HetNets) is one of the key enabling technologies for realizing the future 5G networks. Specifically, HetNets address the problem of spatially heterogeneous distribution of the traffic load within a cell by introducing inside the coverage region of the macrocell so-called small cells, sometimes also referred to as pico- or femtocells, with low-power base stations that are operating under the control of the macrocell. The small cells can offer at a traffic hot spot a high transmission rate to nearby users and thus some of the traffic can be off loaded away from the macrocell to the respective small cell. Such load balancing clearly can benefit the performance of the users, for example, by minimizing the delay. However, in modern systems it also important to consider the energy consumption of the system.

Such energy-aware load balancing in HetNets has been studied considerably recently. Typically, the approaches consider a fixed set of users or the optimization only takes into account average traffic parameters, see, e.g., [1], [2], [3], [4], [5]. However, such approaches do not take into account the randomly varying user population and the delay performance of the users. Our focus is to take these aspects explicitly into account by using a queueing theoretic approach.

In more detail, we study the following scenario. We consider a single macrocell with many small cells inside its coverage region serving downlink traffic. The user traffic consists of

elastic flows, roughly corresponding to file transfers controlled by TCP, that are downloaded through the base stations. New flows arrive according to a Poisson process. Thus, the number of active flows varies randomly over time and the flow-level performance is represented by the mean flow-level delay, i.e., the mean file transfer delay. Upon the arrival of a new flow, a load balancing policy will decide whether to serve the flow through the local small cell or the macrocell. From the energy point of view, the macrocell is always on in order to provide coverage in the whole cell area. However, the small cells can be switched off to save energy during low load. Activating again a sleeping small cell incurs a performance cost in the form of a setup delay, thus giving rise to a performance-energy trade-off in the system. Our objective is to develop energy- and delay-aware load balancing algorithms.

In our flow-level model, the macrocell is represented by a multiclass M/M/1-PS queue, where the classes represent flows that arrived in a given small cell but are served by the macrocell. The small cells are modelled as a single-class M/M/1-PS queues with a setup delay, and we additionally allow another control parameter, the so-called idle timer, which defines how long a small cell waits before it falls into sleep mode after the small cell becomes idle (i.e., becomes empty of flows). Thus, the system consists of a set of parallel queues and the load balancing policy decides whether the arrival is routed to the small cell or the macrocell.

To characterize the performance-energy trade-off, we represent the system cost as a weighted sum of the delay and energy. To optimize the cost, we consider dynamic state-dependent policies and apply the theory of Markov Decision Processes (MDP). MDPs have been applied for the load balancing problem in HetNets in [6], [7], [8], [9], but they do not contain any explicit forms of the resulting policies, i.e., value iteration method is used to numerically solve the problem. Also, the system models are different to ours in these papers, or energy aspects are not considered. Note that in our earlier paper [10], we have applied the so-called FPI approach, to be discussed below, but the policy is obtained numerically and, in addition, the system model is slightly different (no idle timer in small cells and macrocell energy model is simpler).

As our main analytical contribution, we are able to define explicitly the (near) optimal policy. Specifically, we utilize the first step of the policy iteration algorithm [11], which gives the so-called FPI (First Policy Iteration) policy. By using a

static probabilistic policy as an initial policy, the cost under the FPI policy becomes separable and the routing decision depends only on the state of the local small cell and the state of the macrocell. The FPI policy is characterized by the so-called value functions of the macrocell and the small cells. We derive the explicit form of the value functions of the delay and power for the small cells represented by an M/M/1-PS queue with setup delay and idle timer. This extends our earlier analytical results for the M/M/1 queue with setup delay but no idle timer in [12]. Also, for the multiclass M/M/1-PS queue, we derive the value function of the power part, while the general form of the performance part is known, see [13], [14]. The resulting policy is still scalable as it only involves comparing the additional cost of allocating the flow in the macrocell or the small cell. The policy is near optimal since typically in policy iteration the largest gain over the initial policy is already obtained with the first iteration step.

Due to the explicit form of the value functions, insights can also be obtained from how the (near) optimal policy behaves. The marginal cost for the performance part in small cells has a weighted JSQ-like (Join-the-Shortest-Queue) structure with an additional constant factor that reflects the state of the server (setup or busy), while the marginal energy cost is just a constant independent of the setup/busy. In the macro cell, the marginal performance cost also has a similar linear form (JSQ-like) and the marginal energy cost is a constant.

In addition to formulating the optimization problem as a minimization problem of the weighted sum, we formulate the problem as a constrained optimization problem. We apply the theory of constrained MDPs [15], [16] and consider minimizing the mean power subject to a constraint on the mean delay. This formulation is often more natural than the weighted sum variant, where the weight is arbitrary. We observe that, by Lagrangian relaxation of the constraint, the constrained problem can be interpreted as the minimization of the weighted sum of the power and delay, where the weight must be determined to satisfy the constraint. Thus, we can apply our results for the FPI policy and we give an iterative algorithm to obtain the optimal weight parameter that defines the (near) optimal policy and solves the constrained problem.

We highlight the properties of the dynamic policies through numerical examples. In particular, we are interested in the impact of the idle timer, which has not been studied previously in this context. We have shown already in our earlier paper that with a single energy-aware M/G/1-PS queue the optimal value of the idle timer is always either zero or infinity for the weighted sum cost function, see [17]. Our numerical results indicate that the same remains true also in this much more complex scenario with parallel queues and non-Poisson arrivals. However, in the case of the constrained optimization formulation, if the setup delay is short enough, a strictly positive and finite optimal choice for the idle timer exists.

The paper is organized as follows. The system model is given in Section II. The optimal load balancing problem is defined in Section III, where also our main analytical contributions are derived. Section IV considers the load bal-

ancing problem as a constrained MDP. Numerical results and conclusions are in Sections V and VI, respectively.

II. MODEL

We consider a heterogeneous wireless system consisting of a single macrocell and K separate small cells located inside the coverage area of the macrocell. The macrocell is indexed by 0 and the small cells by $k = 1, \dots, K$. We assume that the small cells operate in an outband mode, i.e., they have their own radio resources and do not interfere with the macrocell. In addition, we assume that the small cells are far enough from each other so that they do not interfere with each other either.

Traffic consists of elastic downlink data flows (such as TCP file transfers). Let λ_k denote the arrival rate of new flows within the area of small cell k . Each such a flow can be served either by the small cell itself or the macrocell (but not by the other small cells). Upon the arrival, a routing decision must be made whether the flow is attached to the small cell or the macrocell. In addition, let λ_0 denote the arrival rate of those flows (outside the “hotspot” areas covered by the small cells) that can only be served by the macrocell. All the arrival processes are assumed to be independent Poisson processes.

