A Non-stochastic Learning Approach to Energy Efficient Mobility Management

Cong Shen, *Senior Member, IEEE,* Cem Tekin, *Member, IEEE,* and Mihaela van der Schaar, *Fellow, IEEE*

***Abstract*—Energy efficient mobility management is an important problem in modern wireless networks with heterogeneous cell sizes and increased nodes densities. We show that optimization-based mobility protocols cannot achieve long-term optimal energy consumption, particularly for ultra-dense networks (UDN). To address the complex dynamics of UDN, we propose a *non-stochastic* online-learning approach which does not make any assumption on the statistical behavior of the small base station (SBS) activities. In addition, we introduce *handover cost* to the overall energy consumption, which forces the resulting solution to explicitly minimize frequent handovers. The proposed Batched Randomization with Exponential Weighting (BREW) algorithm relies on *batching* to explore in bulk, and hence reduces unnecessary handovers. We prove that the regret of BREW is sublinear in time, thus guaranteeing its convergence to the optimal SBS selection. We further study the robustness of the BREW algorithm to delayed or missing feedback. Moreover, we study the setting where SBSs can be dynamically turned on and off. We prove that sublinear regret is impossible with respect to arbitrary SBS on/off, and then develop a novel learning strategy, called ranking expert (RE), that simultaneously takes into account the handover cost and the availability of SBS. To address the high complexity of RE, we propose a contextual ranking expert (CRE) algorithm that only assigns experts in a given context. Rigorous regret bounds are proved for both RE and CRE with respect to the best expert. Simulations show that not only do the proposed mobility algorithms greatly reduce the system energy consumption, but they are also robust to various dynamics which are common in practical ultra-dense wireless networks.**

***Index Terms*—Energy efficient mobility management, ultra-dense networks (UDN), frequent handover (FHO), non-stochastic learning.**

I. Introduction

The ultra-dense deployment of small base stations (SBS) [1] introduces new challenges to the wireless network design. Among these challenges, mobility management has become one of the key bottlenecks to the overall system performance. Traditionally, mobility management was designed for large cell sizes and infrequent handovers, which works well with the RF-planned macro cellular networks. The industry protocol is simple to implement and offers reliable handover performance [2]. However, introducing SBSs into the network drastically complicates the problem due to the irregular cell sizes, unplanned deployment, and unbalanced load distributions [3]. Furthermore, ultra-dense deployment makes the problem even harder, as user equipments (UE) in ultra-dense networks (UDN) can have many possible serving cells, and mobile UEs may trigger very frequent handovers even without much physical movement. Simply applying existing macro solutions leads to a poor SBS mobility performance. In particular, total energy consumption can be significant when the mobility management mechanism is not well designed [4].

To address these challenges, research on mobility management has recently attracted a lot of attention from both academia and industry [3]. The research has mainly been based on *optimization theory*, i.e., given the various UE and BS information, the design aims at maximizing certain system utility by finding the best UE-BS pairing. The problem is generally non-convex and optimal or suboptimal solutions have been proposed. In [5], a utility maximization problem is formulated for the optimal user association which accounts for both user's RF conditions and the load situation at the BS. For energy efficient user association, [6] aims at maximizing the ratio between the total data rate of all users and the total energy consumption for downlink heterogeneous networks. Althunibat et. al. [7] propose a handover policy in which low energy efficiency from the serving BS triggers a handover, and the design objective is to maximize the achievable energy efficiency under proportionally fair access. Another optimization criterion of minimizing the total power consumption, while satisfying the user's traffic demand for UDN is considered in [8].

These existing solutions have been proved effective for less-densified heterogeneous networks, but they may perform poorly when the network density becomes high. Examples include the so-called frequent handover (FHO), Ping-Pong (PP), and other handover failures (HOF) problems, which commonly occur when the UE is surrounded by many candidate BSs [9]. In this scenario, a UE may select its serving SBS based on some optimization criterion, e.g., best biased signal strength as in 3GPP, or other system metric as in [5], [6], [7], [8]. However, system dynamics such as very small movement of the UE or its surrounding objects can quickly render the solution sub-optimal, triggering user handover procedure in a frequent manner. Probably more important than the loss of throughput, the FHO and PP problems significantly increase the system energy consumption, as much energy is wasted on unnecessary handovers.

These new problems have motivated us to adopt an *online*

C. Shen is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China. E-mail: congshen@ustc.edu.cn.

C. Tekin is with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey, 06800. E-mail: cemtekin@ee.bilkent.edu.tr.

M. van der Schaar is with the Electrical Engineering Department, University of California, Los Angeles (UCLA), USA. E-mail: mihaela@ee.ucla.edu.

