Hindawi
Mobile Information Systems
Volume 2017, Article ID 2495934, 17 pages
https://doi.org/10.1155/2017/2495934

Research Article

A Practical Framework for Energy-Efficient Node Activation in Heterogeneous LTE Networks

Antonio Virdis,[1] Giovanni Stea,[1] Dario Sabella,[2] and Marco Caretti[3]

[1]*Dipartimento di Ingegneria dell'Informazione, University of Pisa, Pisa, Italy*
[2]*Intel Deutschland GmbH, Munich, Germany*
[3]*Telecom Italia, Turin, Italy*

Correspondence should be addressed to Antonio Virdis; a.virdis@iet.unipi.it

Received 31 January 2017; Revised 11 April 2017; Accepted 27 April 2017; Published 7 June 2017

Academic Editor: Massimo Condoluci

This paper presents a framework to activate and deactivate micronodes in a heterogeneous multicell LTE network, based on load and energy efficiency consideration. The framework exploits historical data (i.e., per-macro-cell load curves) to select a set of candidate switch-on/switch-off instants of microcells, assuming a *limited* number of state changes is allowed in a day. The switching instants are instead determined *online*, by taking into account the actual traffic as well as the load curves. Moreover, intercell interference is fully accounted for. Our simulations show that this framework allows a multicell network to sustain peak-hour load when necessary and to reconfigure to a minimum coverage baseline whenever feasible, thus saving power (up to 25% in our scenarios). Moreover, the framework is *robust*, meaning that deviations of the actual traffic with respect to the prediction offered by the load curves can easily be handled.

1. Introduction

Computer networks have become a major contributor to electricity consumption all over the world [1], which has fostered a large amount of research on how to make them more energy-efficient. This problem can be targeted from several, concurrent points of view: on the one hand, working at the hardware level to make network equipment more energy efficient; on the other hand, designing algorithms that exploit the inherent *redundancy* of computer networks and temporarily switch off unused or unnecessary portions of the networks during off-peak hours. This allows networks to be designed for peak, while operating at lower-than-peak power most of the time. As far as cellular networks are concerned, energy efficiency in LTE networks has been the subject of several studies. Leaving aside those that investigate battery saving at the *user* side (e.g., [2–4]), many works concentrate on the *infrastructure* side, which is where energy consumption meets OPEX (see, e.g., [5–9] and the references therein and again [10–18]).

LTE power saving schemes exploit the fact that cellular coverage is *overlapping*, due to both reliability and (mostly) performance design issues and that traffic load is *highly variable*. Coverage is designed to carry *peak-hour* loads, which normally occur during working hours in business areas, while nodes use very few resources during (long) off-peak periods, for example, at night. Therefore, some nodes can switch off during off-peak hours, and nearby nodes will increase their transmission power accordingly to guarantee coverage. When the load is high, the first node will be reactivated and the second one will revert to normal radius. A similar concept can be applied to *heterogeneous* cellular network deployments, where a *macro* node presides over a relatively large area and *micro* nodes provide additional localized capacity to cover hotspots within the former. Micronodes can be switched on to offload the macro node (which is supposed to remain always on, lest coverage holes appear) when necessary or profitable from an energy efficiency point of view.

Switch-off schemes—for both macro nodes, in homogeneous settings, and micronodes in heterogeneous settings—can be either *offline* or *online* [5]. Offline schemes rely on knowing the evolutions of load in time (the so-called *load curves,* e.g., [19]) and decide the optimal switch-on/switch-off intervals based on the shape of these curves. They are often

used for resource provisioning purposes and, being based on *load predictions*, which may always prove wrong, need an *online* counterpart to react to unforeseen circumstances (flash crowds, accidents, etc.).

Online algorithms look at the *current* load and decide when and where it is high enough to warrant a switch-on (or low enough for a switch-off) of some nodes. Online schemes normally work at coarse timescales, for example, 15-minute slots, because switching operations are not instantaneous (in fact, they may take up to minutes [10]). Moreover, the ensuing topology alterations may trigger avalanche effects, such as massive handovers, or cause interference ripples in neighboring cells, which are difficult to predict and manage. For instance, switching off a micro node at location x may indeed increase the load in *nearby* areas, because the macro node at location x takes over its load and hence generates more interference around itself.

Most works in the existing literature usually make some assumptions which render their scheme impractical: the first one is that the number of per-node switch operations is *unlimited*. While we can expect future base stations to be able to tolerate frequent power transitions, most of the ones that are currently operational have not been designed with this characteristic [6]. This means that frequent switches will decrease their energy efficiency and possibly decrease their mean time between failures (MTBF), something that has a significant impact on OPEX. This calls for algorithms that can maximize the power saved by the network while limiting the number of power transitions for the node to a predefined maximum. Another frequent assumption is to rely on a particular *shape* of the load curves (e.g., one with a clearly distinguishable peak-hour and off-peak valley). While this allows one to build an optimal algorithm, the same algorithm would make suboptimal decisions when the shape of the load curve is different. Several algorithms assume *clairvoyance*: for instance, they assume that they know the exact amount of resources that a micro, currently switched off, will use to serve a given input traffic. This cannot be known in advance in practice and depends on the user positions and their perceived interference. Finally, they often make *local* decisions neglecting their *global* effects: as anticipated, switching off a micro (based on local knowledge, e.g., few users connected to it) will save the network owner the power to operate it; however, the extra resources required to serve the same users at the macro will foster a higher interinterference in the surrounding area, whose net effect may well be to increase the power consumption. Therefore, the *global* effects of switching decisions should be considered.

In this paper, we design and evaluate an *offline-online* framework for heterogeneous cellular networks, whose job is to satisfy a given load demand with the minimum possible energy cost. Our framework exploits offline information, that is, load curves, to select which configurations are *feasible* (i.e., able to carry the nominal load) at a given point in time. An optimization algorithm is then run on the set of feasible configurations, to decide the optimal switch-on points for the micro nodes, constrained to a *maximum number of switch-ons* in a day. To the best of our knowledge, ours is the only work that takes the latter aspect into account. The only work we know of that considers the number of switch-ons, that is, [20], seeking to minimize the latter rather than taking it as a constraint. The *online* part of the framework, instead, measures the current load and *enforces* switch-on/off of micro nodes so as to satisfy the demand in an energy-efficient way. More specifically, it may bring forward or postpone switching instants with respect to the optimal ones computed offline, when it is more energetically efficient to do so. The scheme described in this paper is designed so as to require few hypotheses: it can work with load curves of arbitrary shape, it does not assume clairvoyance, and it takes into account global effects of switching decisions through an accurate estimation of intercell interference. We evaluate our scheme using both synthetic and real-life load curves taken from operator data. Our results show that power savings up to 25% are achievable over a whole day and that neglecting intercell interference leads to overoptimistic results. Moreover, we show that our scheme is *robust* against variation in the expected load.

The rest of the paper is organized as follows: in Section 2 we describe our assumptions and state our problem formally. The related work is discussed in detail in Section 3. Section 4 describes the proposed framework, which is evaluated in Section 5. Finally, Section 6 concludes the paper.

2. System Model and Problem Statement

In LTE, cell transmissions are arranged in time slots called Transmission Time Intervals, (TTIs), whose duration is 1 ms. In a TTI, the node allocates *frames,* that is, vectors of *(Virtual) Resource Blocks* (RBs) to its associated user equipment (UE) [21]. UE is associated with *one* node at a time. Each RB carries a fixed number of symbols, which translate to different amounts of bits depending on the *modulation and coding scheme* (MCS) used by the antenna on that RB. In general, more information-dense modulations (e.g., 64QAM, yielding 6 bits per symbol) are favored when a better channel to the UE is perceived (i.e., one with less interference). The quality of the wireless channel varies over both time and frequency. For this reason, UE reports their perceived downlink channel state to the eNodeB as a Channel Quality Indicator (CQI), computed according to the measured Signal to Interference and Noise Ratio (SINR).

We consider a large-scale multicell LTE network as shown in Figure 1. The network is logically divided into N areas (called *hexagons* henceforth for simplicity), each one under the coverage of a *macro node* (M in the figure), that is, an eNodeB with high radiation capabilities. Macro nodes are anisotropic, as shown in Figure 1. A hexagon h can also accommodate L_h *micro nodes* (m), $1 \leq h \leq N$, $L_h = 2$, in the figure. The latter, usually located in hotspots or at hexagon edge, can be used to provide additional localized capacity at a lower power cost. The term "micro" will henceforth be used in a broad sense, to denote nodes with smaller power and coverage, thus encompassing everything from a micro to possibly femto, with the following provisos: micro nodes alone *cannot* provide full coverage of a hexagon (not even jointly), and all nodes are owned by the same entity (i.e., a telecommunication operator), which pays the electrical bill. Either macro or micro nodes are used to serve UE

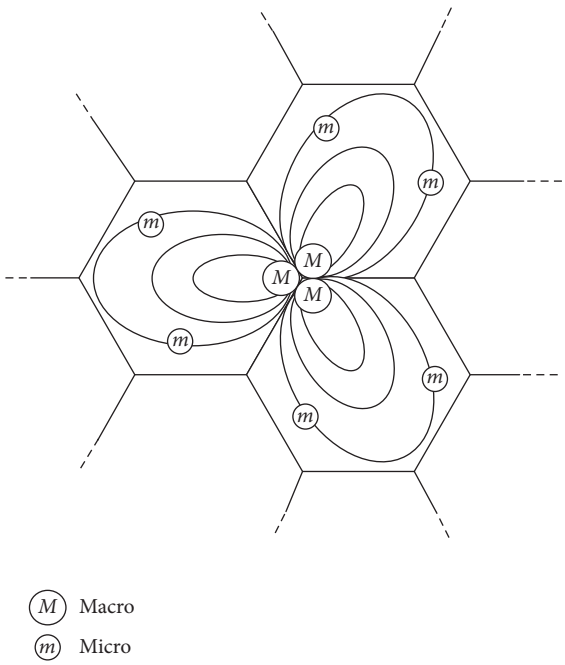

Figure 1: System scenario. Macro and micro nodes are represented with M and m, respectively.

within the network. UE associates with the node from which they perceive the highest SINR, possibly using a Cell Range Expansion (CRE) boost for the micros.

All the nodes share the same spectrum and hence interfere with each other. We assume *no coordination*, either among or within hexagons. A node that has to allocate RBs to its users will pick them at random from a vector of B RBs and schedule each UE on each RB with the same probability. Cell coordination and Coordinated Multipoint (CoMP) techniques are a promising avenue of research to cope with interference and have been shown to increase the power saving opportunities [2]. However, they work at a significantly faster pace than algorithms for node switching and are thus orthogonal to them.