Each small cell k is modeled by a single server PS queue, which implies that flows are scheduled in each time slot so that all resources are given to one flow at a time and the flows are served in a round-robin manner. We assume that the service time S_k of an arbitrary flow in small cell k is exponentially distributed with mean $E[S_k] = 1/\mu_k$. The mean service time $E[S_k]$ is the average time needed to complete the transfer of a random flow if there were no other flows to be carried by the same cell. We note that the service time is affected at least by the size of the original flow, the location of the corresponding mobile terminal (within the small cell), and the radio channel conditions during the flow transfer. However, since the scheduler of the small cell does not utilize these features, we do not model them separately.

For each small cell, we apply the DELAYEDOFF sleep state control (according to the terminology of [18]). As long as there are flows in the small cell to be served, the state of the server is said to be BUSY, but as soon as the system becomes empty, the state changes to IDLE. The server remains IDLE as long as one of the following events take place. Either a new flow arrives, in which case the server becomes again BUSY and starts serving the new flow, or the idle timer (associated with the idle server) expires, in which case the server is immediately switched OFF. In our model, the length I_k of the idle timer of small cell k is assumed to be independently and exponentially distributed with mean $E[I_k] = 1/\omega_k$. In the latter case, the server remains OFF until a new flow is routed to small cell k and the server is put to the SETUP state. After a setup delay D_k , which in our model is assumed to be independently and exponentially distributed with mean $E[D_k] = 1/\delta_k$, the server becomes again BUSY and starts serving the waiting flows.

The state of small cell k at time t is described by the pair $(X_k(t), B_k(t))$, where $X_k(t)$ refers to the number of flows

and $B_k(t)$ to the state of the server. Note that server k is

$$\begin{aligned} \text{BUSY,} & \quad \text{if } X_k(t) > 0 \text{ and } B_k(t) = 1; \\ \text{IDLE,} & \quad \text{if } X_k(t) = 0 \text{ and } B_k(t) = 1; \\ \text{OFF,} & \quad \text{if } X_k(t) = 0 \text{ and } B_k(t) = 0; \\ \text{SETUP,} & \quad \text{if } X_k(t) > 0 \text{ and } B_k(t) = 0. \end{aligned}$$

We denote the power consumption in these energy states of small cell k by $P_k^{\text{E-STATE}}$, and we assume that

$$0 = P_k^{\text{o}} < P_k^{\text{i}} \leq \min\{P_k^{\text{s}}, P_k^{\text{b}}\},$$

where superscripts o, i, s, and b refer to OFF, IDLE, SETUP, and BUSY states, respectively.

The macrocell (0) serves $K + 1$ different classes of flows. Class 0 refers to those flows that can only be served by the macrocell and class k to those flows that arrive in the area of small cell k but are routed to the macrocell. The macrocell itself is modeled by a single server multiclass PS queue, which again implies that flows are scheduled in each time slot so that all resources are given to one flow at a time and the flows are served in a round-robin manner. We assume that the service time $S_{0,k}$ of an arbitrary flow in class $k \in \{0, 1, \dots, K\}$ is exponentially distributed with mean $1/\mu_{0,k}$.

For the macrocell, we do not apply any sleep state control, but it is a NEVEROFF server (according to the terminology of [18]). Thus, the state of the macrocell at time t is described by the vector $X_0(t) = (X_{0,0}(t), X_{0,1}(t), \dots, X_{0,K}(t))$, where $X_{0,k}(t)$ refers to the number of flows in class k . Note that server 0 is

$$\begin{aligned} \text{BUSY,} & \quad \text{if } |X_0(t)| > 0; \\ \text{IDLE,} & \quad \text{if } |X_0(t)| = 0, \end{aligned}$$

where

$$|X_0(t)| = X_{0,0}(t) + X_{0,1}(t) + \dots + X_{0,K}(t)$$

denotes the total number of flows in macrocell. The power consumption of macrocell is denoted by $P_0^{\text{E-STATE}}$, and we assume that

$$0 < P_0^{\text{i}} \leq P_0^{\text{b}}.$$

III. OPTIMAL LOAD BALANCING PROBLEM

Our objective is to develop dynamic, state-dependent load balancing policies that simultaneously take into account both the power consumption and the delay of the flows. The load balancing policy decides for each arriving flow in the small cells, whether the arrival is served by the local small cell or the macrocell. For our model, optimal policies can be developed in the framework of MDPs [11].

Before discussing the optimization, we state a necessary condition for the stability of the system. As in [10], the macrocell is the bottleneck in the system, because it serves as an overflow system for the arrivals in the small cells, and thus, the maximal stability condition for any load balancing policy is given by

$$\frac{\lambda_0}{\mu_{0,0}} + \sum_{k=1}^K \frac{(\lambda_k - \mu_k)^+}{\mu_{0,k}} < 1. \quad (1)$$

For the optimization, the cost rates with respect to the performance (delay) and the energy need to be defined first. Let the vector $x_0 = (x_{0,0}, \dots, x_{0,K})$ denote a given state of the macrocell. Similarly, we denote by $x = (x_1, \dots, x_K)$ and $b = (b_1, \dots, b_K)$ vectors for the number of flows and the state of the server in each small cell. Given that there are (x_0, x) flows in the system, the instantaneous cost rate for the performance, $c^{\text{P}}(x_0, x)$, is given by

$$c^{\text{P}}(x_0, x) = \sum_{k=0}^K x_{0,k} + \sum_{k=1}^K x_k, \quad (2)$$

i.e., it is the total number of flows in the system. For the energy, the instantaneous cost rate, $c^{\text{E}}(x_0, x, b)$, is the total instantaneous power in the given state and it equals

$$\begin{aligned} c^{\text{E}}(x_0, x, b) = & \quad 1_{|x_0|=0} P_0^{\text{i}} + 1_{|x_0|>0} P_0^{\text{b}} + \\ & \quad \sum_{k=1}^K (1_{x_k>0, b_k=1} P_k^{\text{b}} + 1_{x_k=0, b_k=1} P_k^{\text{i}} + 1_{x_k>0, b_k=0} P_k^{\text{s}}), \end{aligned} \quad (3)$$

where 1_A denotes the indicator function of the event A . To characterize the trade-off between performance and energy, the total instantaneous cost rate in state (x_0, x, b) , $c(x_0, x, b)$, is defined as the weighted sum of performance and energy,

$$c(x_0, x, b) = w_1 c^{\text{P}}(x_0, x) + w_2 c^{\text{E}}(x_0, x, b),$$

where $w_1, w_2 \geq 0$ are the weight parameters.