The work of C. Shen has been supported by National Natural Science Foundation of China under project 61572455. The work of M. van der Schaar has been supported by NSF grants 1218136 and 1462245.

learning approach, rather than an optimization one, to mobility management. The rationale is that the goal of mobility should not be to maximize the *immediate* performance at the time of handover, as most of the existing handover protocols do. Rather, mobility should build a UE-BS association that maximizes the *long-term* performance. In fact, one can argue that the optimization approach with immediate performance maximization inevitably results in some of the UDN mobility problems such as FHO and PP. This is because the optimization-based solutions depend on the system information, and once the system configuration or the context information evolves, either the previously optimal solution no longer offers optimal performance[1], or a new optimization needs to run which can lead to increased energy consumption when the optimality criterion is frequently broken. Furthermore, optimization-based solutions rely on the accurate knowledge of various system information, which may not be available *a priori* but must be learned over time.

The key challenge for efficient mobility management is the unavailability of accurate information of the candidate SBSs in an *uncertain* environment. Had the UE known *a priori* which SBS offers the best long-term performance, it would have chosen this SBS from the beginning and stuck to it throughout, thus avoiding the frequent handovers which lead to energy inefficiency while achieving optimal energy consumption for service. Without this omniscient knowledge, however, the UE has to balance immediate gains (choosing the current best BS) and long-term performance (evaluating other candidate BSs). Multi-armed bandit (MAB) can be applied to address such exploration and exploitation tradeoff that arises in the mobility learning problem, and there are a few works applying the *stochastic* bandit algorithms [10], [11] to address this challenge. Mobility management in a heterogeneous network with high-velocity users is considered in [10], where the solution uses stochastic MAB theory to learn the optimal cell range expansion parameter. In [11], the authors propose a stochastic MAB-based interference management algorithm, which improves the handover performance by reducing the effective interference.

There are three major issues in applying a stochastic bandit approach to the considered mobility management problem. Firstly, one must be able to assume that there exists a well-behaved stochastic process that guides the generation of the reward sequence for each SBS. In practice, however, it is difficult to unravel such statistical model for the reward distributions at SBS. Practical wireless networks with a moderate number of nodes or users are already complex enough that simple stochastic models, as often used in stochastic bandit algorithms, cannot accurately characterize their behavior. Another problem is that the time duration within which a particular statistical model may be adequate is short due to high UDN system dynamics. As a result, there may not exist enough time to learn which statistical model to adopt, let alone utilize it to achieve optimal performance. Furthermore, in a practical system, there may be multiple UEs being served by one SBS, and the energy consumption depends not only on the SBS activity but also on the activities of other UEs, including their time-varying mobility decisions, traffic load, service requirement, etc. As a result, an *uncontrolled* stochastic process cannot adequately capture practical interactions between the UEs and the SBSs, and probabilistic modelling may not accurately match the real-world UDN energy behavior. Second, the majority of the stochastic MAB literature considers reward sequences that are generated by either an *independent and identically distributed* (i.i.d.) process [12], [13], or a *Markov* process [14]. These restrictions may not accurately capture the SBS/UE behavior, resulting in a mismatch to the real-world performance. Lastly, the existing solutions cannot solve the FHO problem, because they do not consider the additional loss incurred when the UE performs handover from one SBS to another. In fact, most of the stochastic bandit solutions incur fairly frequent "exploration" operations, which directly lead to the FHO problem.

Due to the aforementioned challenges that cannot be easily addressed by a stochastic approach, we opt out from using the stochastic MAB formulation. In this work, we solve the UDN mobility problem with the objective of minimizing long-term energy consumption, by using a *non-stochastic* model. Specifically, we do not make any assumption on the statistical behavior of the SBS activities. Instead, the energy consumption of any SBS is allowed to vary arbitrarily. This is a fundamental deviation from the previous *stochastic* solutions. A comparison of this work to the existing literature is provided in Table I. Note that the non-stochastic MAB problem is significantly harder than the stochastic counterpart due to the adversarial nature of the loss sequence generation. We develop a new set of algorithms for loss sequences with switching penalties, delayed or missing feedback, and dynamic on/off behavior, and proved their effectiveness with both rigorous regret analysis and comprehensive numerical simulations.

The main contributions of this paper are summarized below.