We assume that the operator possesses the following information: a per hexagon historical *load curve*, detailing (at least) the overall *bitrate requested in a hexagon* over time in a sample day. There may be several such curves, of course, for a single hexagon, depending on the *context*. For instance, working days and holidays have different load curves (see, e.g., [19]), and other human activities do affect the context: the typical example is an hexagon covering a major sports venue, whose load curve on the game days (e.g., Sundays) depends on whether the local sports team plays home or away. Moreover, the operator possesses a similar curve (or set thereof) detailing the number of users per cell over time, so that the average *per-user bitrate* can be inferred (a possible method to obtain such information is using *monitoring systems* that measure the values of specific KPIs (e.g., number of types of UE) with a fixed time resolution [22]. These measurements are repeated in different days with the same operating conditions (e.g., working days) in order to produce a daily profile of the average values for the given KPI.). Note that these curves do not contain information about the *position* of single users within a hexagon. The latter may be known in a *probabilistic* sense, for example, uniform within a hexagon. We make no assumption on the *shape* of the load curves, and—in particular—we do not rely on their having a single peak during business hours. The time resolution at which these curves are plotted is that of a sufficiently large interval (e.g., 15 minutes), called a *snapshot* henceforth. We denote with K the number of such snapshots in a day. Furthermore, we assume that we can retrieve the *current* amount of allocated RBs at each macro/micro node, averaged over a snapshot.

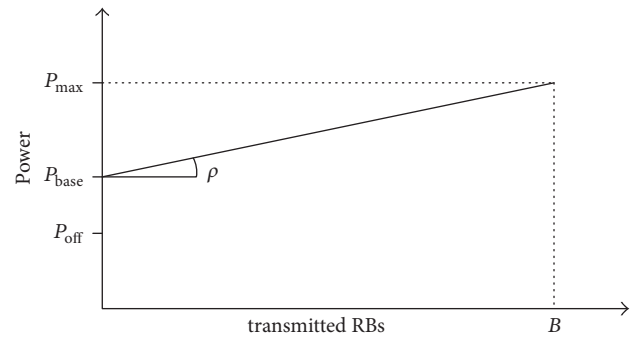

Figure 2: Node power model.

We only consider the downlink direction, which is the most critical from the point of view of both carried load and infrastructure power consumption. Figure 2 shows the power consumed by an active node (taken from [23]). The power is an affine function of the number of transmitted RBs on each TTI, that is, $p = P_{base} + \rho \cdot n$, where P_{base} is a *baseline* power and $n \leq B$ is the number of allocated RBs. The power/RB curve has the same shape for all the nodes, whether macro or micro, whereas the values P_{off}, P_{base}, P_{max}, and ρ do depend on the type of node. According to equipment manufacturers and cellular operators, the value of the above parameters has changed over the years thanks to the benefits introduced by new technologies and will keep changing, following the progressive shift towards the fifth generation of mobile networks (5G). However, the affine shape of the power/RB curve has remained unchanged, and such trend is expected to persist [23].

A *macro* node cannot be powered off and hence, the *variable* part of its consumed power is $p = \rho \cdot n$. For a *micro* node, the variable part is instead $p = (P_{base} - P_{off}) + \rho \cdot n$.

Since load curves represent what happens in a snapshot, whereas power is consumed by allocating RBs at each TTI, a method to infer the latter from a snapshot's load value is required. RBs are computed as follows: first we infer a per-user requested data rate from the load curves, then we show how to compute the number of RBs needed to meet that data rate given the UE's SINR, and then we show how to compute the SINR of a UE undergoing interference using a statistical interference model.

Focus on a single snapshot and a single hexagon, and call L and U the load and number of users for that hexagon

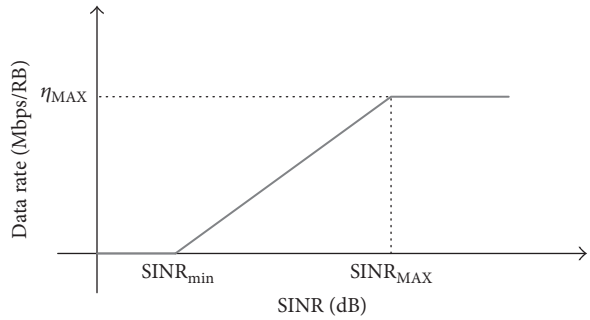

Figure 3: Data rate versus SINR.

at that snapshot. The *average demand* per UE at a snapshot is $D = L/U$. The *data rate per RB* achievable by UE u associated with node e is a function of its *average* SINR, $F(\mathrm{SINR}_u^e)$, an example of which is represented in Figure 3 (curves like the one in Figure 3 are obtained through link-level simulations, see, e.g., [24]). η_{MAX} is the maximum data rate that can be achieved for values of SINR equal or above $\mathrm{SINR}_{\mathrm{MAX}}$. UE whose SINR is below $\mathrm{SINR}_{\mathrm{min}}$ is considered out of range. Thus, the average number of RBs required to satisfy u's demand can be computed as

$$\mathrm{RB}_u = \frac{D}{F\left(\mathrm{SINR}_u^e\right)}. \tag{1}$$

Note that RB_u may not be integer. This is not a problem, since RB_u is an *average* value, and the time span of a snapshot is large enough as to allow a fluid approximation. For instance, an allocation of $\mathrm{RB}_u = 2.5$ RBs per TTI can be enforced in practice by reserving two RBs on even TTIs and three on odd ones and so on.

In order to compute SINR_u^e, we need to quantify both the signal received from node e and the interference from nodes $x \neq e$. Call n_e and n_x the number of RBs allocated by nodes e and x, respectively. Given that they do not coordinate, the probability that the same j RBs are allocated *simultaneously* by both is [25]

$$\pi_{e,x,j} = \frac{\binom{B}{j} \cdot \binom{B-j}{n_e-j} \cdot \binom{B-n_e}{n_x-j}}{\binom{B}{n_e} \cdot \binom{B}{n_x}} = \frac{\binom{n_e}{j} \cdot \binom{B-n_e}{n_x-j}}{\binom{B}{n_x}}, \tag{2}$$

with $\max(0, n_e + n_x - B) \leq j \leq \min(n_e, n_x)$. Therefore, the expected value of the number of overlapping RBs between e and x is

$$\Delta_{e,x} = \sum_{j=\max(0,n_e+n_x-B)}^{\min(n_e,n_x)} j \cdot \pi_{e,x,j}. \tag{3}$$

Thus, the percentage of RBs where x will interfere is $\Delta_{e,x}/n_e$, whereas the remaining $1 - \Delta_{e,x}/n_e$ will be exempt from interference from x. Call $P_{x,u}$ the power received by u from node x (which depends on the distance and angle between them, the propagation model and the transmitting power of x). Then, the *average* SINR of UE u in the snapshot is

$$\mathrm{SINR}_u^e = \frac{P_{e,u}}{N_G + \sum_{x \neq e} P_{x,u} \cdot \Delta_{e,x}/n_e}, \tag{4}$$

where N_G is the Gaussian noise.

As far as switching is concerned, as already specified, macro nodes cannot be powered off, because this would create holes in the coverage. Micro nodes, instead, *can* be powered off, for example, during low-load periods. However, switch-on/off operations should be *few*, for several reasons: first, especially with older equipment, it takes a considerable amount of time before a micro node is fully operational (in the order of tens of seconds or minutes), due to the operating system boot procedure, control and management plane setup operations (e.g., path setup in the Evolved Packet Core network), and so on, and it takes some time to shut it down as well. Second, topology alterations often have unpredictable ripples in a multicell network: a switch-off may cause massive handovers (even when *cell wilting* is used, [6]), possibly overloading the corresponding macro *and* affecting nearby macros through a change in the interference pattern. Third, switching operations increase the wear and tear of equipment, hence reducing its MTBF [6]. For this reason, we assume that an upper bound of $2 \cdot S_{\max}$ switching operations must be enforced on all micro nodes. In other words, micro nodes can be powered up in *at most* $S_{\max}$ *disjoint* time intervals within a day.

The aim of our framework is to *plan* and *enforce* switching operations at each snapshot for each micro node, so that

(1) user requests are satisfied, if it is possible to do so. In other words, no *overload* occurs that prevents UE from obtaining the required rate;

(2) as little energy as possible is consumed by the network, provided that UE requirement is met;

(3) micro nodes are switched on up to $S_{\max}$ times per day.

Our framework *plans* switching operations in advance (i.e., offline), based on historical load curves, so as to keep the macro overload probability below a predetermined threshold, and *enforces* switching operations *online* either to avoid overload or when it is energetically efficient to do so, by comparing the current load conditions to the expected (historical) ones.

3. Related Work

As already outlined, several schemes have been proposed in the past few years to deal with macro and micro node deactivation. Surveys [5–9] provide a general overview and taxonomy, besides the customary wealth of references.

In the context of homogeneous networks, many works advocate cell *breathing* (e.g., [11]): the radius of some cells is shrunk during off-peak hours, and nearby cells increase theirs accordingly to guarantee coverage. If the scenario allows it, a node can be completely switched off, which increases the saving. Cell *wilting* and *blossoming* are procedures for graceful shut-down and power-up of nodes, whose aim is to avoid massive handovers during topology changes [12].

It must be observed that operators often consider hazardous to vary the cell radius, because it makes it difficult, at best, to ensure the absence of coverage holes [6]. Moreover, most of the current equipment can only be powered on or off, lacking the capability to switch among a discrete number

of power levels that is instead required in such schemes. Paper [13] models the energy saving sleep problem assuming a single macro and several small cells, with symmetric traffic, no interference, and "1:1 convertible" resource allocation between nodes. A larger-scale scenario (cluster of 7 macros, with no micro) is considered in [10], where cell breathing is advocated. Authors of [14] describe an offline algorithm that does not allow multiple switching of the same node in a day. In [15] a planning tool is described, which also takes cell breathing into account. A similar problem is tackled in [26], where authors focus on positioning of both macro and micro stations to achieve more capacity at a small cost in power consumption. Finally, [16] determines the optimal set of cells to be switched at any time t, given the instantaneous load at that time.

Among the above works, those advocating offline schemes often make assumptions on the shape of load curves, meaning that they *only* work when the load curves match those envisaged. Our work does not make such an assumption, and can work with *any* shape of load curves. Moreover, these works seldom consider intercell interference, which—as shown in the previous section—clearly influences the switching decisions. None of the above works considers (or may easily incorporate) constraining the number of switching decisions at a node, something which—as we will demonstrate—changes the picture completely.

Finally, we observe that the problem of energy-efficient resource allocation in LTE base stations can be tackled at different timescales. Besides the *resource planning* timescale (i.e., hours or days), which is the one of all the above works, we can also work at the *frame timescale*, that is, few tens of ms, for example, by scheduling MBSFN subframes in a frame [27]. During off-peak periods, even short ones, the eNB pools the few RBs that it has to transmit into as few subframes as possible, leaving the other ones blank, thus trading a modest reduction of responsiveness for a remarkable power saving. These solutions are complementary, rather than alternative, to the ones dealt with in this paper.

4. Node-Switching Framework

We first explain some of the practical challenges for a switching algorithm, which also sets the rationale behind our framework. Then we describe the framework at a high level and each of its components in detail.

We will use the shorthand m to indicate that *all* the micros in a group of hexagons are switched on and – to indicate that they are all off. We will often distinguish two groups of hexagons: a *tagged* one, whose performance we monitor, and the rest of the network. Accordingly, $m \mid -$ will indicate that micros are switched *on* in the tagged hexagon and *off* in the other ones.