We denote by π the set of all possible load balancing policies that are stable under the condition (1). For a given policy π , let $E[X^\pi]$ and $E[P^\pi]$ denote the mean total number of flows in the system and the resulting mean power consumption, respectively. Our objective is to consider the following optimization problem,

$$\min_{\pi} w_1 E[X^\pi] + w_2 E[P^\pi]. \quad (4)$$

Note that by dividing (4) with the total arrival rate $\sum_k \lambda_k$, the objective is, by Little's law, equal to to the weighted sum of the mean flow delay and the mean energy per flow.

A. Optimal dynamic policy

The optimization problem (4) is an MDP and can be solved numerically iteratively with the policy iteration method [11], as follows. Let y denote a vector of the entire state of the system, i.e., $y = (x_0, x, b)$. In the load balancing policy, associated with each state (x_0, x, b) there is a set of actions $\mathcal{A}(y)$ relating to the decisions where the arrivals are routed. Let π_n denote the policy at the n^{th} iteration step. The iterated policy policy at the next step, π_{n+1} , is obtained by solving in each state y the following optimality equation

$$\begin{aligned} \pi_{n+1}(y) = & \\ \arg \min_{a \in \mathcal{A}(y)} & \left(c(y) - \bar{c}^{\pi_n} + \sum_{y'} q_{y,y'}(a) v^{\pi_n}(y') \right), \forall y, \end{aligned} \quad (5)$$

where \bar{c}^{π_n} is the mean cost under policy π_n , $q_{y,y'}(a)$ is the transition rate from state y to state y' when action a is taken

and $v^{\pi_n}(y')$ is the so-called value function of state y' for policy π_n .

The value function of each state under given policy π gives the mean difference in the cost when starting initially the process from the state y and the long term average cost \bar{c}^π . The value function of each state are, on the other hand, characterized by the following set of linear equations

$$c(y) - \bar{c}^\pi + \sum_{y'} q_{y,y'}(a^\pi(y))(v^\pi(y') - v^\pi(y)) = 0, \quad \forall y, \quad (6)$$

where $a^\pi(y)$ is the action taken in state y under policy π .

The policy iteration algorithm can be started from any stable initial policy π^0 , for which the value functions are first solved from (6). Then the iterated policy π^1 is obtained from (5), and the process repeats. The policy iteration is guaranteed to converge to the optimal policy in a finite number of iterations [11]. Unfortunately, for the state space consists of $(3K + 1)$ dimensions which makes it impossible to solve the optimal policy numerically.

B. Near-optimal FPI policy

In the FPI (First Policy Iteration) approach, the idea is that by selecting the initial policy appropriately the first step of the optimization (5) can be carried out explicitly. Typically this first step already gives the largest improvement, which makes the FPI policy near optimal.

Consider now a probabilistic policy determined by the vector $p = (p_1, \dots, p_K)$, where each component p_k gives the probability to route the incoming class- k flow to the small cell k and with probability $(1 - p_k)$ the flow is routed to the macrocell. By using a probabilistic policy as the initial policy renders the stochastic behavior of the macrocell and the small cells independent of each other. Thus, the relative value of the state $y = (x_0, x, b)$ can be expressed as

$$v(x_0, x, b) = v_0(x_0) + \sum_{k=1}^K v_k(x_k, b_k), \quad (7)$$

where $v_0(x_0)$ and $v_k(x_k, b_k)$ are the value functions of the macrocell and small cell k , respectively.

A reasonable selection for the initial probabilistic policy is to balance the load in all the cells, as much as possible. We denote this policy by p^{LB} . Assume that the classes of the small cells, $k = 1, \dots, K$, are ordered in a descending order according to the cell loads, i.e., $\lambda_1/\mu_1 > \dots > \lambda_K/\mu_K$. Note that for all those small cells where the load is already less than the load of the macrocell when serving its own traffic, no traffic can be moved to the macrocell, i.e., the corresponding $p_k^{\text{LB}} = 1$. Thus, let k^* denote the index value of the last small cell from which traffic can be moved to the macrocell, i.e.,

$$k^* = \{\max k = 1, \dots, K : \lambda_k/\mu_k > \lambda_0/\mu_{0,0}\}.$$

It is easy to see that the load is then equalized by setting

$$\begin{cases} p_k^{\text{LB}} = \frac{\mu_k}{\lambda_k} \cdot \frac{\lambda_0/\mu_{0,0} + \sum_{k=1}^{k^*} \frac{\lambda_k}{\mu_{0,k}}}{1 + \sum_{k=1}^{k^*} \frac{\mu_k}{\mu_{0,k}}}, & k = 1, \dots, k^*, \\ p_k^{\text{LB}} = 1, & k = k^* + 1, \dots, K. \end{cases}$$

The optimization (5) only concerns the arrival events. Thus, the decision to serve the arrival in small cell k or route it to the macrocell simply consists of evaluating the additional cost of adding the arrival to the small cell or to the macrocell. Due to the separable form of the value function (7), the action to serve the arrival in the macrocell or the small cell k in state (x_0, x, b) is given by

$$a_k^{\text{FPI}}(x_0, x, b) = \begin{cases} \text{macrocell,} & \text{if } v_0(x_0, 0, \dots, x_0, k+1, \dots, x_0, K) - \\ & v_0(x_0, 0, \dots, x_0, K) < \\ & v_k(x_k+1, b_k) - v_k(x_k, b_k) \\ \text{small cell,} & \text{otherwise.} \end{cases} \quad (8)$$

In summary, the main advantage in the FPI approach is that it allows us to systematically construct an optimized dynamic policy, which is near optimal and (almost) fully explicit, as will be seen in the following section. Note that through our results, the FPI policy is also fully specified in the entire state space of the system, i.e., there is no need for any truncation to evaluate the mean cost under the FPI policy by simulation.

C. Value functions

Here we present the main analytical results of the paper: explicit value functions for the performance and energy for the M/M/1-PS DELAYDOFF and multiclass M/M/1-PS models.