- We propose a *non-stochastic* energy consumption framework for mobility management. To the best of the authors' knowledge, this is the first work that applies the non-stochastic bandit theory to address mobility management in wireless networks.
- We explicitly add the *handover cost* to the utility function to model the additional energy consumptions due to handovers, and thus force the optimal solution to minimize frequent handovers.
- We present a Batched Randomization with Exponential Weighting (BREW) algorithm that addresses the frequent handover problem and achieves low system energy consumption. The performance of BREW is rigorously analyzed and a finite-time upper bound for performance loss due to learning is proved. We further study the effect of delayed or missing feedback and analyze the performance impact.
- We analyze the dynamic SBS on/off model and prove that sublinear regret is impossible for arbitrary SBS on/off. To solve this challenging problem, we create a novel strategy set, called *ranking expert*, that is used

[1]The industrial intuition is that good performance at the time of handover should carry over for the near future until the next handover is triggered. However, this is no longer valid with UDN, in which RF and load conditions can change dynamically and FHO/PP needs to be avoided.

TABLE I
COMPARISON OF OUR WORK WITH EXISTING SOLUTIONS.

	[5], [7], [6], [8]	[10], [11]	This work
Design tool	Optimization	Stochastic learning	Non-stochastic learning
Optimize for energy consumption	No	Yes	Yes
Distributed solution	No, except [5]	Yes	Yes
Solve FHO	No	No	Yes
Forward-looking	No	Yes	Yes
Robustness	Not Considered	Not Considered	Considered
Performance study	Analysis & Simulation	Simulation	Analysis & Simulation

in conjunction with a BREW-type solution with respect to expert advice. The novelty of the expert construction is that *it simultaneously takes into account both the handover cost and the availability of SBS*. The regret upper bound with respect to the best expert advice is proved.

The rest of the paper is organized as follows. The system model is presented in Section II. Section III discusses the non-stochastic learning approach for mobility management, including the BREW algorithm in Section III-B, regret analysis in III-C, robustness in Section III-D, and performance analysis of the industry solutions in Section III-E. Dynamic SBS presence is studied in Section IV. Simulation results are presented in Section V. Finally, Section VI concludes the paper.

II. System Model

A. Network Model

An ultra-dense cellular network with N small base stations (SBS) and M user equipments (UE) is considered in this work. We denote the SBS set as $\mathcal{N}_{\text{SBS}} = \{1, \cdots, N\}$. We are mostly concerned with stationary or slow moving UEs, which represents a typical indoor scenario where about 80% of the total network traffic occurs [15]. A representative UE in UDN may have multiple SBSs as the potential serving cell, but needs to choose only one serving SBS. In other words, advanced technologies that allow for multiple serving cells are not considered. One exemplary system is illustrated in Fig. 1, where UE 1 may discover up to 6 candidate SBSs in its neighborhood, possibly with very similar signal strength or load conditions. The mobility management system makes decisions on which UE is idly camped on (idle mode mobility) or actively served by (connected mode mobility) which SBS at any given time.

The mobility decision is traditionally made at the SBS (e.g., X2 handover in LTE) or Evolved Packet Core (e.g., S1 handover at the Mobility Management Entity in LTE). Recently, there has been an emerging trend of designing user-centric mobility management, particularly for the future 5G standard [16]. In this work, we consider user-centric mobility management and let the UE make mobility decisions. We assume that mobility management is operated in a synchronous time-slotted fashion. It is worth noting that we do not make any assumption on whether the candidate SBSs are operating in the same channel or different channels, as our work applies to both of these deployments.

The sequence of operations within each slot can be illustrated in Fig 2. Specifically, at the beginning of a slot,

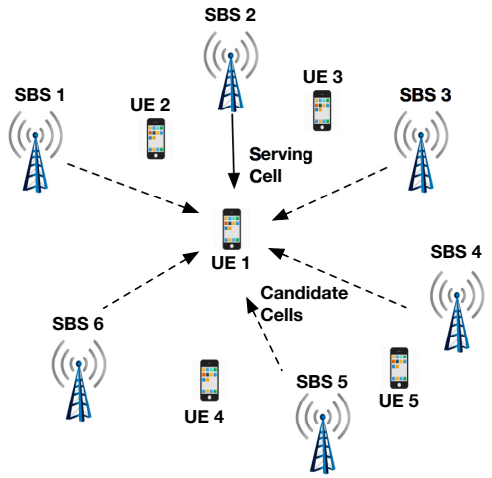


Fig. 1. An illustration of mobility management in UDN.

the UE chooses its serving SBS and starts a downlink data transmission with the paired SBS. Upon the completion of the time slot, the UE can observe the total energy consumed over this slot, and the operation repeats in the next slot.