4.1. Justification and Practical Challenges. As already explained, we consider two performance metrics: overload and energy consumption. We first assess the impact of a micro switch-on/switch-off decision on both metrics.

The interference of a *micro* is, by and large, negligible *outside* its own hexagon. This means that, if switching *on* a micro succeeds in offloading the macro of some RBs, then the effects on overload *at large*, that is, in the whole network, can only be beneficial, since the intercell interference which is mainly due to the macro will be reduced through (4). Unless micros are misplaced, it is usually true that switching on a micro offloads the macro of some RBs and that the overload probability in a hexagon *decreases* when the micros are switched on. This means that switching micros *on* when a hexagon is in overload are the right thing to do. In order to demonstrate these two phenomena let us consider the following example: suppose we are monitoring the channel quality perceived by the UE in hexagon h_0 placed at the center of a scenario composed of other nine hexagons, each of them with one macro and two micro nodes. We define four configurations: a first one $(- \mid -)$ with all the micros turned off, a second one $(m \mid m)$ with all the micros turned on, a third one $(- \mid m)$ with all the micros turned on except within h_0, and a fourth one $(m \mid -)$ with the micro nodes turned on *only* in h_0. In Figure 4 we show the perceived channel quality in h_0: the color of each point represents the best SINR computed among the active nodes of the considered hexagon. As we can see if we compare $- \mid -$ and $- \mid m$, the SINR in h_0 slightly increases even if we just turn on micros in the other hexagons, thus confirming that interference from the latter has decreased. On the other hand, turning on micros improves the SINR of h_0 in both $m \mid m$ and $m \mid -$ scenarios, a benefit that comes along with the added capacity in terms of RBs that are made available by each active micro. An additional analysis of the large-scale effects of interference will be given in Section 5.

From an energy point of view, switching on a micro incurs a baseline penalty given to the difference $P^m_{\text{base}} - P^m_{\text{off}}$. However, RBs allocated at the micro are normally cheaper than those at the macro, since $\rho^m < \rho^M$, where subscripts and superscripts are self-explanatory. When a micro is switched on, and some UE switches from the macro to the micro, there is a global decrease of intercell interference through (4), which is also beneficial energy-wise. From the above, we obtain that keeping a micro on for very few allocated RBs is inefficient. However, it is *impossible* to estimate the minimum number Th^m of RBs that the micro should allocate to pay for its switch-on cost from *online* information only. In fact, that number can only be inferred by solving the following equation:

$$\left(P^m_{\text{base}} - P^m_{\text{off}}\right) + \rho^m \cdot n^m = \rho^M \cdot n^M, \tag{5}$$

where the left-hand side is the increase of power when a micro is switched on and allocates n^m RBs, and the right-hand side is the reduction of power because the macro allocates n^M fewer RBs. However, n^m and n^M depend on which kinds of UE will switch to the micro and what their SINR would be if they were associated with the micro. This is of course unknown, both because you do not know where this UE is located *and* because the micro is switched off, which creates a chicken-and-egg problem.

Similarly, it is unrealistic to assume that we can infer when it is energetically efficient to switch a micro *off* based on online data only. In fact, this would require knowing the SINR

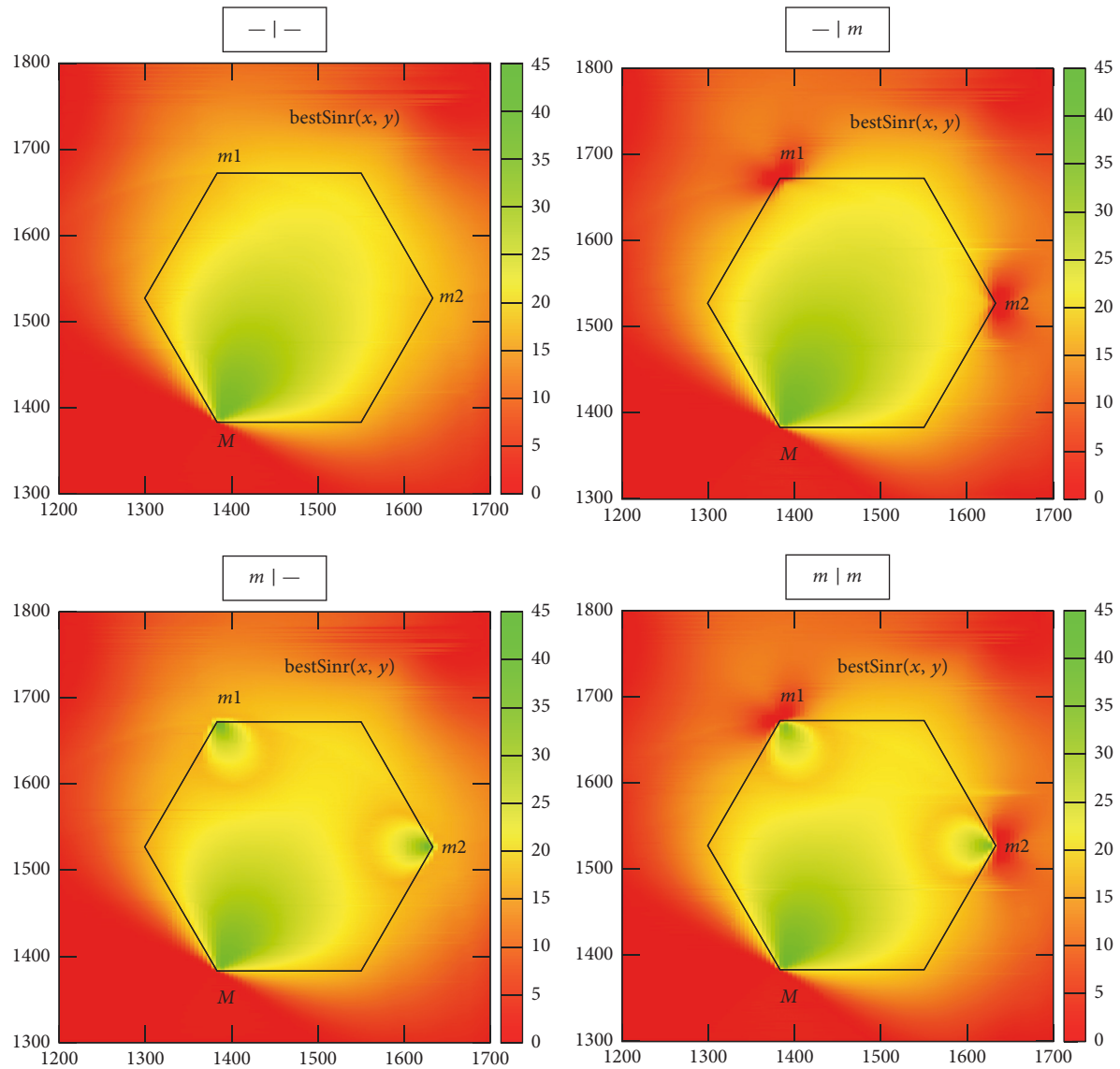

FIGURE 4: Channel quality perceived by UE in $h_0$when all micros are turned off (– | –), all micros are turned on (m | m), micros are turned on only in h_0 (m | –), and micros are all turned on except in h_0 (– | m).

of each UE to the macro *after* the micro has been switched off. The latter is not equal to the one measured *before* the switch-off, due to the mutated intra- and interhexagon interference.

Nevertheless, we can figure out an approximate threshold Th^m by *simulating* the network *offline* using load curves and space distributions. Thus, the insight gathered using offline simulations can surrogate the lack of SINR-omniscience required in an online-only approached.

Another challenge stems by the constraint on the number of switch-ons in a day. This implies that making the decision to switch *off* a micro based on the *current* traffic conditions only (in a purely online approach) is the wrong thing to do. This can be explained through an example.

Example 1. In Figure 5 we show an example of average per-snapshot RBs allocation over time in a scenario with one macro (M) and one micro node (m), constrained by a maximum number of switch-on operations $S_{\max} = 2$. Call n^M and n^m the number of RBs allocated by M and m, and Th^{sat} the macro saturation threshold. We start with m off. The macro saturates at t_1, triggering the activation of m. At t_2 node M is no longer in saturation, and the number of allocated RBs starts to decrease on both nodes, reaching value Th^m on the microside: judging by the allocated RBs only, we switch m off. The same occurs in $[t_4, t_6]$: m is activated when M saturates (t_4), so the second and last micro switch-ons are spent. This means that when the load decreases again later on, we cannot switch m off, even though n^m drops below Th^m, since doing so would prevent us from reacting to possible future saturation conditions. Micro m must therefore be kept active until t_7, where the value of n^m rises again above Th^m. Thus, we are able to cope with the saturation that occurs between

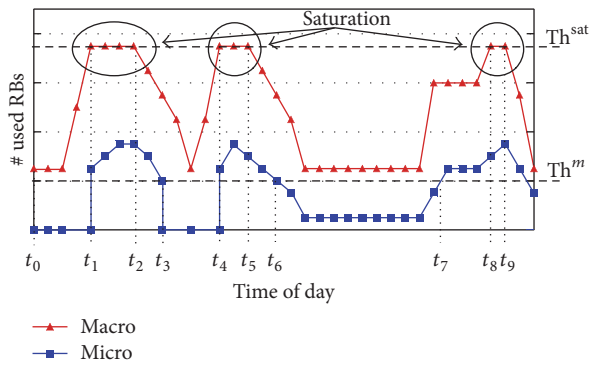

FIGURE 5: Resource block utilization Example 1.

t_8 and t_9. Note that, in the absence of switch-on constraints, the micro would be switched *off* at t_6 and *on* again at t_8.

Using the solution described above, we can react to overload conditions at the macro node and to inefficient utilization of micro ones, by just looking at the *instantaneous* allocation of RBs. Two considerations are in order.

If more energy was consumed in $[t_6, t_8]$ than is saved in $[t_3, t_4]$, it would be more efficient to switch the micro off in the former and keep it active in the latter.

In $[t_7, t_8]$, n^m grows above Th^{m} *before* n^M reaches Th^{sat}. In this case, we can exploit m efficiently, energy-wise, even though the macro node is not in saturation yet.

The above two observations can only be made with hindsight, hence are *precluded* to an online algorithm. However, we can again exploit *historical* knowledge, harvested from the daily load curves, to endow an online algorithm with prediction capabilities and increase its efficiency.

4.2. Description of the Framework. Our framework obtains historical information from the network, processes it *offline*, and makes switching decisions *online*, based on the current load and the historical knowledge. The framework and its information exchange with the network nodes are shown in Figure 6. The offline part of the framework is composed of a *trainer* and an *Offline Analyzer*. The trainer takes as input the historical load curves (detailing the requested bitrate and number of users per hexagon) and *simulates* what happens at each snapshot when micros are switched on and off, respectively. The outputs of this simulation are, for each snapshot, an *overload probability*, that is, the probability that a given hexagon is unable to carry the requested load when the micros are switched off, and a *resource consumption*, that is, the number of RBs allocated within a hexagon when the micros are turned on and off, respectively (note that the trainer could also be realized in different way, which may not involve simulation at all, e.g., as a machine learning system trained with a large set of measurements from the network in various configurations. As long as the trainer is able to compute overload probabilities and resource consumption, the rest of our framework will be able to work with it).

