Energy-Efficient Selective Activation in Femtocell Networks

Michael Lin
CSE Department
Pennsylvania State University
Email:molin@cse.psu.edu

Simone Silvestri
Computer Science Department
Missouri S&T
Email:silvestris@mst.edu

Novella Bartolini
Computer Science Department
Sapienza University, Italy
Email:novella@di.uniroma1.it

Thomas La Porta
CSE Department
Pennsylvania State University
Email:tlp@cse.psu.edu

***Abstract*—Provisioning the capacity of wireless networks is difficult when peak load is significantly higher than average load, for example, in public spaces like airports or train stations. Service providers can use femtocells and small cells to increase local capacity, but deploying enough femtocells to serve peak loads requires a large number of femtocells that will remain idle most of the time, which wastes a significant amount of power. To reduce the energy consumption of over-provisioned femtocell networks, we formulate a femtocell *selective activation* problem, which we formalize as an integer nonlinear optimization problem. Then we introduce GREENFEMTO, a distributed femtocell selective activation algorithm that deactivates idle femtocells to save power and activates them on-the-fly as the number of users increases. We prove that GREENFEMTO converges to a locally Pareto optimal solution and demonstrate its performance using extensive simulations of an LTE wireless system. Overall, we find that GREENFEMTO requires up to 55% fewer femtocells to serve a given user load, relative to an existing femtocell power-saving procedure, and comes within 15% of a globally optimal solution.**

I. INTRODUCTION

In recent years, the use of mobile broadband wireless devices has been growing exponentially. Market studies predict that global mobile broadband subscriptions will reach 4.4 billion by 2016, and have shown that global mobile data traffic has grown at an annual rate of 150% [1]. As music and video streaming have increased in popularity, users have become accustomed to using mobile data everywhere, all the time, and there has been an increase in the occurrence of temporary, high-density concentrations of mobile users with high traffic demands that lead to service outages [2], also known as *flash crowds*. Flash crowds can occur intermittently, such as during sporting events or concerts, or regularly, such as when trains arrive at stations, at airport gates, or in classroom buildings. Although solutions exist to handle temporary flash crowds, recurring flash crowds place different, highly variable demands on the network, and require different design choices.

Current solutions to serving temporary flash crowds include over-provisioning networks and mobile cell sites, but both are ill-suited for recurring flash crowds. Over-provisioned wireless networks can serve peak flash crowd loads, but they are wasteful during normal use. Mobile cell sites [3] are designed to mitigate temporary flash crowd overloads, but they are not suitable for regularly occurring crowds. This paper describes GREENFEMTO, an algorithm for reducing the energy impact of over-provisioned, densely-deployed femtocell networks designed to serve recurring flash crowds.

Femtocells are small, short-range wireless base stations that rely on user-provided Internet for backhaul. Provisioning a short-range femtocell network for a high peak load is more efficient than provisioning a macrocell network for the same load, which makes femtocells ideal for serving flash crowds. However, to serve large crowds, femtocells must be densely deployed, which increases network energy consumption. For example, assuming femtocell power consumption of a constant 5 W, a network of 100 cells covering a subway station would consume more than 5 MWh/year.

Previous solutions for designing high-density wireless networks have largely focussed on quality of service, and only marginally consider power consumption [2], [4], [5], [6]. GREENFEMTO is designed from the ground up to serve large crowds while minimizing energy consumption under certain area and user coverage constraints.

This paper studies the problem of minimizing the energy consumption of dense, redundantly-deployed femtocell networks. To reduce network energy consumption, we model the problem as a selective activation problem. Only a reduced set of femtocells are kept active at a given time. The remaining cells are put to sleep to conserve power and are activated by a dynamic and distributed algorithm as needed to serve users.

In Section II, we motivate the use of selective activation, rather than power management, through measurements of real femtocell energy usage, which show that femtocell power consumption is not dependent on load. This implies that deactivating femtocells is the most effective way to reduce their energy consumption. We describe our network model and problem in Section III. Then, in Section IV, we formalize the selective activation problem as a non-linear optimization problem that considers LTE user resource requirements. Section V describes GREENFEMTO, a distributed algorithm for femtocell selective activation that dynamically adapts the active set of cells on the basis of local user information and requires no network-wide synchronization. We prove that GREENFEMTO converges to a stable solution and that the solution is locally Pareto-optimal in Section VI. Finally, we evaluate the performance of GREENFEMTO against a previously proposed femtocell power-saving algorithm using simulations in Section VII. Our results show that GREENFEMTO outperforms the previous state-of-

TABLE I: Measured Femtocell Power Consumption in Watts

	Number of Users				
Location	0	1	2	3	4
A	4.6 W	4.7 W	4.7 W	4.7 W	4.7 W
B	4.6 W	4.7 W	4.7 W	4.7 W	4.7 W

the-art, reduces network energy consumption by up to 55%, and comes within 10% of a globally optimal solution.

In summary, our contributions are the following:

- We perform real measurements of the energy consumption of femtocells to motivate the use of selective activation;
- We formulate the problem of reducing the energy consumption of densely deployed femtocell networks as a non-linear optimization problem;
- We introduce GREENFEMTO, a distributed algorithm for femtocell selective activation. We prove that GREENFEMTO converges to a locally Pareto-optimal solution;
- We measure the performance of GREENFEMTO through simulations, showing that it successfully reduces energy consumption and outperforms previous solutions;

II. FEMTOCELL POWER MEASUREMENTS

We motivate the use of selective activation through measurements of the power usage of commercially available femtocells. We find that femtocells consume a fixed amount of power, independent of the number of users and their locations, which corroborates results presented in [7]. Although femtocell radio transmission power varies with the number and location of users being served, femtocell radios have a maximum transmit power in the range of 100-250 mW [8], leading to low variation in overall power consumption that is dominated by fixed-power components. Since femtocell power use remains constant relative to load and user position, selective activation is the most effective way of reducing a femtocell network's energy consumption.

The experiments below were conducted using AT&T 4G Microcells [8], which have a maximum capacity of four users. We record the femtocell's power usage with zero to four active users at two locations: location A, 0.5 meters away in the same room, with an average received signal strength indicator of -59 dBm, and location B, at the edge of the femtocell's coverage radius, with an average received signal strength indicator of -91 dBm, approximately 15 meters away, through multiple interior walls. Each value in Table I is the average of five measurements. As the load increases, the femtocell's power usage remains stable.