1) *M/M/1-PS DELAYEDOFF*: Consider a generic M/M/1-PS DELAYEDOFF queue with arrival rate λ , mean service time $E[S] = 1/\mu$, mean idle timer $E[I] = 1/\omega$, and mean setup delay $E[D] = 1/\delta$. Assume that the system is stable, i.e., $\rho < 1$, where load $\rho = \lambda E[S]$. In addition, let P^o , P^i , P^s , and P^b denote the power consumption in energy states OFF, IDLE, SETUP, and BUSY, respectively.

Proposition 1: For a stable M/M/1-PS DELAYEDOFF queue, the relative value function with respect to performance in state (n, b) is given by

$$\begin{aligned} v^p(n, 1) - v^p(0, 0) &= \frac{E[S]n(n+1)}{2(1-\rho)} \\ &- \frac{E[D]n\rho(1+\lambda E[D])}{(1-\rho)(1+\lambda E[D] + \lambda E[I])} \\ &- \frac{E[I]\lambda E[D]}{(1-\rho)(1+\lambda E[D] + \lambda E[I])}; \\ v^p(n, 0) - v^p(0, 0) &= \frac{E[S]n(n+1)}{2(1-\rho)} \\ &+ \frac{E[D]n(1+\lambda E[D])}{1+\lambda E[D] + \lambda E[I]} \\ &+ \frac{E[I](n-1)\lambda E[D]}{(1-\rho)(1+\lambda E[D] + \lambda E[I])}. \end{aligned}$$

Proof: The Howard equations for the system are:

$$\begin{aligned} -\bar{c}^p + \lambda(v^p(1, 0) - v^p(0, 0)) &= 0, \\ n - \bar{c}^p + \lambda(v^p(n+1, 0) - v^p(n, 0)) + \\ \delta(v^p(n, 1) - v^p(n, 0)) &= 0, \quad n \geq 1, \end{aligned}$$

$$\begin{aligned}
n - \bar{c}^p + \lambda(v^p(n+1, 1) - v^p(n, 1)) + \\
\mu(v^p(n-1, 1) - v^p(n, 1)) &= 0, \quad n \geq 1, \\
-\bar{c}^p + \lambda(v^p(1, 1) - v^p(0, 1)) + \\
\omega(v^p(0, 0) - v^p(0, 1)) &= 0,
\end{aligned}$$

where \bar{c}^p denotes the mean number of flows given by

$$\bar{c}^p = E[X] = \lambda E[T] = \frac{\rho}{1-\rho} + \frac{\lambda E[D](1 + \lambda E[D])}{1 + \lambda E[D] + \lambda E[I]}.$$

Now it is a straightforward task to verify that these equations are satisfied by the proposed relative value function. ■

Corollary 1: For a stable M/M/1-PS DELAYEDOFF queue, the marginal performance cost in state (n, b) is given by

$$\begin{aligned}
v^p(n+1, 1) - v^p(n, 1) &= \frac{E[S](n+1)}{1-\rho} \\
&- \frac{E[D]\rho(1 + \lambda E[D])}{(1-\rho)(1 + \lambda E[D] + \lambda E[I])}, \quad n \geq 0; \\
v^p(1, 0) - v^p(0, 0) &= \frac{E[S]}{1-\rho} \\
&+ \frac{E[D](1 + \lambda E[D])}{1 + \lambda E[D] + \lambda E[I]}; \\
v^p(n+1, 0) - v^p(n, 0) &= \frac{E[S](n+1)}{1-\rho} \\
&+ \frac{E[D](1 + \lambda E[D])}{1 + \lambda E[D] + \lambda E[I]} \\
&+ \frac{E[I]\lambda E[D]}{(1-\rho)(1 + \lambda E[D] + \lambda E[I])}, \quad n \geq 1.
\end{aligned}$$

Proposition 2: For a stable M/M/1-PS DELAYEDOFF queue, the relative value function with respect to the energy in state (n, b) is given by

$$\begin{aligned}
v^e(n, 1) - v^e(0, 0) &= n \cdot \gamma \\
&+ \frac{E[I]((P^i - P^o) + \lambda E[D](P^i - P^s))}{1 + \lambda E[D] + \lambda E[I]}, \\
v^e(n, 0) - v^e(0, 0) &= n \cdot \gamma \\
&+ \frac{E[D]((P^s - P^o) + E[I](P^i - P^o))}{1 + \lambda E[D] + \lambda E[I]}, \\
\gamma &= \frac{E[S]((P^b - P^o) + \lambda E[D](P^b - P^s))}{1 + \lambda E[D] + \lambda E[I]} \\
&+ \frac{E[S]\lambda E[I](P^b - P^i)}{1 + \lambda E[D] + \lambda E[I]}.
\end{aligned}$$

Proof: The Howard equations for the system are:

$$\begin{aligned}
P^o - \bar{c}^e + \lambda(v^e(1, 0) - v^e(0, 0)) &= 0, \\
P^s - \bar{c}^e + \lambda(v^e(n+1, 0) - v^e(n, 0)) + \\
\delta(v^e(n, 1) - v^e(n, 0)) &= 0, \quad n \geq 1, \\
P^b - \bar{c}^e + \lambda(v^e(n+1, 1) - v^e(n, 1)) + \\
\mu(v^e(n-1, 1) - v^e(n, 1)) &= 0, \quad n \geq 1, \\
P^i - \bar{c}^e + \lambda(v^e(1, 1) - v^e(0, 1)) + \\
\omega(v^e(0, 0) - v^e(0, 1)) &= 0,
\end{aligned}$$

where \bar{c}^e denotes the mean power consumption given by

$$\bar{c}^e = E[P] = \rho P^b + (1-\rho) \frac{P^o + \lambda E[D]P^s + \lambda E[I]P^i}{1 + \lambda E[D] + \lambda E[I]}.$$

It is again a straightforward task to verify that these equations are satisfied by the proposed relative value function. ■

Corollary 2: For a stable M/M/1-PS DELAYEDOFF queue, the marginal energy cost in state (n, b) is given by

$$\begin{aligned}
v^e(n+1, 1) - v^e(n, 1) &= \gamma, \quad n \geq 0; \\
v^e(1, 0) - v^e(0, 0) &= \gamma \\
&+ \frac{E[D]((P^s - P^o) + E[I]((P^i - P^o)))}{1 + \lambda E[D] + \lambda E[I]}, \\
v^e(n+1, 0) - v^e(n, 0) &= \gamma, \quad n \geq 1,
\end{aligned}$$

where γ is defined in Proposition 2.