The Offline Analyzer takes as input the above information, plus the power models of the macros and micros and

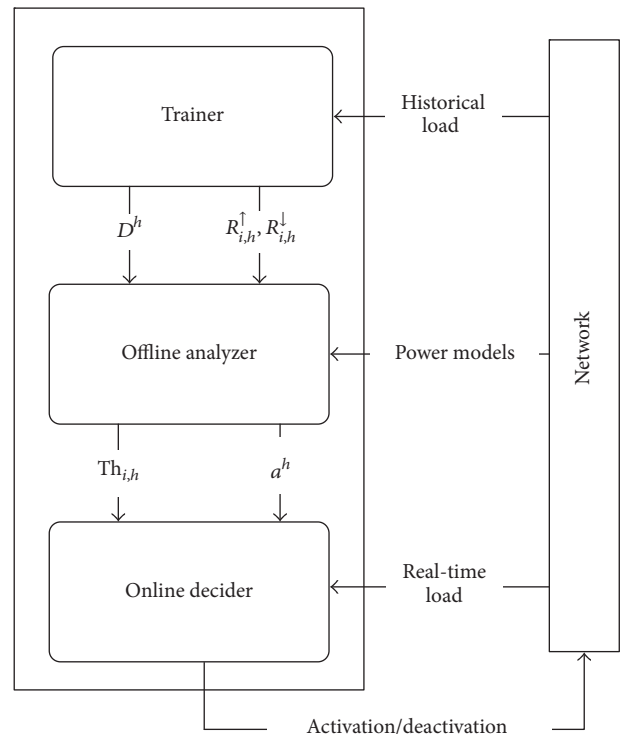

FIGURE 6: High-level view of the switching framework.

the constraint on the number of switch-ons, and finds the *energy-optimal activation pattern* for each hexagon, that is, the intervals where switching micros off allows the highest power saving, given that the requested load is satisfied.

The above information, plus the current load at the hexagons, is used by the *Online Decider* to enforce switching decisions. Hereafter, we present the three blocks separately.

4.2.1. Trainer. The trainer considers one snapshot at a time. In a snapshot, it *simulates* the network with the given number of users per hexagon, their spatial distribution, and requested bitrate, in a *scenario*. A *scenario* is defined by the number of nodes that is switched on: for instance, a *micro-off* scenario is one where all micros are switched off, and a *micro-on* scenario is the one where all micros are switched on.

The simulation is done as follows: users are dropped in the hexagon according to the spatial distribution. Figure 7 shows a uniform (a) and hotspot-based (b) spatial distribution. In the latter, UE is dropped in the portion of hexagon that intersects a circle of center c and radius r. Note that spatial distributions are used *only* by the trainer, while the *Online Decider* totally does without them. At a snapshot boundary, UE is created or deleted to match the required number, and the position of the remaining ones is randomly changed within a configured area. This allows us to simulate random mobility within the cell or parts thereof.

UE is associated *sequentially*, that is, one by one, to the active node with the highest SINR, with a CRE boost for the micros.

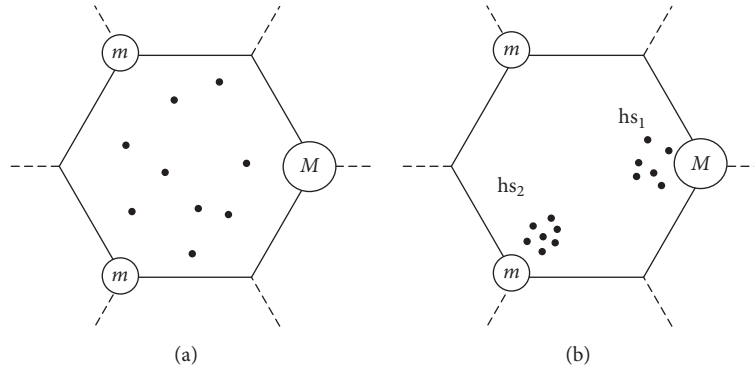

Figure 7: Uniform (a) and hotspot (b) users distribution in an exemplary hexagon. In the latter case two hotspots, hs_1 and hs_2, are shown.

```
(1) iteration = 0
(2) While (interfChanges or iteration > Itmax)
(3)     resetAllocation()
(4)     for each hexagon i
(5)         for each UE u in i
(6)             if(itearation < It) // Assoc. Ph.
(7)                 k = chooseEnb(u)
(8)             else                // Conv. Ph.
(9)                 k = getServingEnb(u)
(10)            allocateBlocks(u,k)
(11)        end for
(12)    end for
(13)    interfChanges = updateInterference()
(14)    iteration++;
(15) end while
```

Algorithm 1: Pseudocode for the *node association* and *RB allocation* procedure.

The average SINR of each UE is computed through (4). Note that there is a circular dependence between the SINR and the RBs allocated to a UE. When a UE is allocated some RBs (say, in hexagon e), in fact, $\Delta_{e,x}$ grows, thus increasing the interference suffered by UE attached to x and reducing their SINR. This in turn increases their RB demand and so on. This means that the average SINR must be computed iteratively, until convergence is reached. The algorithm that does this is shown in Algorithm 1.

The procedure is a cycle repeated for up to It_{max} iterations or until convergence is reached. Each iteration cycles through every UE. We distinguish two phases.

(i) Association Phase. For the first *It* iterations (lines 6-7) UE is allowed to *select* the serving node, according to a best SINR policy (line 7). While doing so, the procedure also allocates RBs according to (1) (line 10).

(ii) Convergence Phase. For all the remaining iterations, UE does not change their serving node, and the allocated RBs are updated (line 10).

In both cases, at the end of an iteration the interference is updated (line 13) through (3). If any $\Delta_{e,x}$ changes significantly from one iteration to the next, the *interfChanges* flag is set, to signal that the convergence has not been reached yet. Both Prx_u^e and $P_{x,u}$ from (4) are computed using the ITU pathloss and angular attenuation models [28].

Note that, at the first iteration of the *Association Phase*, no RBs have yet been allocated at any node, hence the interference is null and the association is distance-based rather than SINR-based. After the first iteration the interference is updated, hence the nearest node may not be the one with the best SINR anymore. This is the reason why the *Association Phase* is repeated *It* times. However, *It* has to be limited; otherwise some UE will end up oscillating indefinitely between two or more nodes. Figure 8 shows how the SINR decreases with the iterations, quickly converging to a stable value.

We say that a hexagon, during snapshot k, is *overloaded* in a scenario if it cannot serve its requested traffic. This means that all the resources of the macro (and possibly of the micros, if they are switched on) are depleted. Our first concern is to ensure that the network is able to satisfy the requested load. Therefore, we need to take note of overloaded hexagons. Moreover, we need to assess whether a scenario is more energy-efficient than another, hence to compute the number of allocated RBs at all active nodes, which contribute to the variable part of the power.

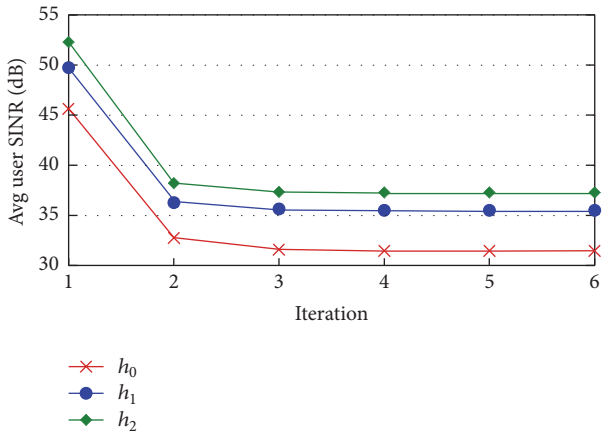

FIGURE 8: Evolution of the average user SINR during iterative interference evaluation of three hexagons in a snapshot.

Obviously, both the above depend on the way UE has been dropped. We recall that UE dropping is random, according to a preestablished distribution. Therefore, the trainer repeats the simulation of the same scenario for r times, dropping UE independently at every replica. This way, we obtain an *overload probability* D_h for each hexagon h in the considered scenario, given by

$$D_h = \frac{1}{r} \cdot \sum_{i=1}^{r} 1_{\{h \text{ overloaded in replica } i\}}. \tag{6}$$

Moreover, we obtain a vector of *average allocated RBs*, whose elements are the RBs allocated at each node in the hexagon, averaged over the n replicas:

$$R_{h,e} = \frac{1}{r} \cdot \sum_{i=1}^{r} R_{h,e}^{i}, \tag{7}$$

where e is the identifier of the node within hexagon h.

The above two quantities can be used to compare scenarios, specifically to identify the minimum-power one having the overload probabilities below a given threshold. Unfortunately, an exhaustive comparison of scenarios is impossible. This would in fact require simulating as many scenarios as the powerset of the set of micros. In a network with few tens hexagons, micros are easily in the hundreds, which makes the task impossible. We therefore settle for a simple trade-off, which proves to be effective nonetheless. We simulate only the two limit scenarios with all micros switched *on* and switched *off*. We denote these two scenarios using subscripts $\uparrow$ and $\downarrow$. The rationale beyond this choice is that the two configurations are expected to achieve, respectively,

(i) the minimum overload probability (which should be null, unless the network is underprovisioned), but a high power consumption;

(ii) a larger overload probability, with a considerably smaller power consumption.

All the quantities defined above are computed for each snapshot, thus values at snapshot k will be represented as $D_h(k)$ and $R_{h,e}(k)$.

4.2.2. Offline Analyzer. Values D_h, $\mathrm{RB}_{\uparrow}^{e,h}$, and $\mathrm{RB}_{\downarrow}^{e,h}$, and the node power models are then passed to a module called *Offline Analyzer* (OA), whose purpose is to process this information and to generate guidelines for online decisions. The OA makes independent decisions for each hexagon. It compares $D_h(k)$ to a predefined overload probability threshold (which may also depend on the snapshot and hexagon) and computes a binary value $O_h(k)$ that marks whether the overload probability is above or below the threshold. First it computes the expected power consumption of each of hexagons $E_{\uparrow}^{h}(k)$ and $E_{\downarrow}^{h}(k)$ in the two configurations as

$$\begin{aligned} E_{\downarrow}^{h}(k) &= P_{\text{base}}^{M,h} + \mathrm{RB}_{\downarrow}^{M,h}(k) \cdot P_{\text{in}}^{M,h} + \sum_{m=0}^{L_h} P_{\text{off}}^{m,h}, \\ E_{\uparrow}^{h}(k) &= P_{\text{base}}^{M,h} + \mathrm{RB}_{\uparrow}^{M,h}(k) \cdot \rho^{M,h} \\ &\quad + \sum_{m=0}^{L_h} P_{\text{base}}^{m,h} + \mathrm{RB}_{\uparrow}^{m,h}(k) \cdot \rho^{m,h}, \end{aligned} \tag{8}$$

where L_h is the number of micros in hexagon h.