We conclude that femtocells consume a fixed amount of power when activated. In the remainder of the paper and the design of the algorithm, we focus on minimizing the number of femtocells that are active, rather than managing the power consumption of individual femtocells or users.

III. NETWORK MODEL AND PROBLEM FORMULATION

In this section we define our network model and problem. We consider an LTE data network with femtocells. The channel bandwidth is divided into subchannels, each of which are further time-divided [9]. Each time division on a subchannel is known as a *resource block* (RB). User transmissions are scheduled across RBs on the base station to which the user is associated. Base stations regularly transmit reference signals (RS) that are used to measure signal quality.

Consider an area of interest $\mathcal{L}$, defined as a set of points $p \in \mathcal{L}$. We define a set $F = \{f_1, \ldots, f_K\}$ of K femtocells deployed over $\mathcal{L}$. We consider two types of coverage: *area coverage* and *user coverage*, with respective thresholds t^a and t^u. Area coverage is defined as coverage of a set of points regardless of users, and is used to detect users. User coverage is defined as the ability to serve a user at his minimum rate, and is used for user service. A location or user is *covered* if the SINR of the RS at that point is above t^a or t^u.

A femtocell f_i's *coverage range*, R_i^a or R_i^u, is defined as the largest contiguous set of locations at which f_i's RS can be received with SINR above t^a or t^u, respectively. $t^u > t^a$, therefore, $||R_i^a|| > ||R_i^u||$. We assume each femtocell is aware of its location and coverage range[1]. A location is redundantly covered if it is covered by at least two femtocells. Femtocells communicate locally using the network that they use for backhaul.

Users are defined as a set $U = \{u_1, \ldots, u_M\}$. Users are served by femtocells that have enough free RBs to schedule the user for transmission. Users are rejected if no femtocell can schedule the user. To emphasize the effect of user rejections, we consider a network without macrocells, however, this is not a requirement for our algorithms. Since femtocell networks are often deployed to improve signal quality in areas with poor-or-no macrocell coverage, such as subways, this is a reasonable assumption.

The selective activation problem is to find a minimal set of *active femtocells*, $F^* \subseteq F$, that provides full area and user coverage. This implies that all locations in $\mathcal{L}$ have an SINR above t^a, and all users have an SINR above t^u. The assignment of users in U to femtocells in F^* must be feasible in terms of femtocell capacity and must satisfy user requirements in terms of minimum acceptable rate. If not all users can be served at their rate targets, users are rejected one by one.

IV. INTEREFERENCE-AWARE FEMTOCELL ACTIVATION OPTIMIZATION

In the following we formulate the problem from Section III as an integer non-linear optimization problem.

An *activation policy* defines the set of active femtocells and is expressed as an *activation vector*, $\bar{X} \triangleq \{x_1, x_2, \ldots, x_M\}$, where x_i is 1 if the femtocell f_i is activated, and 0 if f_i is disabled.

The objective of our problem is to find an activation policy that serves all users in U with the minimum number of active femtocells; that is, an activation vector $\bar{X}$ that minimizes $\sum_{i=1}^{N} x_i$.

[1]In a centrally managed and deployed network, femtocell coverage ranges can be estimated at deployment.

If a cell $f_i \in F$ serves user $u_j \in U$, it schedules u_j for a given number of resource blocks, represented by integer variable y_{ij}. Femtocells have a capacity limitation, as the number of resource blocks available for each cell is limited to $b_{\texttt{MAX}}$. Since the energy consumption of a femtocell is not dependent on the number of resource blocks it allocates, it is beneficial to allocate as many resource blocks as possible, which improves quality of service. Therefore, we assume that every active cell allocates all of its available resource blocks, as expressed in the following constraint:

$$\sum_{j=1}^{M} y_{ij} = x_i \cdot b_{\texttt{MAX}}, \quad \forall i \text{ s.t. } f_i \in F \tag{1}$$

Given a user u_j, served by the cell f_i, and given the activation vector of the network $\bar{X}$, the rate requirement r for u_j can be defined as a function $b_{ij}(\cdot)$, that returns the minimum number of resource blocks $b_{ij}(\bar{X})$ that user u_j needs from cell f_i to operate at rate higher than or equal to r, given that user j's SINR is above t^u. The function b_{ij} depends on system-wide SINR calculations, is non-linear, and there is no known closed form expression. Therefore, the user quality requirement implies that $y_{ij} \geq b_{ij}(\bar{X})$ when the cell f_i is selected to serve user u_j.

Furthermore, note that every user u_j must be served by one and only one cell, and it must receive all resource blocks from the same cell. To express this property, we introduce a binary assignment variable z_{ij}. The variable z_{ij} is 1 if user u_j is served by cell f_i and 0 otherwise. z_{ij} expresses the mutually exclusive constraint that a user is either served by cell f_i with at least $b_{ij}(\bar{X})$ resource blocks (the case when $z_{ij} = 1$), or it is served by one of the other cells f_k, with $k \neq i$ (the case when $z_{ij} = 0$). This property can be expressed by the following set of constraints:

$$y_{ij} \geq z_{ij} \cdot b_{ij}(\bar{X}) \qquad f_i \in F, u_j \in U \tag{2}$$

$$\sum_{\substack{k=1\\k\neq i}}^{N} y_{kj} \leq (1 - z_{ij}) \cdot |F| \cdot b_{\texttt{MAX}} f_i \in F, u_j \in U \tag{3}$$

$$y_{ij} \leq b_{\texttt{MAX}} \cdot z_{ij} \qquad f_i \in F, u_j \in U \tag{4}$$

$$\sum_{\substack{k=1\\k\neq i}}^{N} y_{kj} \geq 1 - z_{ij} \qquad f_i \in F, u_j \in U \tag{5}$$

Finally, we impose a constraint on area coverage. Each cell f_i is able to detect the presence of users in their area coverage range, R_i^a. Note that the area coverage range is larger than the user coverage range R_i^u, and cell f_i may be unable to serve a user at distance R_i^a. Nevertheless, if every point of the area of interest is within the coverage range of an active cell, the network is able to sense any incoming users and activate additional femtocells as needed. To define the area coverage constraint, we use a theorem about coverage of open intervals, which we prove next.