The explicit value functions for the M/M/1-PS DELAYEDOFF model are novel results and not available in the literature. The results in Corollary 1 and 2 are used when evaluating the marginal cost of serving the flow in small cell k , in (8). In the marginal cost expressions of Corollary 1 and 2, the arrival rate in small cell k after the initial policy is $\lambda_k = p_k^{LB} \lambda$.

Also, observe that in Corollary 1, the marginal cost with respect to the performance has a linear cost with respect to the number of jobs, i.e., similarly to the JSQ rule (Join-the-Shortest-Queue), but in addition there is a positive or negative constant factor depending on whether the server is in sleep/setup state or busy state. Thus, it is from the future cost better to keep an already busy server busy than to wake it up, which is also logical. On the other hand, by Corollary 2, the marginal energy cost is interestingly constant and does not depend on busy/setup state, unless the server is switched off.

2) *MULTICLASS M/M/1-PS NEVEROFF:* Consider a generic multiclass M/M/1-PS NEVEROFF queue with $K+1$ classes of customers indexed by $k = 0, 1, \dots, K$. Let λ_k and $E[S_k] = 1/\mu_k$ denote the arrival rate and the mean service time for class k , respectively. In addition, let $\lambda = \lambda_0 + \lambda_1 + \dots + \lambda_K$ denote the total arrival rate and

$$E[S] = \frac{1}{\lambda} (\lambda_0 E[S_0] + \lambda_1 E[S_1] + \dots + \lambda_K E[S_K])$$

refer to the mean service time over all the customers. Assume that the system is stable, i.e., $\rho < 1$, where load $\rho = \lambda E[S]$. Finally, let P^i and P^b denote the power consumption in energy states IDLE and BUSY, respectively.

Proposition 3: For a stable multiclass M/M/1-PS NEVEROFF queue, the relative value function with respect to performance in state $\mathbf{n} = (n_0, n_1, \dots, n_K)$ is given by

$$v^p(\mathbf{n}) - v^p(\mathbf{0}) = \sum_{k=0}^K a_k (n_k^2 + n_k) + \sum_{k=0}^{K-1} \sum_{\ell=k+1}^K 2a_{k,\ell} n_k n_\ell,$$

where the coefficients a_k and $a_{k,\ell}$ ($k < \ell$) are solved from the following system of linear equations:

$$\begin{aligned} 1 + 2 \sum_{i=0}^K \lambda_i a_{k,i} - 2\mu_k a_k &= 0, \quad 0 \leq k \leq K; \\ 1 + \sum_{i=0}^K \lambda_i (a_{k,i} + a_{\ell,i}) - (\mu_k + \mu_\ell) a_{k,\ell} &= 0, \\ 0 \leq k < \ell \leq K, \end{aligned}$$

with notations $a_{k,k} = a_k$ and $a_{k,\ell} = a_{\ell,k}$ for any k, ℓ .

Proof: The general result for a multiclass M/M/1-PS NEVEROFF queue was proved in [13]. ■

Proposition 4: For a stable multiclass M/M/1-PS NEVEROFF queue, the relative value function with respect to energy in state $\mathbf{n} = (n_0, n_1, \dots, n_K)$ is given by

$$v^e(\mathbf{n}) - v^e(\mathbf{0}) = \sum_{k=0}^K E[S_k] n_k (P^b - P^i),$$

where $\mathbf{0} = (0, 0, \dots, 0)$ is the null vector.

Proof: The Howard equations for the system read as follows:

$$\begin{aligned} P^i - \bar{c}^e + \sum_{k=0}^K \lambda_k (v^e(\mathbf{e}_k) - v^e(\mathbf{0})) &= 0, \\ P^b - \bar{c}^e + \sum_{k=0}^K \frac{n_k \mu_k}{n_0 + \dots + n_K} (v^e(\mathbf{n} - \mathbf{e}_k) - v^e(\mathbf{n})) &+ \\ \sum_{k=0}^K \lambda_k (v^e(\mathbf{n} + \mathbf{e}_k) - v^e(\mathbf{n})) &= 0, \quad \mathbf{n} \neq \mathbf{0}, \end{aligned}$$

where \mathbf{e}_k is a unit vector into direction k and \bar{c}^e denotes the mean power consumption given by

$$\bar{c}^e = E[P] = \rho P^b + (1 - \rho) P^i.$$

It is again a straightforward task to verify that these equations are satisfied by the proposed relative value function. ■

Corollary 3: For a stable multiclass M/M/1-PS NEVEROFF queue, the marginal cost with respect to energy in state $\mathbf{n} = (n_0, n_1, \dots, n_K)$ is given by

$$v^e(\mathbf{n} + \mathbf{e}_k) - v^e(\mathbf{n}) = E[S_k] (P^b - P^i),$$

where \mathbf{e}_k is the unit vector into direction k .

The explicit form of the value function for the energy part in the multiclass M/M/1-PS model is a again novel result and not available in the literature. The results in Proposition 3 and Corollary 3 are used when evaluating the additional cost of serving the flow in the macrocell in (8). Note that after the initial policy each class $k = 1, \dots, K$ has arrival rate $\lambda_k = (1 - p_k^{\text{LB}}) \lambda$.

As the value function for the performance has a quadratic form, see Proposition 3, it is clear that the marginal performance cost $v^p(\mathbf{n} + \mathbf{e}_k) - v^p(\mathbf{n})$ has in general a linear form. On the other hand, by Corollary 3, the marginal energy cost is constant.

IV. CONSTRAINED OPTIMIZATION PROBLEM

In Section III the performance-energy tradeoff was characterized by a weighted sum of the performance and the energy (4). In this formulation, defining appropriate weights in a given setting may be difficult. Instead, a more natural characterization of the tradeoff can be to consider the following constrained optimization formulation: the objective is to determine the policy π that minimizes the energy consumption subject to a constraint of the maximum mean delay T^{max} ,

$$\begin{aligned} \min_{\pi} E[P^{\pi}] & \\ \text{s.t. } \frac{1}{\sum_k \lambda_k} E[X^{\pi}] &\leq T^{\text{max}}, \end{aligned} \quad (9)$$

where the instantaneous costs for the performance and energy are given by (2) and (3), respectively. Note that above again Little's result has been applied in the delay constraint.