Note that each node may have its own power model parameters, thus accounting, for example, for different versions of the apparatus and different cost of energy. The same applies to the switching-off thresholds. However, for the sake of readability, we will assume that all the power models are the same and drop the superscript whenever this does not generate ambiguity. For instance, we will use Th^{m} to denote the switch-off threshold.

We can then obtain an activation pattern for micro nodes in a hexagon during a day, as a vector of K binary values $a^h(k)$, where $a^h(k) = 0$ indicates that all micros in the hexagon are switched off and $a^h(k) = 1$ if they are all on. We compute the energy-optimal pattern separately for each hexagon as the optimum of the following problem, where hexagon superscript h is omitted for the sake of readability:

$$\begin{aligned} \min \quad & \sum_{k} E_{\downarrow}(k) \cdot (1 - a(k)) + E_{\uparrow}(k) \cdot a(k) \\ \text{s.t.} \quad & \text{(i)} \ a(k) \geq O(k) \\ & \text{(ii)} \ s(k) \geq a(k) - a(k-1) \\ & \text{(iii)} \ s(0) \geq a(0) - a(K-1) \\ & \text{(iv)} \ \sum_{k=0 \cdots K} s(k) \leq S_{\max} \\ & \text{(v)} \ s(k), \ a(k) \in \{0, 1\}. \end{aligned} \tag{9}$$

$s(k)$ is a binary variable that counts switch-on operations. The objective function to be minimized is the expected overall power consumed by a cell throughout the whole day. Constraint (i) states that micros should be on when saturation may occur, (ii) forces $s(k)$ to one when the micro is switched on at k, (iii) takes care of the wrap-around at the end of the day, and finally (iv) constrains the number of switch-ons to $S_{\max}$. The above is an Integer-Linear Problem (ILP) with $\Theta(K)$ constraints and variables and can be easily solved using commercial solvers such as CPLEX [29].

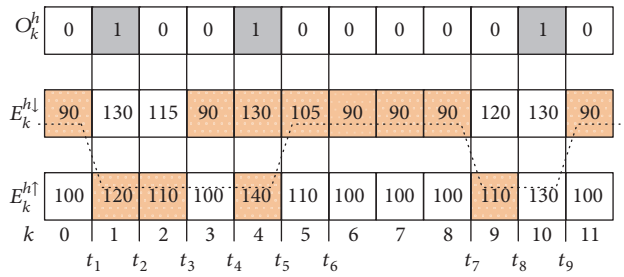

Figure 9: Example of energy-optimal pattern computation.

In Figure 9, we show an example of computation of the energy-optimal activation pattern, based on the same scenario of Figure 5. Each vector represents the values of $O^h(k)$, $E^h_\downarrow(k)$, and $E^h_\uparrow(k)$, respectively, throughout a day of $K = 12$ snapshots within hexagon h. Marked elements in the second and third vectors identify the lowest-power configuration (*all* off or *all* on) at that snapshot. Note that the same time instants t_i as the ones in Figure 5 are reported. The dashed line indicates the pattern that will be chosen by our optimization problem if $S_{\max} = 2$. As we can see we have three snapshots where $O^h(k) = 1$, where keeping micros on is thus mandatory. The optimal solution is to keep the micros on in $[t_3, t_4]$ and switch them off in $[t_5, t_7]$ as the power saved in snapshot three is $E^h_\uparrow(3) - E^h_\downarrow(3) = 10$, whereas in snapshots 5 to 8 is $\sum_{k=5}^{8} E^h_\uparrow(k) - E^h_\downarrow(k) = 35$. Moreover, note that, as observed in Example 1, the optimization problem brings forward the second switch-on event to t_7.

4.2.3. Online Decider. The output of the OA is fed to the *Online Decider* that monitors the system utilization and takes decisions on a per-snapshot timescale. More specifically this module decides which micro to switch on/off, based on *both* current and historical information. It includes an *activation* and *deactivation* procedure, performed sequentially. Figure 10 reports the flow diagrams for such procedures. Each box represents a processing step, whereas each diamond represents a decision. Deactivation takes as an input a list of micro nodes, ordered by increasing load. Each of those micro nodes m is switched off if *all* the following conditions hold simultaneously at snapshot k:

(i) no node in the hexagon is overloaded,

(ii) $a^h(k) = 0$ for the given hexagon,

(iii) m is underloaded, that is, $\mathrm{RB}^m \leq \mathrm{Th}^m$.

Conversely, we activate all the micros of a hexagon if *either of or both* the following conditions hold:

(i) at least one node in the hexagon is overloaded;

(ii) $a^h(k) = 1$.

Example 2. Consider as an example the RB utilization shown in Figure 11 where a hexagon with one macro and one micro is considered and the energy-optimal pattern of Figure 9 is applied. At t_1 the *activation* procedure detects saturation and switches the micro on. At t_3, it is kept on since $a^h(k) = 1$. The first switch-off opportunity occurs at t_5: in this case the micro is kept on until $\mathrm{RB}^m < \mathrm{Th}^m$, that is, at t_6. The last switch on is at t_7, as in the energy-optimal pattern.

4.3. Implementation Issues. The software framework described in this paper is meant to be run at a centralized element, which performs resource planning. The location and nature of this resource planner depend on the operator choices for RAN deployment. In a Distributed RAN, it should be a monitoring server in the operator's network. In a Centralized RAN (C-RAN), it would naturally be a process running in the same data center that hosts the virtualized Baseband Units (BBUs) of the nodes in an area. A software system that supports a decision framework such as the one described in this paper is being designed and deployed in a testbed as part of the Flex5Gware EU 5G-PPP project [30].

Regarding the processing cost, our measurements show that a single-day run (i.e., 96 snapshots) of the trainer occupies approximately 1.5 minutes of a single core on a desktop PC. Solving the optimization problem at optimality takes less than one second and hence is negligible. The number of runs should be calibrated based on the required overload probability threshold: more runs allow one to estimate small probabilities with greater accuracy. Now, the planning normally occurs on a daily basis, and a single core can run the trainer up to ~950 times per day. This means that our framework can be made to work with arbitrary accuracy on off-the-shelf hardware in a day. More to the point, this very fact can also be exploited to compensate for unforeseen context mismatches. Suppose that an *unforeseen* event occurs, such as a large traffic jam, which *predictably* changes the context for the rest of the day (e.g., by postponing by 1-2 snapshots the onset of the daily peak in all the cells of a business district). One can then run the offline trainer again, starting from the current time up to the end of the day, discounting the switching operations that have already occurred from $S_{\max}$, and come up with an optimized planning for the rest of the day in a reasonable time, still executing tens of training runs on a multicore machine within a snapshot's worth of computation time.

The communication overhead is also small to negligible. A vector of 96 binaries must be communicated daily (or on demand, in the above cases) to the macros, and the macros themselves can issue the switch-on/off commands to their micros according to the daily plan. Communications among macro and micro nodes belonging to the same hexagon can be realized exploiting the X2 communication and in particular the X2 Application Protocol (X2AP) [31]. The latter defines a set of standard messages that eNBs can exchange. For example, an overload condition can be signaled using the Load Indication Information Element, and an eNB can be switched on using the cell activation procedure.

5. Performance Evaluation

In this section we analyze the performance of the system described in the previous section. We feed our simulator with both synthetic and realistic load curves, specifying for

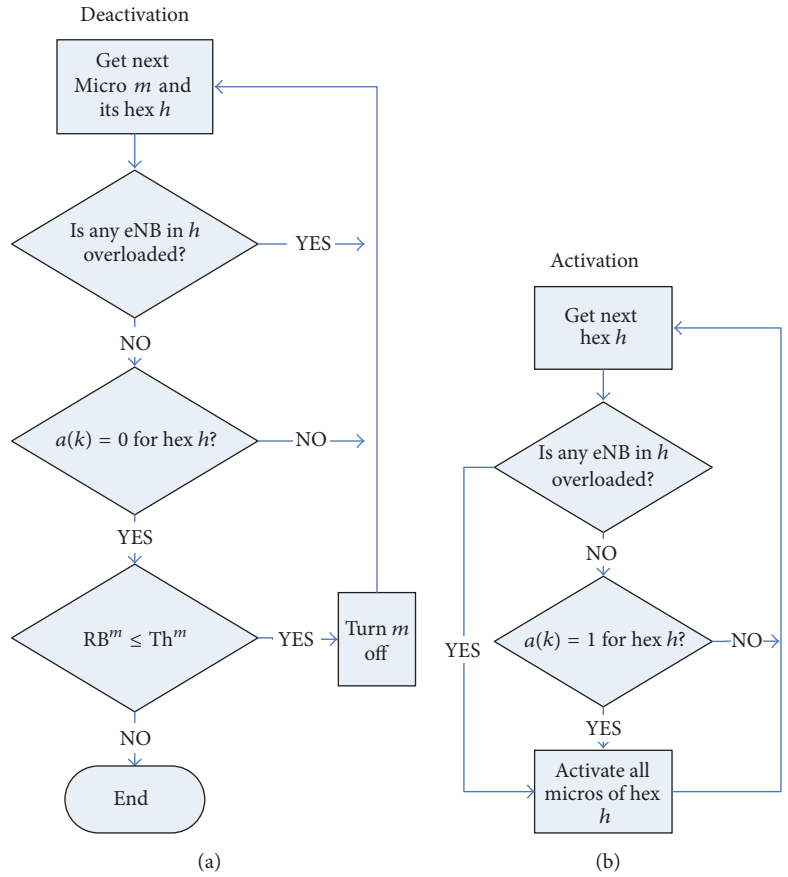

Figure 10: Micro deactivation (a) and activation (b) procedures.

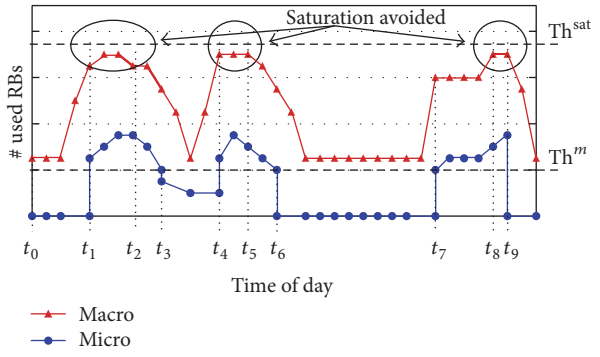

Figure 11: Resource block utilization Example 2.

each snapshot and for each hexagon, the number of active types of UE together with their traffic request in Mbps. Realistic curves are taken from data collected by an operator management system from real deployments. Data are from three hexagons of two different sites, namely, *Site-A* and *Site-B*, and include the average number of active UE and the bandwidth due to data communications during a snapshot. We also consider each UE as performing a VoIP call 1 minute long at 20 kbps. To obtain the total data volume we sum the bandwidth due to data and VoIP communication, and we scale the result by a factor of 1.25 and 1.5 to obtain low, medium, and high system loads. As an example, the resulting average data rates for *Site-A* in the three cases are, respectively, equal to 35, 44, and 53 Mbps. The overall traffic request for the medium load configuration for the whole set of snapshot is represented in Figure 12 by means of curves. Note that their shape is irregular and clearly different from the one assumed in most of the related work.