Recall that coverage ranges are open intervals, that is, femtocell f_i does not cover the border of its area R_i^a. We define an *intersection point* as a point at which the borders of two coverage ranges intersect, or a point at which a coverage range intersects with the border of the area $\mathcal{L}$. We denote by $\mathcal{I}$ the set of all intersection points in $\mathcal{L}$. In the following theorem, we extend a result introduced in [10] to the case of general non-convex-shaped coverage areas, which we use to show that the coverage of the points in $\mathcal{I}$ implies complete coverage of $\mathcal{L}$.

Theorem IV.1. *Given a set of femtocells F, deployed over an area $\mathcal{L}$ and generating a set of intersection points $\mathcal{I} \subset \mathcal{L}, \mathcal{I} \neq \emptyset$, if a subset $\hat{F} \subseteq F$ covers all points in $\mathcal{I}$, then $\hat{F}$ completely covers $\mathcal{L}$.*

Proof: The set of coverage ranges partitions the area $\mathcal{L}$ into coverage patches $H_1, \ldots, H_m$, where the points within each patch have the same coverage degree and patches are bounded by the borders of coverage ranges or by the borders of $\mathcal{L}$. The border of a coverage patch H_i contains intersection points on the border of two coverage ranges, or the border of a coverage range and the border of $\mathcal{L}$.

We proceed by contradiction. Let $p \in \mathcal{L}$ be a point not covered by the femtocells in $\hat{F}$ and let H_p be the coverage patch to which it belongs. H_p always exists since the coverage patches partition $\mathcal{L}$. All points in H_p have the same coverage degree, by definition of coverage patch, therefore H_p is also not covered. Since we define coverage ranges to be open intervals, an intersection point on the border of H_p, generated by the intersection of the borders of two coverage ranges R_i and R_j, is not covered by the femtocell f_i and f_j. The intersection points on the border of H_p, if any, are also uncovered.

This is a contradiction of the assumption that all intersection points are covered by the cells in $\hat{F}$, therefore the uncovered point p does not exist. ∎

$\mathcal{I}$ includes all the intersection points generated by the coverage regions of all the femtocells in F. According to Theorem IV, if all points of $\mathcal{I}$ are within the coverage range of an active cell, all of $\mathcal{L}$ is covered, and the active cells are able to detect the presence of a user in any point of the area.

Therefore, we include area coverage as follows. Let $l \in \mathcal{I}$ be a point in the region of interest. Let p_{il} be a binary constant coefficient which is equal to 1 if the point l is in the coverage range of femtocell f_i. Then, in order to have complete area coverage, the activation set $\bar{X}$ must satisfy the following constraint.

$$\sum_{i=1}^{N} x_i p_{il} \geq 1, \quad \forall l \in \mathcal{L} \tag{6}$$

Summarizing the previous discussion, we obtain the following integer non-linear optimization problem, hereby referred to as **MinActivation**:

$$\min \sum_{i=1}^{N} x_i \tag{7}$$

$$\sum_{j=1}^{N} y_{ij} = x_i \cdot b_{\mathtt{MAX}} \qquad \forall i\colon f_i \in F \tag{8}$$

$$y_{ij} \geq z_{ij} \cdot b_{ij}(\bar{X}) \qquad f_i \in F, u_j \in U \tag{9}$$

$$\sum_{\substack{k=1\\k\neq i}}^{N} y_{kj} \leq (1 - z_{ij}) \cdot |F| \cdot b_{\mathtt{MAX}} f_i \in F, u_j \in U \tag{10}$$

$$y_{ij} \leq b_{\mathtt{MAX}} \cdot z_{ij} \qquad f_i \in F, u_j \in U \tag{11}$$

$$\sum_{\substack{k=1\\k\neq i}}^{N} y_{kj} \geq 1 - z_{ij} \qquad f_i \in F, u_j \in U \tag{12}$$

$$\sum_{i=1}^{N} x_i p_{il} \geq 1 \qquad \forall l \in \mathcal{L} \tag{13}$$

Using a reduction to Set Cover, it is straightforward to show that **MinActivation** is NP-Hard; we omit the proof for space considerations.

V. The GreenFemto Algorithm

In this section we introduce GreenFemto, a distributed algorithm that finds a locally Pareto optimal solution to **MinActivation**. The algorithm consists of two parts that address two different aspects of user management: user detection and user reassignment.

A. Overview

A femtocell can be *active* or *inactive*. Active femtocells provide area coverage and can serve users, but consume energy. Inactive femtocells do not provide coverage or serve users, but do not consume energy. We assume that femtocells can be turned on remotely using Wake-on-LAN [11].

For each femtocell $f_i \in F$ we define $U_i \subseteq U$, the set of users f_i can serve. We say that a femtocell f_j is a *neighbor* of a femtocell f_i, if their coverage ranges overlap, that is $R_i^u \cap R_j^u \neq \emptyset$. We refer to the set of all neighbors, awake or sleeping, of f_i as B_i. The set of active neighbors is denoted by B_i^*, and inactive neighbors by B_i^o. c_i is the residual capacity of the femtocell f_i, that is, the number of free RBs it can schedule. We assume that each femtocell f_i knows the coverage ranges, R_j^u and R_j^a, of each of its neighbors.

GreenFemto keeps a minimal set of *sentinel* femtocells active for area coverage. This set is found by disabling redundant femtocells, as identified using the procedure in Section V-D. Users are detected by sentinel femtocells, and either served directly by the sentinel femtocell or served by a newly awoken femtocell. After femtocells are awoken, *user reassignment* and a *redundancy test* are performed to minimize the number of active femtocells. These procedures are described in detail in the following sections.

B. User Detection and Femtocell Activation

When a new user u_j joins the network, at least one femtocell detects the user, since femtocells provide area coverage. This femtocell is referred to as the *sentinel femtocell*, f_s. Furthermore, we assume that u_j notifies f_s of the active set of cells $F^{u_j} = \{f_1^{u_j}, \ldots, f_k^{u_j}\}$ that can serve it. Based on our model assumptions, the cells in F^{u_j} are neighbors of each other.