The optimization problem (9) can be analyzed as a constrained MDP [15], [16] by applying standard Lagrangian techniques. The idea is to introduce the so-called Lagrange parameter $\beta \geq 0$, and consider the following form for the immediate costs

$$c(x_0, x, b; \beta) = c^p(x_0, x) + \beta c^e(x_0, x, b).$$

In this way, for any given fixed value of β , the problem is in fact a standard unconstrained MDP with a weighted objective function (4) as defined in Section III, and the optimal policy $\pi^*(\beta)$ can be obtained by using, e.g., the policy iteration algorithm. Assuming that a feasible solution to (9) exists, there exists an optimal β^* such that the resulting policy $\pi^*(\beta^*)$ is also the optimal solution to (9), or it may be that to satisfy the constraint in (9) as an equality, the optimal policy is an appropriately randomized policy between two policies associated with β_L^* and β_H^* , such that $E[X^{\pi^*(\beta_L^*)}] / \sum_k \lambda_k < T^{\text{max}}$ and $E[X^{\pi^*(\beta_H^*)}] / \sum_k \lambda_k > T^{\text{max}}$, respectively, see [16], [15].

We have applied the following simple iterative algorithm to determine β^* , similarly as in [19]. Let n here denote the iteration round. Given the current value β_n at the n^{th} iteration, the next value β_{n+1} is obtained from

$$\beta_{n+1} = \beta_n + \frac{1}{n} \cdot \left(\frac{E[X^{\pi(\beta_n)}]}{\sum_k \lambda_k} - T^{\text{max}} \right). \quad (10)$$

The iteration is stopped after a given number of iterations. Additionally, for each value of β_n , due to the high dimensionality of the state space in our problem, the policy iteration can not be iterated until the optimal policy is found. Instead, we apply only the FPI policy for which the explicit expressions have been given in Section III for the weighted cost. Also due to the enormous size of the state space, the performance and energy costs are obtained through simulation. This gives us finally the near optimal policy and the simulated solution to the original constrained optimization problem (9).

To illustrate the constrained optimization approach, we consider the simple model with one small cell with the following parameters: $E[I] = 1$, $E[D] = 0.1$, $\lambda_1 = 12.6$, $\mu_1 = 18.73$, $P_1^b = P_1^s = 100$, $P_1^i = 70$ and $P_1^o = 0$. In the macrocell, there

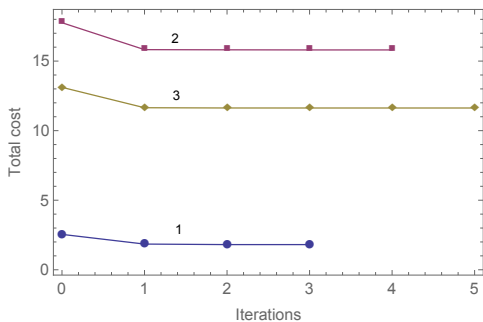


Fig. 1. Illustration of the convergence of the cost in the policy iteration as a function of the iterations (x-axis) and the convergence of the Lagrange parameter β for first three β -iterations (labeled as 1, 2 and 3) in the constrained MDP.

are thus two classes with following parameters: $\lambda_0 = 2$, $\mu_{0,0} = 12.34$, $\mu_{0,1} = 6.37$, $P_0^b = 1000$ and $P_0^i = 700$. Figure 1 illustrates the policy iteration and the iteration with respect to β . In the figure, we show the result of the convergence of the policy iteration for the first three values of β . The initial value of β corresponds to the curve labeled 1 and on the x-axis the policy iteration convergence is shown as a function of the iteration count. In this case, the policy converged with 3 iterations. Then β is adapted according to (10) and the policy iteration is applied until convergence, see curve labeled 2, etc. In this small example, the policy iteration can be numerically evaluated until convergence. Later in our numerical results this is not possible due to the enormous state space and we apply the FPI policy only, which is then simulated to obtain the cost. However, observe that the largest improvement is always obtained with the first step of the policy iteration, i.e., the FPI policy, and the reduction in cost after that is marginal. Thus, we can argue that the FPI policy is indeed near optimal.

V. NUMERICAL RESULTS

Here, we study the performance of proposed policies through simulation runs. We consider a system consisting of one macro and four small cells. Small cells can be in a sleep state where they do not consume power. Otherwise, power consumption of small cell k in the busy, setup and idle states are $P_k^b = P_k^s = 100$ and $P_k^i = 70$ W, respectively. The macro cell consumes 1000 W when it is busy and 700 W when idle.

Service rate for a request originating from within a small cell is $\mu_k = 18.73 \text{ s}^{-1}$ if it is served by the small cell, and $\mu_{0,k} = 6.37 \text{ s}^{-1}$ if offloaded to the macro. Additionally, the macro cell also serves users that are outside the coverage area of the small cells with a rate $\mu_0 = 12.34 \text{ s}^{-1}$, obtained by assuming file sizes of 5 Mb and typical measured mean channel qualities, see [10] for more detailed justifications. In the entire simulation study we set arrival rate in the macro cell to $\lambda_0 = 2 \text{ s}^{-1}$, and systematically choose small cell arrival rates so that the load on the small cell is 0.05, 0.2, 0.32, and 0.5.

We study the impact of mean idle timer ($E[I]$) and mean setup delay ($E[D]$) on the performance of the proposed FPI policy. We start by considering unconstrained minimization of

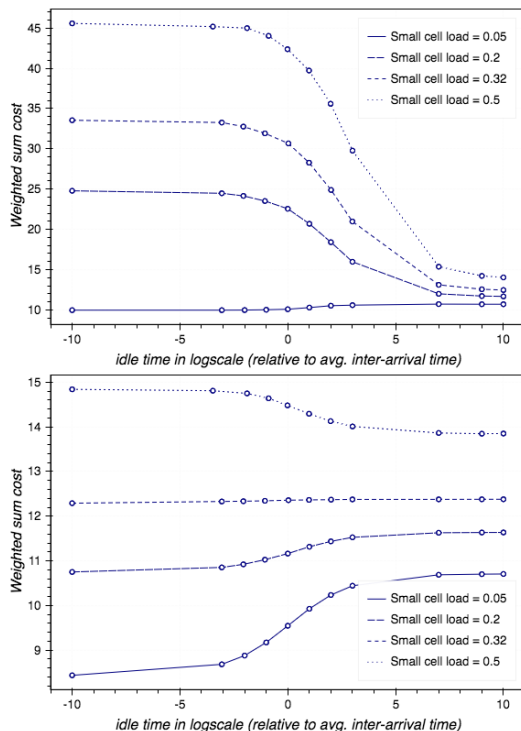


Fig. 2. Energy response time weighted sum cost for $E[D] = 1$ (upper figure) and $E[D] = 0.05$ (lower figure). Mean idle timer (x-axis) is shown in base-2 logarithmic scale.

the weighed sum cost in the following section and then the constrained problem.