We first analyze the effects of interference on the system, using synthetic load curves with artificially unbalanced load *among* the hexagons. Then we compare the performance of our algorithm in various configurations against two baselines: a first one $- \mid -$ with all the micros always off and a second one $m \mid m$ with all the micros always on. The above analysis is performed both in case of uniform and nonuniform spatial user distribution *within each* hexagon. Finally, we will test our framework for robustness, verifying how it reacts to changes in the expected load.

System performance will be evaluated in terms of the power variation over $m \mid m$. We will consider only the *variable* part of the power consumption and assess the saving of configuration X over baseline B as $S = (P^X_{\text{var}}/P^B_{\text{var}}) - 1$.

The main simulation parameters are summarized in Table 1.

5.1. Synthetic Load. In this section, we demonstrate the importance of keeping large-scale effects (such as intercell

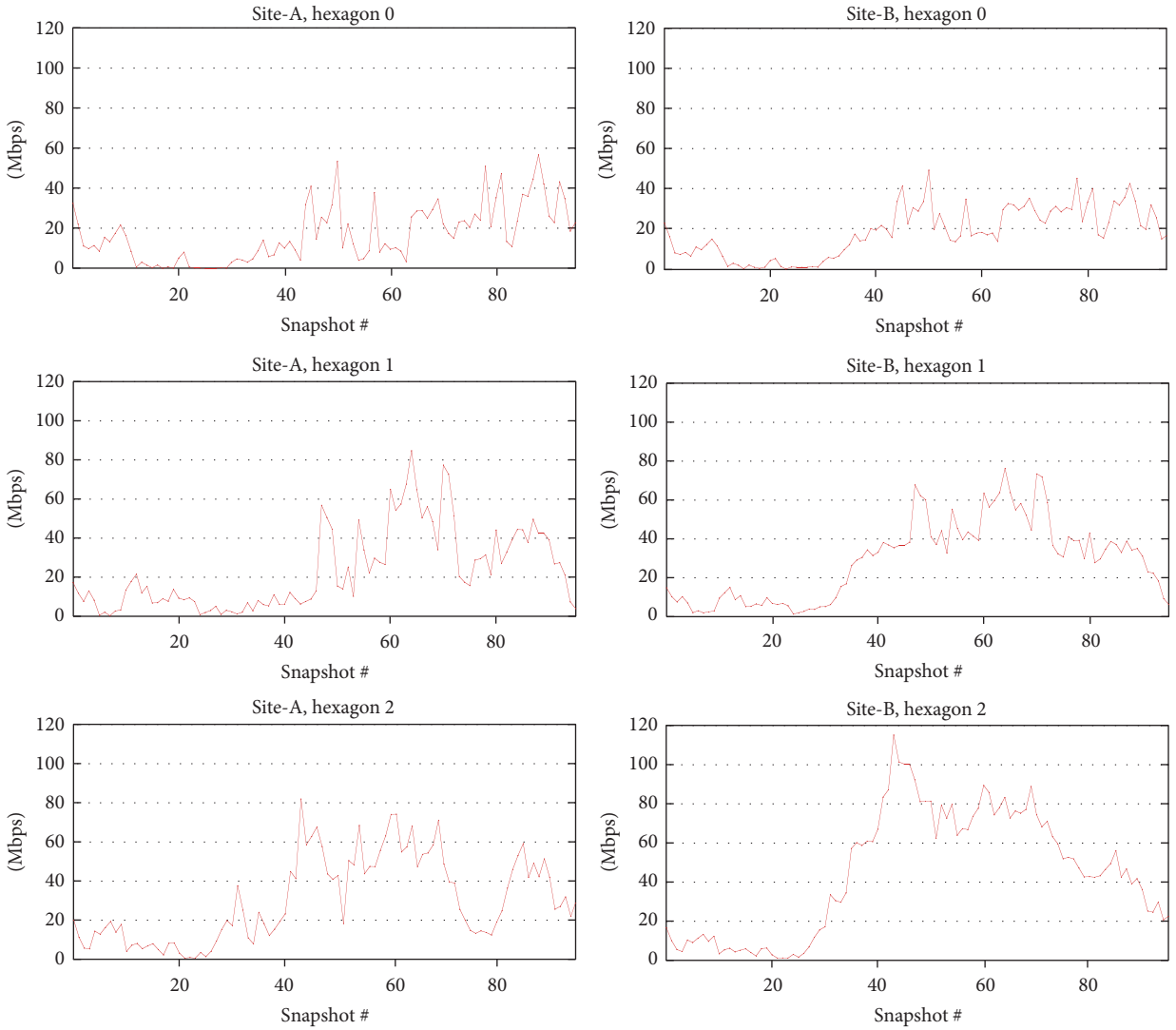

Figure 12: Load curves for medium load, *Site-A* and *Site-B*, and hexagons 0 to 2. *x*-axis reports the snapshot number.

Table 1: Simulation parameters.

	Macro	Micro
Tx power	40 dBm	36 dBm
P_{base}	320 W	16 W
P_{off}	174 W	7 W
ρ	2.51 W	0.450 W
Antenna gain	18 dBi	11 dBi
Intercell distance	500 m	
Path loss model	Urban macro (ITU-R)	
Bandwidth	10 MHz (50 RBs)	
#runs	5 (simulation) 20 (training)	
#snapshots	96	

interference) into account. For this purpose we define a set of *synthetic* load curves, as depicted in Figure 13: hexagon 0 (h_0)—the one being observed— has a constant requested load, while *all* the others (1 to 56) have a variable one. Figure 14 shows the RB occupancy of h_0 in the configurations previously defined ($- \mid -$ and $m \mid m$) plus a third one ($- \mid m$) with *all* the micros on, except the ones in hexagon 0. Two considerations are in order: first, the number of allocated RBs in h_0, thus its power consumption, varies proportionally with the load of neighboring hexagons, despite the requested load in the former being constant (see curve $m \mid m$). The interference from the other hexagons increases in fact with their load.

Second, using micros offloads the macro node, thus avoiding saturation and using resources more efficiently at high loads, as shown by curve $m \mid m$. Note that the interference suffered by h_0, thus its power consumption, drops also when activating micro nodes in neighboring cells only ($- \mid m$).

5.2. Realistic Load. We now analyze the behavior of the system with realistic traffic curves with 57 hexagons, each one composed of one macro and two micro nodes. Each set of three load curves is applied to the central triplet of hexagons and is then replied in the rest of the system using the same

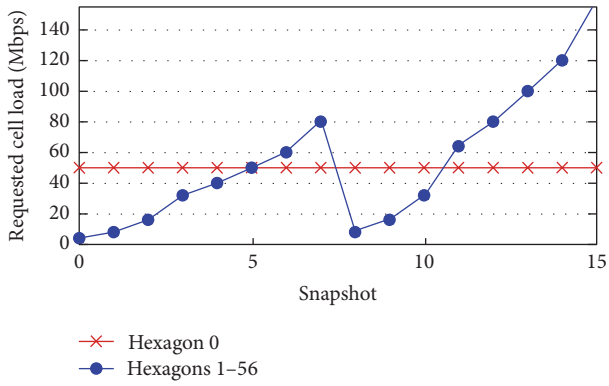

FIGURE 13: Load curves for the *synthetic* scenario.

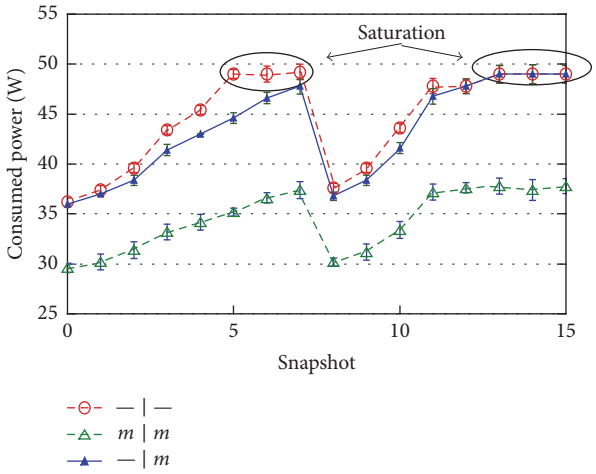

FIGURE 14: Power consumed in h_0 for the *synthetic* scenario. – | –: *all* micros off; m | m: all micros on; and – | m: *all* micros on *except* in hexagon 0.

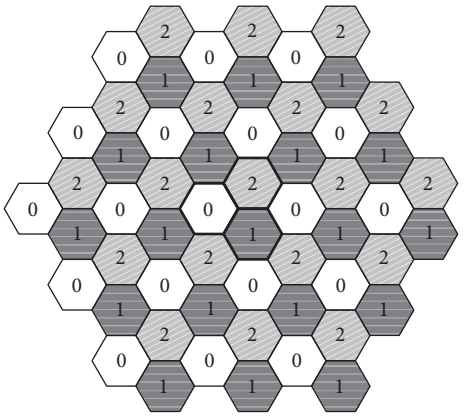

FIGURE 15: Multihexagon network for the realistic load scenario.

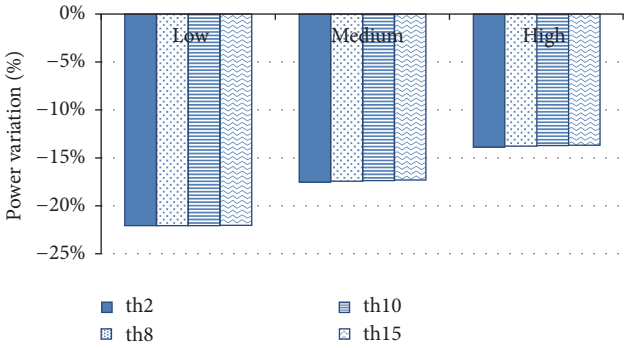

FIGURE 16: Power variation with respect to m | m, obtained in *Site-A* with uniform user distribution, for various Th^m and $S_{\max} = \infty$.

pattern, as represented in Figure 15. With reference to the latter, all the hexagons of the same color will request the same data rate. However, UE is dropped randomly within each hexagon, thus hexagons will differ—possibly in a significant manner—in terms of requested *resources* (which also depend on the UE SINR). This approach allows us to have diverse levels of request resources throughout the system, while still using realistic values of requested data rates.

As a first step we analyze the performance of our system with various values of the threshold Th^m, then we will investigate the effects of $S_{\max}$. Figure 16 shows the power variation in the central triple of hexagons. As we can see the latter is not affected by the value of Th^m, regardless of the number of switches $S_{\max}$. This happens because our algorithm tries to follow the suggested energy-optimal pattern, thus mitigating the effects of erroneous settings of Th^m.

Figure 17 instead shows the power variation in *Site-A* for various configurations of $S_{\max}$. As we can see the highest savings are achieved at low loads, when micros are switched off most of the time. As the load increases, the impact of $S_{\max}$ increases as well, but not in a significant manner. Similar things occur with *Site-B*, as shown in Figure 18, with an impact of $S_{\max}$ that is even lower. This effect is due to the "smoothness" of the load curves and to the uniform distribution of UE among the system. When comparing *Site-A* and *Site-B* loads, higher saving is obtained with the former, as it has lower load requests throughout the day, thus allowing for longer periods with the micros turned off.