Algorithm 1: GreenFemto user detection

Input: a new user, u_j, attaching to the network
$\bar{X}$, the current femtocell activation vector
Result: a femtocell is selected for u_j
user u_j is detected by sentinel femtocell f_s;
F^{u_j} := user u_j's active cell list;
B_s := femtocell f_s's neighbors;
if $c_s > b_{sj}(\bar{X})$ **then**
  $c_s -= b_{sj}(\bar{X})$;
  return f_s
if $c_i > b_{ij}(\bar{X})\colon i \in F^{u_j}$ **then**
  f_i := user u_j's selection of best femtocell in F^{u_j};
  $c_i -= b_{ij}(\bar{X})$;
  return f_i
/* at this point, $c_s = b_{sj}(\bar{X})$ and $c_i = b_{ij}(\bar{X})$ $\forall i \in F^{u_j}$ */
wake up neighbors in B_s;
f_n := user u_j's selection of best femtocell in B_s;
$F(u_j) := f_n$;
$t_m := [0, \tau_{max}]$;
schedule redundancy test at t_m;
return $F(u_j)$

If f_s is able to serve the user, that is, $c_s > b_{sj}(\bar{X})$, it decreases its residual capacity c_s and broadcasts an `Info` message with its new capacity. If u_j cannot be served, f_s checks whether u_j can be served by another cell in F^{u_j}, i.e. whether $\exists f_i \in F^{u_j}$ s.t. $c_i > b_{sj}(\bar{X})$. In this case, f_s does not serve the user and the user selects a femtocell to attach to from F^{u_j}. Otherwise, if all the cells in F^{u_j} have insufficient residual capacity, f_s broadcasts a `WakeUp` message to its sleeping neighbors in B_s^o.

Inactive neighbors that receive a `WakeUp` message become active, set their capacities to c^{max}, and broadcast an `Awake` message. Neighboring cells that receive the `Awake` message reply with an `Info` message with their capacity to inform the newly awakened cells of the current state of the network.

Cells receiving `Awake` messages schedule a timeout, t_i, in the interval $(0, \tau_{max}]$, that triggers user reassignment. We assume that timeouts are non-overlapping. As the newly awakened cells receive `Info` messages, they also schedule timeouts. If a new user is detected by an awakened femtocell f_a during its timeout interval, the user detection algorithm runs as normal, but the timeout for f_a is *not* reset. When the timeout occurs, the scheduled user reassignment takes place.

A pseudocode description of the algorithm, as executed in the sentinel femtocell, is in Algorithm 1.

C. User Reassignment

When a timeout occurs for a femtocell f_i, it determines whether it can become inactive by performing a *redundancy test*, described in detail in the next section. A femtocell f_i is redundant if its currently active neighbors B_i^* can both cover f_i's area coverage range, R_i^a, and serve all of f_i's users, U_i. The test begins with each user $u_i \in U_i$ sending f_i

Algorithm 2: The RedundancyTest algorithm

Input: femtocell f_i on which to perform the redundancy test
Result: $\mathcal{T}_f$:= true if f_i is redundant, false otherwise

begin coverage test
  $\mathcal{T}_f^c$:= true;
  B_i^* := femtocell f_i's active neighbors;
  $\mathcal{I}_i$:= femtocell f_i's intersection points with neighbors $f_n \in B_i^*$;
  foreach $p_i \in (R_i \cap \mathcal{I}_i)$ **do**
    if *p_i not covered by a femto $f_n \in B_i^*$* **then**
      $\mathcal{T}_f^c$:= false;

begin reassignment test
  $\mathcal{T}_f^r$:= true;
  perform generalized assignment of users U_i to neighboring femtocells B_i^*;
  if *no feasible solution* **then**
    $\mathcal{T}_f^r$:= false;

return $\mathcal{T}_f \rightarrow \mathcal{T}_f^c \wedge \mathcal{T}_f^r$

the active set of femtocells to which it can connect. If every user responds with at least two femtocells, f_i can determine whether there is enough residual capacity in its neighbors to support a reassignment.

A *user reassignment* is a function $\mathcal{R} : U_i \rightarrow B_i^*$ such that for each user $u \in U_i$, $\mathcal{R}(u)$ is the neighboring cell of f_i to which u_j is reassigned. In Section V-D we present a method for performing a redundancy test and determine a reassignment. If f_i is redundant, it sends a `Sleep` message, which notifies its neighbors of the reassignment and reserves space on the new cell for incoming users, transfers its users to its neighbors according to the reassignment, and finally turns itself off.

When a femtocell f_j receives a `Sleep` message and reassignment $\mathcal{R}(\cdot)$, it updates its stored state of f_i and reserves resources for incoming users according to $\mathcal{R}(\cdot)$. Furthermore, to prevent race conditions, if a timeout for a redundancy test for f_j was active, it is reset. Femtocell f_i then initiates a handoff for the users it is transferring to f_j. This ensures that users will not be handed off to femtocells that no longer have sufficient capacity. The reassignment handoff must occur within a time, T_R, after which users are accepted according to the standard method[2].

D. Redundancy Test

The full redundancy test algorithm is listed in Algorithm 2. Redundancy tests take place when timeouts occur in newly awakened femtocells. A femtocell f_i is redundant if it can be turned off without creating an area coverage hole, and if all the users served by f_i can be reassigned to the other active femtocells in F^*. In the following section we describe how these tests can be performed.

1) Coverage Test: Theorem IV.1 can be used to determine whether a femtocell is redundant in terms of coverage, as stated by the following corollary.

[2]The timeout is introduced to take into account possible changes to the network, such as user movement, which may prevent the application of the reassignment. However, since the reassignment can be performed at a small time scale relative to user movement, these changes are unlikely.

Corollary 1. *A femtocell f_i, with area coverage range R_i^a, is redundant in terms of coverage if the intersection points in $R_i^a \cap P$ are covered by its set of active neighbors B_i^*.*

Note that this test can be performed locally provided that femtocells are aware of the intersection points that lie in their coverage range, as we assume in this paper.

2) User Reassignment: User reassignment can be modeled as a generalized assignment problem [12] with unit costs, where all the users $u_j \in U_i$ on f_i, with respective RB requirements $b_{ij}(\bar{X}) \forall u_j \in U_i$, must be assigned to $|B_i^*|$ neighboring femtocells, with respective residual capacities $c_k, \forall f_k \in B_i^*$. This is an NP-complete problem, therefore there is no exact polynomial-time algorithm. However, there are both efficient, bounded approximation algorithms, and exact algorithms [12] that perform well on small problem sizes. In our experiments, the number of active neighbors $|B_i^*|$ is always less than ten, and the problem can be solved exactly.