A. Weighted sum cost

Figure 2 shows the weighted sum cost of mean response time and mean power consumption as a function of idle timer for two mean setup delay values, $E[D] = 1$ (upper figure) and $E[D] = 0.05$ (lower figure), and weight $\beta = 0.01$, i.e., $w_1 = 1$ and $w_2 = 0.01$. That is, the employed policy is $\pi^{\text{FPI}}(0.01)$. The FPI policy is guaranteed to decrease the cost relative to the initial policy, which is the static load balancing policy. In this setting, the reduction in the cost is typically approximately 10%, e.g., with $E[D] = 1$ and $E[I] = 0$ at load $\rho = 0.5$ the corresponding static LB cost is 50.5, while the cost of the FPI policy is 46. However, the gain can be significantly affected by the choice of weight parameter β ; decreasing β would increase the gain.

A more important aspect here is the impact of the mean idle timer. When setup delay is long, weighted sum cost decreases as a function of idle timer except at low load. Similarly in the short setup delay case, weighted sum cost remains monotonous as a function of idle timer, and whether it is an increasing or decreasing function depends on the load. Even though the unit on the x-axis in Figure 2 is the mean interarrival time (in log-2 base), these properties remain the same even if the curves would be shown as a function of the absolute value of the mean idle timer. Thus, in this scenario with FPI policy, the optimal configuration (with respect to idle timer) is either

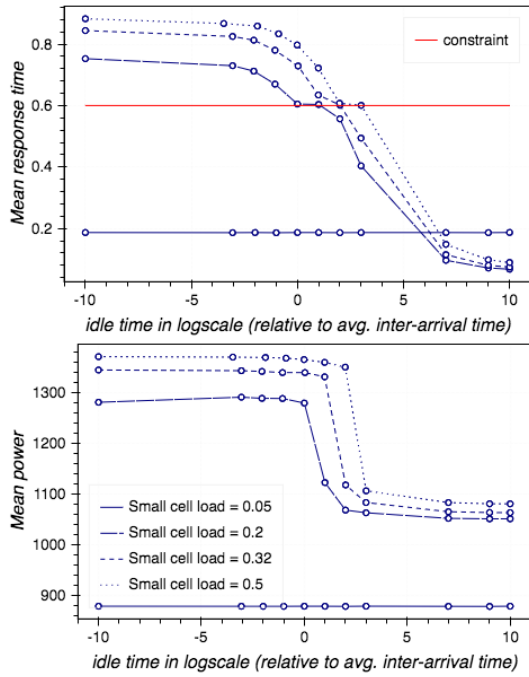


Fig. 3. Performance of $\text{FPI}(\beta^*)$ policy as a function of mean idle timer with $E[D] = 1$ s. Mean idle timer (x-axis) is shown in base-2 logarithmic scale.

NEVEROFF or INSTANTOFF, with low load and short setup delay favoring INSTANTOFF. The results indicate that the general theoretical result that in a single energy-aware M/G/1-PS queue the optimal value of the idle timer is always either zero or infinity, see [17], applies also in this more complex setting without Poisson arrivals (FPI policy makes arrivals at each queue non-Poisson) for the ERWS cost function.

B. Constrained optimization

Figure 3 illustrates mean response time and mean power consumption of the system under $\pi^{\text{FPI}}(\beta^*)$ policy as a function of average idle timer on small cells with $E[D] = 1$. The system behaves more intuitively with respect to mean response time, that is, the longer the cells are allowed to wait in idle state the shorter the response time will be on average. We also observe that largest improvements in mean response time are achieved for average idle timer values between 0.5 and 3 times the mean inter-arrival time. However, for a very light load (0.05 in the figure), longer idle timer has no effect as the policy puts all the traffic in the macro cell.

For points above the delay constraint (red solid line), the policy selects $\beta = 0$, as this is the best we can do to force mean response time as close as possible to the delay constraint under the given idle timer value. However, as soon as idle timer value is high enough so that the constraint is met, the policy starts to minimize mean power by selecting a β^* value different from zero, which explains the less steep decrease in mean response time near the constraint.

Now we focus on the power consumption plot in Figure 3. For the smallest load value of 0.05, the policy uses the

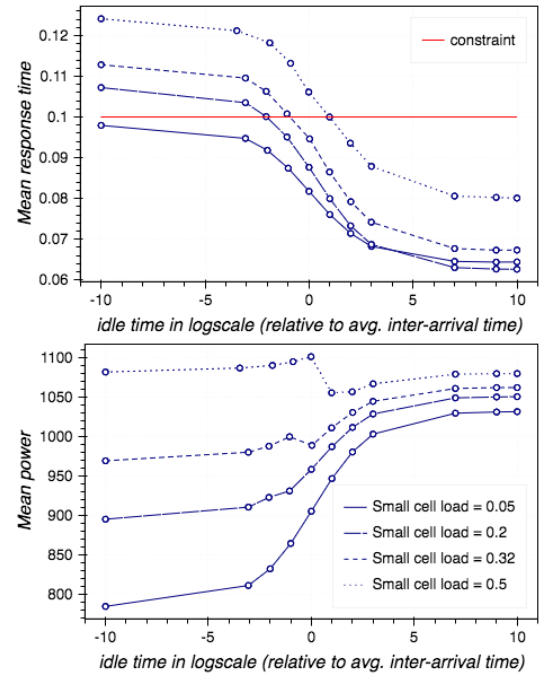


Fig. 4. Performance of $\text{FPI}(\beta^*)$ policy as a function of mean idle timer with $E[D] = 0.05$. Mean idle timer (x-axis) is shown in base-2 logarithmic scale.

macro cell only. In this case, mean power consumption is not affected by the choice of idle timer value, as long as it is finite, i.e., they are not configured as NEVEROFF. For load values 0.2, 0.32, and 0.5, the average power consumption decreases as idle timer increases. This looks counter-intuitive as shorter idle timers should enable the small cells to go to sleep more frequently, which should result in reduced average power. However, all three load values are too large to be handled by the macro cell alone, which means sleeping small cells will need to be started frequently. Idle power is avoided by putting cells to sleep, only to be followed by a higher power consumption in the setup state. The effect is amplified by the fact that setup delay is much longer than the average inter-arrival time (and hence the idle timer) resulting in a system that consumes more power than its energy-aware counterpart.