To better understand the effects of $S_{\max}$, in Figure 19 we report the cell load of h_0 in *Site-A* with high load. Moreover in the lower part of the figure we show three binary curves, each one representing the optimal energy-activation pattern for the micros, for three values of $S_{\max}$, in the shape of three binary curves indicating the values of a_k^h. There are two saturation zones (circled peaks at the top of the figure). In these cases every micro *must* be turned on so as to provide additional capacity. There are two more zones (boxed areas) where the macro is not saturated, but the amount of requested resources is such that having the micros on is energetically profitable. In the remaining snapshots the best solution is having the micros off. The first two conditions (saturation) are satisfied with every value of $S_{\max}$, while the last one is fully achieved only for $S_{\max}$, when micros are switched only when necessary, regardless of the number of switches.

In Figures 20 and 21 we show instead the total number of overloaded RBs during the whole day, that is, the number of RBs, beyond those available, which would be required to cope with the traffic request of a day. This allows us to measure

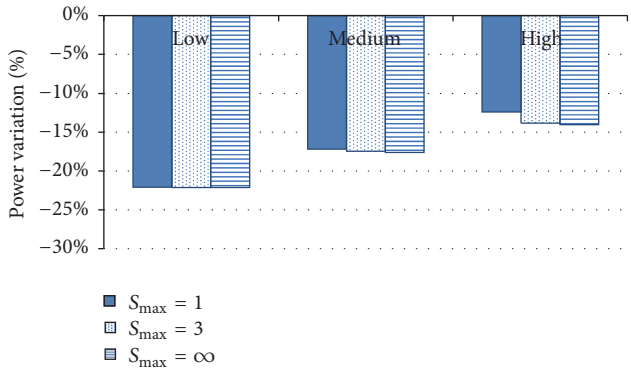

FIGURE 17: Power variation with respect to $m \mid m$, obtained in *Site-A* with uniform user distribution.

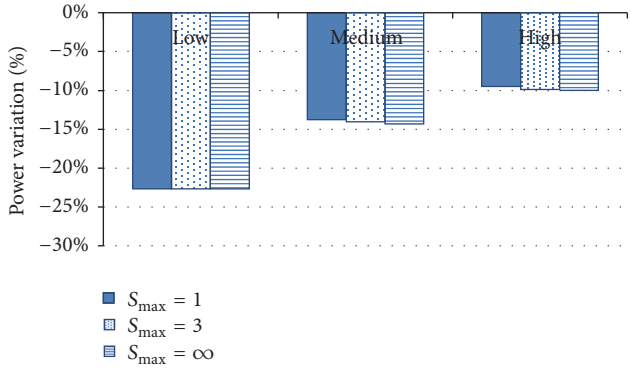

FIGURE 18: Power variation with respect to $m \mid m$, obtained in *Site-B* with uniform user distribution.

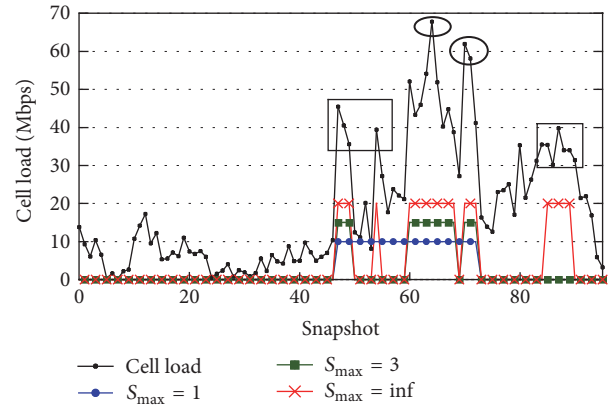

FIGURE 19: Switches for h_0, *Site-A* with level = high, $S_{max} = \{1, 3, \infty\}$, and uniform user distribution.

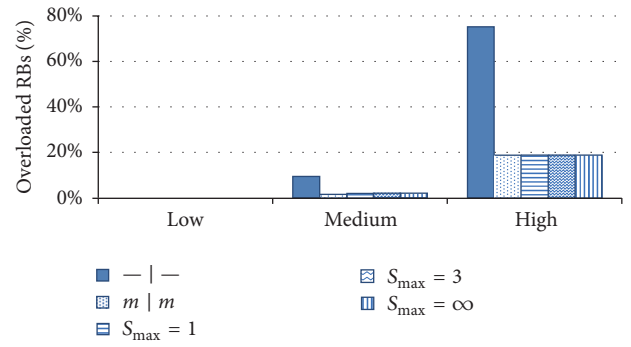

FIGURE 20: Per hexagon overloaded RBs over a day, reported as a percentage of the number of available RBs, obtained in *Site-A* with uniform user distribution.

the ability of the system to satisfy the traffic request itself. As we can see, configuration $m \mid m$ performs way better than $- \mid -$; thus its performance has to be considered as the target of our algorithm. The latter in fact obtains the same number of overloaded RBs as $m \mid m$, regardless of the choice of S_{max}, thus enforcing energy saving with *no impact* on the carried load.

5.3. Nonuniform User Distribution. In this section, we will analyze the performance of our algorithm in case of nonuniform and time-varying distribution of UE within the system. For each hexagon, users are randomly dropped inside *one* hotspot of center c and radius r, as presented in Figure 7(a). Value c is changed every five snapshots, alternating between two positions, closer to the macro (hs_1) and to the micro (hs_2), respectively. The value of r is instead picked randomly in the interval $[20m, 30m]$, every snapshot.

In Figures 22 and 23 we show the power variation obtained with respect to $m \mid m$, for both *Site-A* and *Site-B*. The impact of S_{max} is in this case greater than with a uniform user distribution. Moreover, this effect increases with load, as the energy-wise efficiency of micros is higher with larger traffic demands. As in the previous scenario, the performance of our algorithm from the carried load point of view, is equal to those obtained with $m \mid m$ and is thus not shown.

The analysis in Figures 17–23 suggests that a larger S_{max} (which, as we remark, is an input datum and not a decision variable) yields diminishing returns in terms of power savings. Most of the power, in fact, is saved with the first few switch-offs—sometimes even a single one—beyond which topology alterations get progressively less useful. However, while there *must* be a point where increasing S_{max} stops yielding tangible benefits (unless load curves are collections of IID points, which is not what appears from measurements), its location depends on several factors: the shape of the load curves, the distribution of the position of UE, the power profiles of the nodes, and so on. In fact, Figures 22 and 23 show that increasing S_{max} from one to three in a hotspot scenario yields considerable benefits, unlike with a uniform distribution (see Figures 17 and 18).

5.4. Resilience Analysis. In this section, we will test how our solution reacts to variation in the expected load. More in detail, we want to check if the *Online Decider* is able to recover from wrong suggestions coming from the *Offline Analyzer* or to react to unexpected changes in the actual load of the system.

As a first step, given a load curve of an hexagon for a certain site, for every snapshot we take the *original* bitrate request, calling it b_r, and we compute the *modified* bitrate

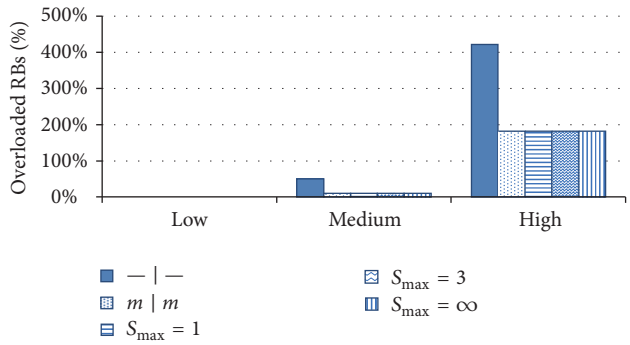

FIGURE 21: Total number of per hexagon overloaded RBs over a day, reported as a percentage of the number of available RBs, obtained in *Site-B* with uniform user distribution.

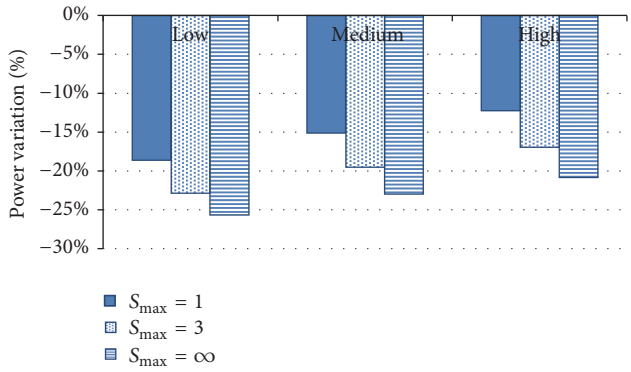

FIGURE 22: Power variation with respect to $m \mid m$, obtained in *Site-A* with hotspot user distribution.

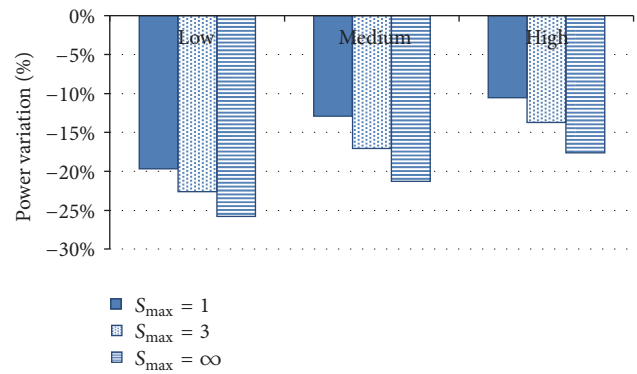

FIGURE 23: Power variation with respect to $m \mid m$, obtained in *Site-B* with hotspot user distribution.

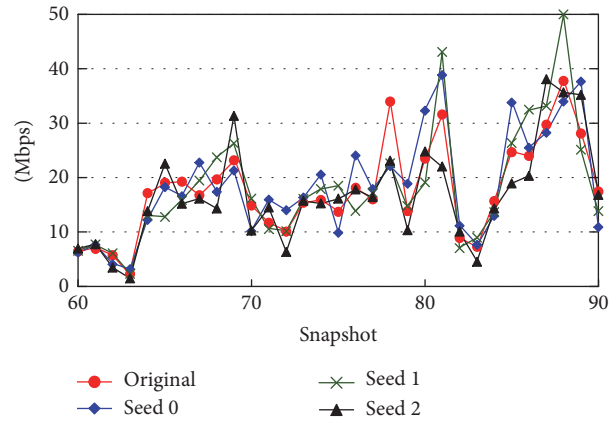

FIGURE 24: Comparison of *original* load against three *modified* loads, computed with γ extracted from the interval $[0.6, 1.4]$ with three different seeds.

request as $\tilde{b}_r = b_r \times \gamma$, where γ is a random value extracted from a uniform distribution in the interval $[\gamma_L, \gamma_R]$. Note that we are not varying the number of UE within a hexagon, but only their load requests. Figure 24 shows a portion of the *original* load curve in h_0 for *Site-A* load, compared against three *modified* load curves obtained with a γ extracted in $[0.6, 1.4]$ with three different seeds.