E. User Departure and Handoffs

When a user u_j served by femtocell f_i leaves the network, f_i increases its capacity and broadcasts an `Info` message to its neighbors, to alert them of this change. If f_i is not serving any users and is not needed for area coverage, it broadcasts a `Sleep` message and enters the OFF state.

If a mobile user is leaving the coverage area of its attached femtocell, as detected by decreasing signal quality reports, and there are no active femtocells with available capacity to which the user can be handed off[3], the current femtocell broadcasts a `WakeUp` message to its sleeping neighbors. This awakens those neighbors, and allows them to be the target of a handoff from the current femtocell. Specifically, the `WakeUp` message is triggered when the current user's SINR drops below a threshold $t^{handoff}$. $t^{handoff}$ is larger than $t^{coverage}$, so a handoff `WakeUp` is triggered before the user loses coverage.

VI. Algorithm Properties

In this section we prove that GreenFemto converges. We assume that GreenFemto is run with an exact generalized assignment algorithm.

GreenFemto reconfigures the network as long as users move, join or leave the network. In the following we prove that the algorithm converges to a stable configuration after the last dynamic event, provided that no new event occurs during the execution of the algorithm.

Recall that an *activation policy* state is a vector $\bar{X} = \{x_1, \ldots, x_n\}$, where $x_i = 1$ if f_i is active and 0 if it is inactive. In a network with n femtocells and m users, the space of possible activation vectors $\mathbb{X}$ is finite and its size is bounded by $|\mathbb{X}| < 2^n$.

Theorem VI.1. *The algorithm* GreenFemto *converges to a stable configuration.*

[3]Recall that each femtocell is aware of the residual capacity of its active neighbors.

Proof: We define the function $g : \mathbb{X} \rightarrow \mathbb{N}^+$:

$$g(S) = \sum_{i=1}^{n} x_i \tag{14}$$

The function g is trivially lower bounded by 0.

Now consider the three possible network events: joining, leaving, or moving. Events can occur in any order, but we assume that events themselves are atomic. If a femtocell is scheduled for a redundancy test, any further redundancy test timeouts at that femtocell are ignored, therefore events will be sequentialized by the length of each femtocell's initial non-overlapping timeout. Since any combination of events leads to a sequence of redundancy tests, it suffices to show that all events lead to redundancy tests, and redundancy tests cannot change a state S to a state S' such that $g(S) < g(S')$, and if $g(S) = g(S')$, $S = S'$.

Consider a user u_j joining the network. If u_j can be served by the sentinel femtocell f_s, or another currently active femtocell, the algorithm halts and no changes to the network state are made. If u_j cannot be served by any active femtocell, f_s wakes up its inactive neighbors B_s^o, and $g(S)$ increases by $|B_s^o|$. Each femtocell $f_k \in B_s^o$ schedules a timeout for a redundancy test.

Consider a user u_j leaving the network. When u_j leaves the network, one of two cases are possible; redundancy tests are performed in both. If u_j is not the last user on a femtocell f_i, f_i broadcasts an `Info` message. If u_j is the last user on a femtocell f_i, f_i broadcasts a `Sleep` message and disables itself. Neighboring femtocells receiving either message start a timeout for a redundancy test.

Consider a user u_j moving from femtocell f_i to femtocell f_k. The actions of f_i are identical to when a user leaves the network. If f_k has sufficient capacity to serve u_j, u_j is handed over to f_k, and no more state changes occur. If f_k does not have sufficient capacity to serve u_j, neighboring femtocells B_k^o are woken up. The remainder of the proof is identical to when a user joins the network.

Finally, consider a femtocell f_i performing a redundancy test, with initial state S, and final state S'. f_i checks if it is needed for satisfaction of the area coverage constraint and if its users can be reassigned to neighboring femtocells. If one of the two tests fails, f_i remains active, and $g(S) = g(S')$ and $S' = S$. If the tests are successful, f_i is disabled and the new state S' is such that $g(S) > g(S')$. Since the network state space is finite and the function g is lower bounded, GREENFEMTO eventually converges to a stable state where no more state transitions are possible. ∎

We now prove that the final configuration to which GREENFEMTO converges is locally Pareto-optimal, defined as follows:

Definition VI.1 (Local Pareto-Optimality)**.** *A network state S is locally Pareto-optimal if any of the two following conditions holds:*

1) *For all active femtocells $f_i \in F^*$, deactivating f_i violates the area coverage constraint*
2) *For all active femtocells $f_i \in F^*$, there does not exist a reassignment of all the users of f_i to its active neighbors*

A locally Pareto-optimal configuration cannot be unilaterally improved by a femtocell; that is, the energy consumption of the network cannot be reduced by any active femtocell based only on its knowledge and coordination with its immediate neighbors. A global reconfiguration may still reduce the number of active femtocells, but this requires global information on the network state which is not available to individual femtocells.

Theorem VI.2. GREENFEMTO *converges to a locally Pareto-optimal network state.*

Proof: We proceed by contradiction. Let us assume that the final network state S_f is not locally Pareto optimal, i.e. there exists at least one active femtocell f_o such that turning off f_o does not violate the area coverage constraint, and there exists a reassignment of all users of f_o to its active neighbors.

f_o performs a redundancy test if any of three events occurs:

1) f_o detects a new user
2) a user assigned to f_o leaves the network
3) f_o receives an `Info`, `Awake`, or `Sleep` message from one of its neighbors

Consider, T, the most recent redundancy test performed by f_o. Since f_o is active in S_f, T was not successful, either because turning off f_o would violate the area coverage constraint, or because f_o was unable to find a reassignment of all of its users. Therefore, immediately after T, the network state, S_T, is locally Pareto optimal. Since no further redundancy tests occur between T and S_f, we can assume that none of the three events listed above occurred, since if one of the events had occurred, a test T' would have occurred between T and S_f, which violates the assumption that T is the most recent test performed by f_o. Since none of the events occurred, no changes to the local Pareto optimality of the network could have occurred. As a result, f_o cannot be disabled, and S_f is locally Pareto optimal, contradicting the hypothesis. ∎