Therefore, for the given setup delay value, the optimal configuration is either INSTANTOFF or NEVEROFF. INSTANTOFF is optimal when load is low enough so that small cells can be switched off completely, whereas NEVEROFF is optimal in all other cases.

We further investigate the impact of setup delay by keeping all other parameters and considering a very short mean setup delay value of $E[D] = 0.05$, which is in the same order as the service time in the small cells. Figure 4 shows mean response time and mean power of such as system.

The mean response time curve roughly exhibits the same behavior as discussed above. However, mean power increases as a function of idle timer except for some discontinuities (explained below). With the setup delay being very short, the high setup power has less impact on the mean power

compared to the idle power, which makes the INSTANTOFF configuration of small cells an optimal choice for minimizing energy. However, looking back at the mean response time plot, the response time constraint enforces a non-zero idle timer for most of the load values. In this case, a DELAYEDOFF configuration will be optimal with the idle timer set to the smallest value that satisfies the response time constraint.

Notice the discontinuities in the mean power curves. When idle timer is too small, the mean response time does not meet the constraint, which leads to $\beta = 0$. But when the idle timer is long enough, so that the constraint is met with something to spare we start optimizing with respect to mean power, which explains the discontinuities. Note also that for the lowest load scenario, all idle timer values give feasible mean response times resulting in a smooth mean power curve.

VI. CONCLUSIONS

We have considered energy efficient load balancing in a system consisting of a macrocell with several small cells inside its coverage area. The system is modeled as a set of parallel queues consisting of a multiclass M/M/1-PS queue, representing the always-on macrocell, and each small cell is characterized by an energy-aware M/M/1-PS queue with a sleep state and setup delay. As an additional control feature, the model of the small cells included an idle timer, which is used for controlling how long the small cell waits after the end of a busy period until it falls into sleep state.

By applying the theory of MDPs and the first step of the policy iteration method, we developed a near optimal policy for the performance-energy trade-off. Our main contribution was the derivation of the explicit forms of the value functions for the energy and performance, which yields the FPI policy. The explicit form of the FPI policy yielded insights to the general properties of the (near) optimal policy: the marginal performance cost has a JSQ-like linear dependence on the number of flows in the macrocell and small cell models and an additional constant cost in the small cell model reflecting the server state, while the marginal energy cost is constant both in the macrocell and small cell models. The FPI policy was initially derived by characterizing the performance-energy tradeoff as the weighted sum of performance and energy. We also showed how the same FPI policy can be applied through Lagrangian techniques in a constrained MDP setting, as well, where the objective is to minimize the energy subject to a performance constraint. In our numerical studies, we focused on the impact of the idle timer. In the case of the weighted sum objective function, the optimal timer value appears to be either zero or infinite. However, in the constrained optimization a finite idle timer can be optimal when the setup delay is short enough relative to the service times.

Possibilities for future research are many. On the algorithmic side, one research direction can be to seek for simple heuristic policies that achieve nearly the same performance as the FPI policy. Generalizations worth investigating include analyzing the impact of non-exponential service time distributions as well as interference between the base stations.

However, these extensions are analytically very difficult to handle but simulations can be used to this end.

ACKNOWLEDGEMENTS

This research has been partially supported by EIT Digital under the HII ACTIVE project and by the Academy of Finland under the ITTECH5G project (Grant No. 284735).

REFERENCES

- [1] M. F. Hossain, K. S. Munasinghe, and A. Jamalipour, "Distributed inter-BS cooperation aided energy efficient load balancing for cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 11, pp. 5929–5939, 2013.
- [2] 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.
- [3] J. Zheng, Y. Cai, X. Chen, R. Li, and H. Zhang, "Optimal base station sleeping in green cellular networks: A distributed cooperative framework based on game theory," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4391–4406, 2015.
- [4] S. Cai, Y. Che, L. Duan, J. Wang, S. Zhou, and R. Zhang, "Green 5G Heterogeneous Networks Through Dynamic Small-Cell Operation," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1103–1115, 2016.
- [5] A. Rangiseti, T. Pasca, and B. R. Tamma, "QoS Aware load balance in software defined LTE networks," *Computer Communications*, vol. 97, pp. 52–71, 2017.
- [6] G. Carvalho, I. Woungang, A. Anpalagan, and E. Hossain, "QoS-aware energy-efficient joint radio resource management in Multi-RAT heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6343–6365, 2016.
- [7] E. Khloussy, X. Gelabert, and Y. Jiang, "Investigation on MDP-based radio access technology selection in heterogeneous wireless networks," *Computer Networks*, vol. 91, pp. 57–67, 2015.
- [8] A. Roy and A. Karandikar, "Optimal radio access technology selection policy for LTE-WiFi network," in *Proc. of WiOpt*, May 2015, pp. 291–298.
- [9] Y. Song, P. Y. Kong, and Y. Han, "Potential of network energy saving through handover in HetNets," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10 198–10 204, 2016.
- [10] I. Taboada, S. Aalto, P. Lassila, and F. Liberal, "Delay- and energy-aware load balancing in ultra-dense heterogeneous 5G networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28:e3170, 2017.
- [11] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.
- [12] M. Gebrehiwot, S. Aalto, and P. Lassila, "Near-optimal policies for energy-aware task assignment in server farms," in *Proc. of CCGrid*, May 2017, pp. 1017–1026.
- [13] J. Leino and J. Virtamo, "Determining the moments of queue-length distribution of discriminatory processor-sharing systems with phase-type service requirements," in *Proc. of NGI*, May 2007, pp. 205–208.
- [14] P. Osti, P. Lassila, and S. Aalto, "Optimal intercell coordination for multiple user classes with elastic traffic," in *Proc. of NGI*, June 2012, pp. 25–32.
- [15] E. Altman, *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.
- [16] F. J. Beutler and K. W. Ross, "Optimal policies for controlled Markov chains with a constraint," *Journal of Mathematical Analysis and Application*, vol. 112, pp. 236–252, 1985.
- [17] M. E. Gebrehiwot, S. Aalto, and P. Lassila, "Energy-performance trade-off for processor sharing queues with setup delay," *Operations Research Letters*, vol. 44, no. 1, pp. 101–106, 2016.
- [18] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, pp. 1155–1171, 2010.
- [19] C. Sun, E. Stevens-Navarro, V. Shah-Mansouri, and V. W. Wong, "A constrained MDP-based vertical handoff decision algorithm for 4G heterogeneous wireless networks," *Wireless Networks*, vol. 17, no. 4, pp. 1063–1081, May 2011.