Although the shape of the curve is roughly the same, pointwise variations can be clearly observed. We repeated the above procedure in order to obtain various modified load curves. For each of them we first fed the *trainer* with the original load curve, and then we perform simulations using the modified ones. We used this method for both *Site-A* and *Site-B* and for each load level. As we can see in Figures 25 and 26, the effects of the variations on the power saving are negligible in all cases. This means that, as far as the shape of the load curve is maintained, the *Online Decider* is able to react to variation in the requested load.

6. Conclusions

This paper presented a framework for the activation/deactivation of nodes in a heterogeneous network, where the number of switch-ons in a day is constrained. We first analyzed the limits of both online- and offline-only algorithms, demonstrating how optimality requires unrealistic hypotheses. We then proposed and evaluated an online-offline method consisting of three phases: training, offline analysis, and online decision. The training phase is done offline and allows one to evaluate the energy consumption and saturation probabilities of alternative configurations, taking into account intercell interference *at large*. The offline analysis provides the optimal (constrained) switching pattern, and the online decision algorithm evaluates the current network load and makes switching decisions, erring on the safe side as far as saturation probability is concerned. Our framework works with arbitrary load curves and power figures, and it takes a predefined maximum number of switch-offs as an input. Simulation results show performance savings between 10 and 25% with both synthetic and real traffic loads and with both uniform and nonuniform user distributions. Future work includes developing and analyzing switching algorithms that takes into account more detailed information on the spatial distribution of UE, as well as the possible benefits given by *enhanced Intercell Interference Coordination* (eICIC) techniques at the eNBs. Both frequency-based (e.g., coordinated scheduling or partial frequency reuse) and time-based (e.g., provisioning of Absolute Blank Subframes) can easily be taken into account into our framework. All it takes is

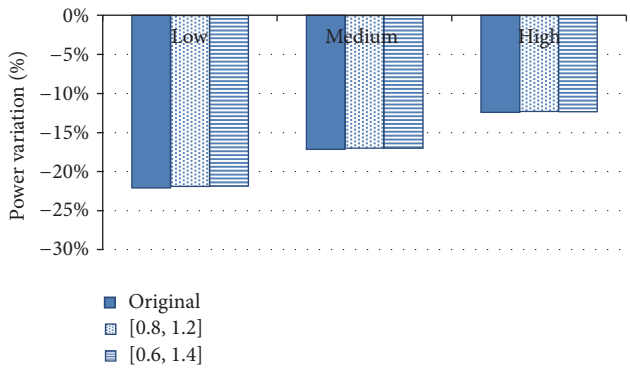

FIGURE 25: Power variation obtained using $S_{max} = 1$ with respect to $m \mid m$, with *original Site-A* load curves and *modified* ones with γ randomly extracted from two intervals.

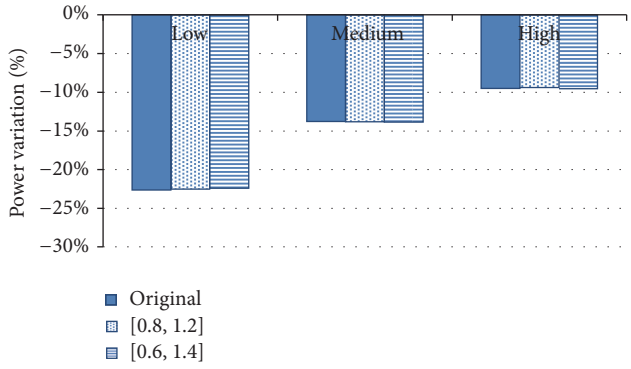

FIGURE 26: Power variation obtained using $S_{max} = 1$ with respect to $m \mid m$, with *original Site-B* load curves and *modified* ones with γ randomly extracted from two intervals.

to modify expression (4), by correctly accounting the number of overlapping RBs between couples of eNBs depending on the eICIC technique.

Finally, minimizing the number of topology alterations (i.e., switch-ons and switch-offs) required to achieve a target power saving in a network is a related problem that we plan to investigate in the future.

LTE-Related Acronyms

CoMP:	Coordinated Multipoint
CQI:	Channel Quality Indicator
CRE:	Cell Range Expansion
eNB:	Evolved Node-B
LTE:	Long-Term Evolution
LTE-A:	Long-Term Evolution Advanced
MCS:	Modulation Coding Scheme
MTBF:	Mean Time between Failures
MBSFN:	Multicast Broadcast Single Frequency Network
RB:	Resource Block
SINR:	Signal to Interference and Noise Ratio
TTI:	Transmission Time Interval
UE:	User Equipment.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The subject matter of this paper includes description of results of a joint research project carried out by Telecom Italia and the University of Pisa. Telecom Italia reserves all proprietary rights in any process, procedure, algorithm, article of manufacture, or other results of said project herein described. A preliminary version of this paper was presented to the 5th InfQ Workshop, Torino, Italy, October 2014. The authors would like to thank Alessandro Donati, former M.S. student at the University of Pisa, now with Telecom Italia, for his help with the simulations. This work was carried out while Dario Sabella was with Telecom Italia, Turin, Italy.

References

[1] S. Lambert, W. Van Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet, "Worldwide electricity consumption of communication networks," *Optics Express*, vol. 20, no. 26, pp. B513–B524, 2012.

[2] M. Sajid Mushtaqa, S. Fowlerb, A. Mellouka, and B. Augustina, "QoE/QoS-aware LTE downlink scheduler for VoIP with power saving," *Journal of Network and Computer Applications*, vol. 51, pp. 29–46, 2015.

[3] G. Stea and A. Virdis, "A comprehensive simulation analysis of LTE discontinuous reception (DRX)," *Computer Networks*, vol. 73, pp. 22–40, 2014.

[4] R. Dinga, C. Hava Munteanb, and G. Munteana, "Energy-efficient device-differentiated cooperative adaptive multimedia delivery solution in wireless networks," *Journal of Network and Computer Applications*, vol. 58, pp. 194–207, 2015.

[5] Ł. Budzisz, F. Ganji, G. Rizzo et al., "Dynamic resource provisioning for energy efficiency in wireless access networks: a survey and an outlook," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 2259–2285, 2014.

[6] Y. Zhang, Ł. Budzisz, M. Meo et al., "An overview of energy-efficient base station management techniques," in *Proceedings of the 2013 24th Tyrrhenian International Workshop on Digital Communications - Green ICT (TIWDC)*, Genoa, Italy, September 2013.

[7] W. Vereecken, W. V. Heddeghem, M. Deruyck et al., "Power consumption in telecommunication networks: overview and reduction strategies," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 62–69, 2011.

[8] J. B. Rao and A. O. Fapojuwo, "A survey of energy efficient resource management techniques for multicell cellular networks," *IEEE Communications Surveys and Tutorials*, 2013.

[9] D. Feng, C. Jiang, G. Lim, L. J. Cimini, G. Feng, and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Communications Surveys & Tutorials*, 2013.

[10] M. Ajmone Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Switch-off transients in cellular access networks with sleep modes," in *Proceedings of the 2011 IEEE International Conference on Communications Workshops, ICC 2011 Workshops*, Kjoto, Japan, June 2011.

[11] A. Bousia, E. Kartsakli, L. Alonso, and C. Verikoukis, "Energy efficient base station maximization switch off scheme for LTE-advanced," in *Proceedings of the 2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2012*, Barcelona, Spain, September 2012.

[12] A. Conte, A. Feki, L. Chiaraviglio, D. Ciullo, M. Meo, and M. A. Marsan, "Cell wilting and blossoming for energy efficiency," *IEEE Wireless Communications*, vol. 18, no. 5, pp. 50–57, 2011.

[13] P. Dini, M. Miozzo, N. Bui, and N. Baldo, "A model to analyze the energy savings of base station sleep mode in LTE HetNets," in *Proceedings of the IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing (GreenCom-iThings-CPSCom '13)*, Bejing, China, August 2013.

[14] M. Ajmone Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "On the effectiveness of single and multiple base station sleep modes in cellular networks," *Computer Networks*, vol. 57, no. 17, pp. 3276–3290, 2013.

[15] M. Deruyck, E. Tanghe, W. Joseph, and L. Martens, "Characterization and optimization of the power consumption in wireless access networks by taking daily traffic variations into account," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, 2012.

[16] B. Zhuang, D. Guo, and M. L. Honig, "Energy management of dense wireless heterogeneous networks over slow timescales," in *Proceedings of the 2012 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012*, pp. 26–32, Allerton, USA, October 2012.

[17] J. Wu, S. Zhou, and Z. Niu, "Traffic-aware base station sleeping control and power matching for energy-delay tradeoffs in green cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 4196–4209, 2013.

[18] S. H. Lee and I. Sohn, "Distributed energy-saving cellular network management using message-passing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 635–644, 2017.

[19] F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *IEEE/ACM Transactions on Networking*, 2017.

[20] N. Yu, Y. Miao, L. Mu, H. Du, H. Huang, and X. Jia, "Minimizing energy cost by dynamic switching on/off base stations in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7457–7469, 2016.

[21] 3GPP - TS 36.300 Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2.

[22] D. Sabella, M. Caretti, W. Tomaselli et al., "Evaluation of ON-OFF schemes and linear prediction methods for increasing energy efficiency in mobile broadband networks," *NETWORKING 2012 Workshops*, 2012.

[23] D. Sabella, D. Rapone, M. Fodrini et al., "Energy management in mobile networks towards 5G," in *Energy Management in Wireless Cellular and Ad-hoc Networks, Studies in Systems, Decision and Contro*, M. Z. Shakir, M. A. Imran, K. A. Qaraqe, M.-S. Alouini, and A. V. Vasilakos, Eds., Springer, 2016.

[24] J. Olmos, A. Serra, S. Ruiz, M. García-Lozano, and D. Gonzalez, "Link level simulator for LTE downlink," in *Proceedings of 7th European Meeting COST-2100. Pervasive Mobile & Ambient Wireless Communications. Braunschweig*, COST, 2009.

[25] R. Fantini, D. Sabella, and M. Caretti, "An E3F based assessment of energy efficiency of relay nodes in LTE-advanced networks," in *Proceedings of the 2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'11*, pp. 182–186, can, September 2011.

[26] S. Thapar and P. Karmakar, "Performance evaluation of LTE network: An energy saving and capacity gain perspective," in *Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2532–2538, Jaipur, India, September 2016.

[27] D. Migliorini, G. Stea, M. Caretti, and D. Sabella, "Power-aware allocation of MBSFN subframes using discontinuous cell transmission in LTE systems," in *Proceedings of the 2013 IEEE 78th Vehicular Technology Conference, VTC Fall 2013*, Las Vegas, USA, September 2013.

[28] "ITU-R, M.2135-1, Guidelines for evaluation of radio interface technologies for IMT-Advanced, 12/2009".

[29] ILOG CPLEX Software, http://www.ilog.com.

[30] N. Iardella, G. Nardini, G. Stea, A. Virdis et al., "Flexible dynamic Coordinated Scheduling in Virtual-RAN deployments," in *Proceedings of FlexNets 2017*, Paris, FR, May 2017.

[31] "3GPP - TS 36.432 Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 application protocol (X2AP)".