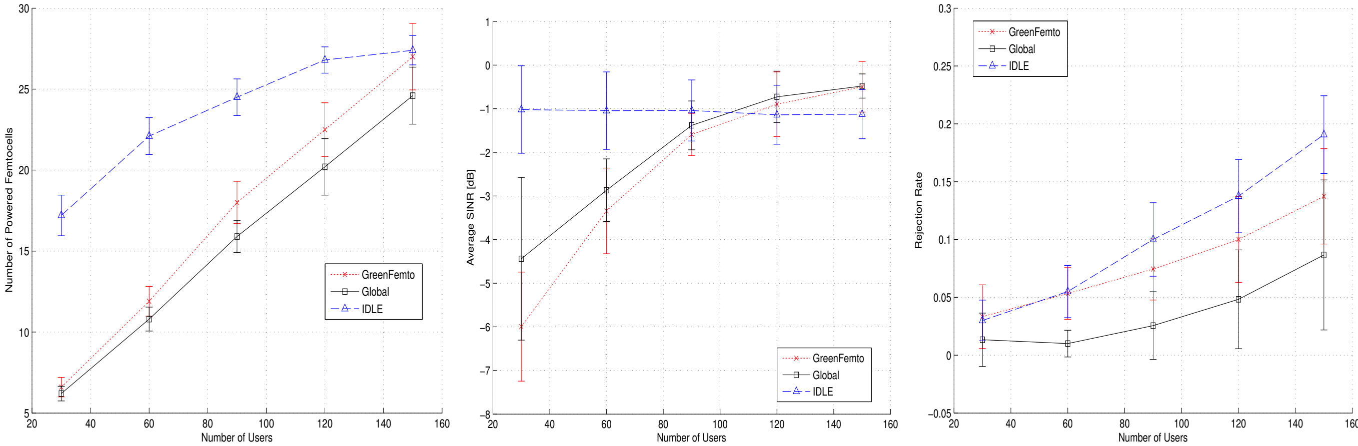
VII. SIMULATION RESULTS

In this section we describe our simulations and results.

A. Simulation Details

We use a Matlab LTE system-level simulator [13] to evalutate GREENFEMTO. A full description of the simulator is available in the reference. Parameters used in the simulation are listed in Table II

We consider a dense femtocell deployment with no underlying macrocell coverage. Femtocells use unidirectional antennas in a one-sector configuration. Since the femtocells are densely deployed, we assume that they are connected to the same local network, through which they can communicate using either a local management server or directly in a peer-to-peer fashion. Users are uniformly randomly distributed across $\mathcal{L}$.



(a) Active femtocells (b) Mean SINR (c) Rejection rate

Fig. 1: 50 femtocells, increasing user scenario

TABLE II: Simulation Parameters

Parameter	Value	Description
$\|\mathcal{L}\|$	$100\ m \times 100\ m$	Area of interest
n_{TX}	2	Number of transmit antennas
n_{RX}	2	Number of receive antennas
f	2140 Mhz	System frequency
b	20 Mhz	System bandwidth
p_M^{tx}	40 dBm	Macrocell transmission power
p_F^{tx}	24 dBm	Femtocell transmission power
μ_s	0 dB	Shadow fading mean
σ_s	10 dB	Shadow fading standard deviation
n_r	9 dB	Receiver noise figure
T_k	-174 dBm/Hz	Thermal noise density
t^a	19 dB (SNR)	Area coverage threshold
t^u	-7 dB (SINR)	User coverage threshold
N	100	Resource blocks per base station

B. Results

We compare three algorithms: GREENFEMTO, GLOBAL, a computed global reassignment optimization based on GreenFemto, and IDLE, which is a modification of an algorithm for femtocell power saving described in [7]. We begin with results on stationary users to highlight the performance of the algorithms as the number of users changes, and discuss mobile users at the end of the section.

GLOBAL uses the Gurobi [14] solver to solve the **Min-Activation** problem in Section III, given the activation vector $\bar{X}$ found by GREENFEMTO and the functions $b_{ij}(\cdot)$ for all femtocells f_i and users u_j as input. It then iterates on this solution until convergence[4]. In the case that the problem is infeasible due to user coverage or femtocell capacity, the users that cannot be served are rejected one-by-one and the problem is reattempted without them until a solution is found. GLOBAL prioritizes serving users over reducing the number of active femtocells: users are only rejected if there is no femtocell that can serve them.

GLOBAL is used to evaluate the femtocell activation and user assignment that GREENFEMTO finds. It finds the optimal assignment of users to femtocells given a set of active femtocells. By using the solution that GREENFEMTO finds as input, we can determine whether there is a global reassignment of users to femtocells that uses fewer femtocells, improves SINR, or rejects fewer users. The performance difference between GLOBAL and GREENFEMTO highlights the impact of limiting GREENFEMTO to local reassignments.

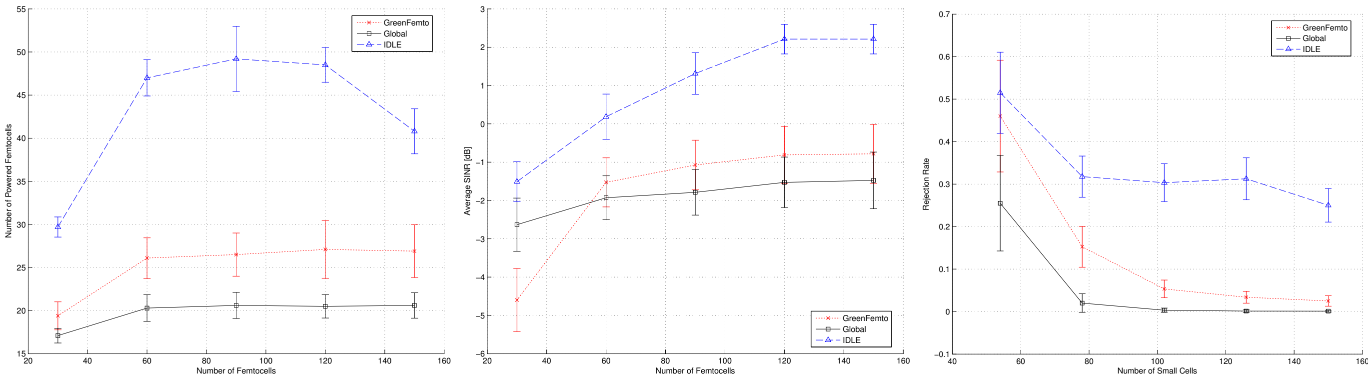
The IDLE algorithm partially disables a femtocell's electronics when no users are present, leaving only a low-power user detection mode active. When a user enters the network, it is detected by all of the femtocells in range. The user connects to the femtocell that can serve it with the highest SINR, while the remaining femtocells turn off. In IDLE, each femtocell is independent and does not communicate with other femtocells on the network. We modify the IDLE algorithm by requiring that area coverage is provided, as in GREENFEMTO.

Figure 1a shows the number of active femtocells, with 50 total femtocells, as the number of users in the system increases. All graphs show the average of 10 runs, with error bars indicating the 95% confidence interval. With 30 users, the number of femtocells activated by GREENFEMTO is dominated by area coverage. The difference between GREENFEMTO and GLOBAL is small, since most of the femtocells activated by GREENFEMTO are also needed for area coverage. IDLE activates three times the number of femtocells as GREENFEMTO.

As the number of users increases, the number of femtocells activated by GREENFEMTO increases linearly, while the number of femtocells activated by IDLE increases more slowly. IDLE turns on the best femtocell for a user, which in most cases is the nearest femtocell. Therefore, IDLE's performance is dominated by the number of femtocells that are the nearest femtocell to a user, which is a factor of the user and femtocell distributions. The gap between the global optimal and GREENFEMTO's solution increases slightly due to the additional degrees of freedom in the ILP problem made available by the activation of more femtocells to cover users.

Figure 1b shows the average user wideband SINR with 100 femtocells total. This SINR measurement is an average of the user SINR measured across the entire channel bandwidth; due

[4]Since **MinActivation** can only activate as many or fewer femtocells as were active in the input activation vector, it is clear to see that it converges.



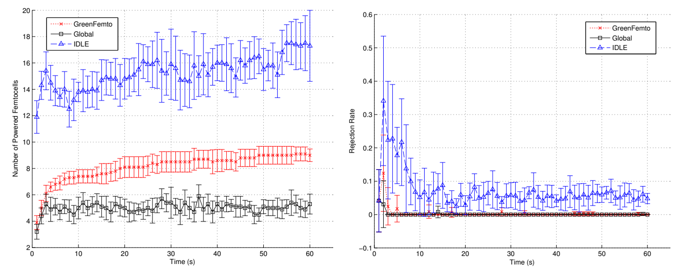
(a) Powered femtocells (b) Mean SINR (c) Rejection rate

Fig. 2: 100 users, increasing femtocell scenario

to small-scale fading, individual subchannels may have SINRs above or below the wideband SINR. The SINR threshold, $t^u = -7$ dB, and all three algorithms find femtocell activations that reach this target on average. Since IDLE activates the best femtocell for each user, it reaches a high, near-constant average SINR that is dependent on the average pathloss between users and femtocells, with interference reducing SINRs as the number of active femtocells increases. SINRs are lower with GREENFEMTO until more femtocells are activated. Since our goal is to reduce the number of active femtocells, the increased SINR of IDLE is wasteful–there is excess capacity that is unused. The difference between GREENFEMTO and GLOBAL can be explained by the path-dependence and local reassignment limitations inherent in the design of GREENFEMTO. GREENFEMTO focuses on reducing femtocell power consumption, therefore GREENFEMTO will not reassign users to improve their SINR.

Figure 1c shows the user rejection rate. IDLE rejects a large number of users because it activates a large number of femtocells when new users enter the network, which increases interference. GLOBAL's performance indicates that for approximately 50% of the users that GREENFEMTO rejects, there exists a way to serve them given the current active set of femtocells, however, it is not possible for GREENFEMTO to find these solutions, because they require global information and reassignment. For example, this could occur if a new user can only be served by one femtocell, which is full of users that could be served on other femtocells. Since GREENFEMTO does not preempt users, the new user is rejected, while an assignment is possible using GLOBAL.

Figure 2a shows the number of active femtocells as the total number of femtocells varies, with a fixed number of users. Below 50 femtocells, the number of active femtocells increases with the total number of femtocells, because not all users can be served, as can be seen in the high rejection rate in Figure 2c. IDLE is unsuitable for scenarios with a large number of femtocells. The rejection rate *increases* with the number of total femtocells, and the number of active femtocells *decreases* due to the increased rejections. The large number of rejections is due to the naive way that IDLE wakes up femtocells to serve



(a) Powered femtocells (b) Rejection rate

Fig. 3: 50 users, mobility scenario

users, with all femtocells that detect the presence of a user waking up, rather than only a sentinel femtocell's neighbors. The larger number of woken up femtocells cause significant co-channel interference, which leads to high user rejections.

Figure 2b shows the average SINR as the total number of femtocells increases. As the number of femtocells increases, the distance between femtocells and users decreases on average. SINRs under IDLE increase because it turns on each user's best femtocell, however, IDLE overshoots the SINR target significantly, leading to excess capacity. There is a significant gap between GREENFEMTO and GLOBAL when relative load is high, despite the larger number of active femtocells under GREENFEMTO. However, as the number of femtocells increases, GREENFEMTO's solutions improve. GREENFEMTO's sensitivity to the total number of femtocells is due to the local search it performs for alternative femtocells. When there are a small number of femtocells, the local search performs poorly.

Figure 3a shows the number of active femtocells with 50 total mobile users joining and leaving the network using a Poisson arrival process with exponential call times and $\lambda = 3$, and moving at 3 m/s using a random waypoint mobility model. This scenario allows the algorithms' behavior in a realistic setting to be analyzed. The number of active femtocells varies significantly with the number of users in the system when using the IDLE algorithm. GREENFEMTO exhibits almost no variation in the number of active femtocells, while using half the femtocells and rejecting fewer users, as seen in Figure 3b.

Overall, GREENFEMTO finds solutions that use significantly fewer femtocells than IDLE, and within 15-50% of a globally optimal reassignment.

VIII. Related work

Femtocell energy consumption has been studied extensively in recent years. Several works design power control schemes for femtocell networks [4], [5], [6]. In particular, [4] proposes a distributed algorithm based on game theory to adjust the transmission power of femtocells. The authors of [5] introduce a distributed algorithm to jointly optimize the power consumption of femtocells and the scheduling order for serving users. The work in [6] studies a Pareto optimal power control and scheduling algorithm which aims at improving the spectral efficiency. The aforementioned papers focus on adjusting transmission power and do not consider the possibility of turning off femtocells to save energy.

The problem of minimizing the energy consumption of a two-tier network composed by a macrocell and several femtocells has been addressed in [15], [16], [17], [18]. The works [15], [16], [18] design centralized algorithms, while in [17] a hierarchical reinforcement learning approach is proposed. Centralized algorithms do not scale as well in dense and dynamic scenarios such as the one considered in this paper. We focus on scenarios with fewer degrees of freedom, in which the femtocell network does not belong to the service provider, but to a different institution which seeks to reduce its own energy expenses.

The problem of reducing the energy consumption of densely deployed femtocell networks through selective activation has been considered in [19], [20], [7]. The authors of [19] propose a low complexity sleeping mechanism which makes use of user traffic prediction. This work is complementary to ours, as traffic prediction can be used to further improve the performance of GreenFemto.

The authors of [21] consider the problem of user scheduling in self-organizing femtocell networks. This approach does not focus on network power consumption; instead it focuses on quality of service. In [22] the authors propose an analytical model to characterize the power consumption of macrocell, microcell, picocell and femtocell based networks. In [23] the authors investigate solutions for reducing the number and size of active macrocell to reduce the network energy consumption. Finally, in [24] the authors study heterogeneous networks with cognitive radio capabilities in order to reduce the energy consumption by exploiting spectrum sharing.

IX. Conclusion and Future Work

This paper introduced GreenFemto, a distributed algorithm that finds a solution to **MinActivation**. We proved that GreenFemto converges to a locally Pareto-optimal solution. We found that GreenFemto reduces the number of active femtocells relative to a previous femtocell power-saving method, finding solutions that use up to 55% fewer femtocells, and comes within 15% a globally optimal solution.

Acknowledgments

Supported by the Leonhard Chair in the College of Engineering at the Pennsylvania State University, and NSF Grant CNS-1218597.

References

[1] Cisco Visual Networking, “Global mobile data traffic forecast update, 2010-2017,” *White Paper, February*, 2013.
[2] Aruba. High-density wireless networks for auditoriums validated reference design. [Online]. Available: http://www.arubanetworks.com/wp-content/uploads/DG_HighDensity_VRD.pdf
[3] “Pctel mobile tower solutions.” [Online]. Available: http://www.antenna.com/apg_product_groups.cgi?id_num=1052
[4] L. B. Le, D. Niyato, E. Hossain, D. I. Kim, and D. T. Hoang, “Qos-aware and energy-efficient resource management in ofdma femtocells,” *IEEE TWC*, vol. 12, no. 1, pp. 180–194, 2013.
[5] G. Cao, D. Yang, and X. Zhang, “A distributed algorithm combining power control and scheduling for femtocell networks,” *IEEE WCNC*, 2012.
[6] H. Burchardt, S. Sinanovic, G. Auer, and H. Haas, “Pareto optimal sinr scheduling for femto-cell deployment in wireless networks,” *IEEE VTC*, 2012.
[7] I. Ashraf, L. Ho, and H. Claussen, “Improving energy efficiency of femtocell base stations via user activity detection,” in *IEEE WCNC*, April 2010, pp. 1–5.
[8] “Fcc test report (part 22) rf130726c18,” Bureau Veritas Consumer Products Services (H.K.) Ltd., Tech. Rep., 2013.
[9] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); LTE physical layer; General description,” 3rd Generation Partnership Project (3GPP), TS 36.201, Mar. 2010. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36201.htm
[10] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration for energy conservation in sensor networks,” *ACM TOSN*, vol. 1, no. 1, pp. 36–72, 2005.
[11] “White paper: Wake on lan technology,” Lieberman Software, Tech. Rep., 2006. [Online]. Available: http://www.liebsoft.com/pdfs/wake_on_lan.pdf
[12] S. Martello and P. Toth, *Knapsack Problems*. John Wiley Sons, Ltd, 1990, ch. 7.
[13] S. Schwarz, J. Ikuno, M. Simko, M. Taranetz, Q. Wang, and M. Rupp, “Pushing the limits of lte: A survey on research enhancing the standard,” *Access, IEEE*, vol. 1, pp. 51–62, 2013.
[14] Gurobi. [Online]. Available: http://gurobi.com
[15] Y.-L. Chung, “Green dynamic configuration in two-tier lte femtocell networks,” *IEEE ICOIN*, 2013.
[16] Z. Pan and S. Shimamoto, “Cell sizing based energy optimization in joint macro-femto deployments via sleep activation,” *IEEE WCNC*, 2013.
[17] X. Chen, H. Zhang, T. Chen, and M. Lasanen, “Improving energy efficiency in green femtocell networks: A hierarchical reinforcement learning framework,” *IEEE ICC*, pp. 2241–2245, 2013.
[18] L. Saker, S.-E. Elayoubi, R. Combes, and T. Chahed, “Optimal control of wake up mechanisms of femtocells in heterogeneous networks,” *IEEE JSAC*, vol. 30, no. 3, pp. 664–672, 2012.
[19] G. Wang, C. Guo, S. Wang, and C. Feng, “A traffic prediction based sleeping mechanism with low complexity in femtocell networks,” *IEEE ICC*, pp. 560–565, 2013.
[20] I. Ashraf, F. Boccardi, and L. Ho, “Power savings in small cell deployments via sleep mode techniques,” *IEE PIMRC*, pp. 307–311, 2010.
[21] H.-L. Chao, S.-H. Wu, Y.-H. Huang, and S.-C. Li, “Cooperative spectrum sharing and scheduling in self-organizing femtocell networks,” *IEEE ICC*, 2014.
[22] A. Mukherjee, S. Bhattacherjee, S. Pal, and D. De, “Femtocell based green power consumption methods for mobile network,” *Elsevier Computer Networks*, vol. 57, no. 1, pp. 162–178, 2013.
[23] C. Khirallah and J. S. Thompson, “Energy efficiency of heterogeneous networks in lte-advanced,” *Springer Journal of Signal Processing Systems*, vol. 69, no. 1, pp. 105–113, 2012.
[24] R. Xie, F. R. Yu, and H. Ji, “Energy-efficient spectrum sharing and power allocation in cognitive radio femtocell networks,” *IEEE INFOCOM*, pp. 1665–1673, 2012.