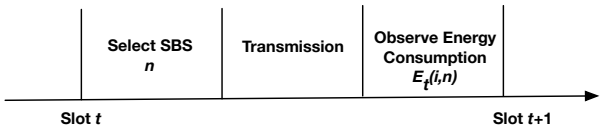


Fig. 2. The mobility management operation for UE i in time slot t.

B. Non-stochastic Energy Consumption Model

We study the mobility problem with minimizing long-term energy consumption as the system design objective, and adopt a *non-stochastic* modelling of the energy consumption of each SBS. Specifically, SBS n, $1 \leq n \leq N$ incurs a total energy consumption $E_t(i,n)$ if it serves UE i at time slot t, $1 \leq t \leq T$. It is assumed that $E_t(i,n) \in [E_{\min}, E_{\max}]$. In this work, we make no statistical assumptions about the nature of the process that generates $E_t(i,n)$. In fact, we allow $\{E_t(i,n)\}$ to be any arbitrary sequence for any n and any i. This is a fundamental difference to the stochastic MAB based mobility solutions [10], [11]. Note that $E_t(i,n)$ may include additional energy consumptions if the UE decides to switch from one SBS to another. The set of energy consumptions $\{E_t(i,n), n = 1, \cdots, N; t = 1, \cdots, T\}$ is unknown to UE i. We are interested in finding a SBS selection sequence

$\{a_{i,t}, t = 1, \cdots, T\}$ for UE i that minimizes the total energy consumption over T slots: $\sum_{t=1}^{T} E_t(i, a_{i,t})$.

III. BREW: A NON-STOCHASTIC MOBILITY MANAGEMENT ALGORITHM

A. Problem Formulation with Handover Cost

We take a representative UE and drop the UE index i from the notation. In the non-stochastic multi-armed bandit model, each arm n corresponds to a SBS for which there exists an arbitrary sequence of energy consumptions up to time T if the UE is served by this SBS. Let a_t denote the SBS selected by the UE at time t. It is assumed that after time slot t, the UE only knows the energy consumptions $E_1(a_1), \cdots, E_t(a_t)$ of the previously selected SBS $a_1, \cdots, a_t$. In other words, the UE does not gain any knowledge about the SBSs which it does not choose. Thus, the UE's mobility algorithm can be stated as selecting a sequence $a_1, \cdots, a_T$ where a_t is a mapping from the previous actions[2] and the corresponding observed energy consumptions from time 1 to $t-1$ to the selection of a SBS at t. Note that the knowledge of past SBS activities can be practically enabled by leveraging the "UE History Information" element in the 3GPP LTE specs [17].

At each time slot $t = 1, 2, \cdots, T$, the UE chooses SBS a_t from $\mathcal{N}_{\text{SBS}}$, and then observes an energy consumption $E_t(a_t)$ for data transmission, which is sent to the UE as feedback from the SBS. For an arbitrary sequence of energy consumptions $\{E_t(a_t)\}$ and for any $T > 0$, we denote

$$E_{\mathbf{a},T,1} \doteq \sum_{t=1}^{T} E_t(a_t) \tag{1}$$

as the total energy consumption without considering any handover cost, at time T of policy $\mathbf{a}$. Note that $E_{\mathbf{a},T,1}$ captures the total energy consumption up to T, corresponding to the service the UE receives from its (possibly varying) serving SBS. Clearly, $E_{\mathbf{a},T,1}$ depends on the arbitrary energy consumption sequences at each SBS as well as the UE actions. We refer to $E_{\mathbf{a},T,1}$ as the total *service energy consumption*.

In practice, switching from one SBS to another incurs additional cost, and frequent switching incurs large energy consumption that is not captured by the service energy consumption $E_{\mathbf{a},T,1}$. To address this issue, we explicitly add additional energy consumption whenever a handover occurs, and thus force the optimal solution to minimize frequent handovers. For simplicity, we assume that a homogeneous energy consumption $E_s \geq 0$ is incurred whenever a UE is handed over from one SBS to another. This cost includes all energy consumptions that are associated with handovers, such as sending additional overhead signals and forwarding UE packets. The total *handover energy consumption* can be computed as

$$E_{\mathbf{a},T,2} = E_s \sum_{n=1}^{N} \sum_{t=2}^{T} \mathbb{1}_{\{a_t = n, a_{t-1} \neq n\}} \tag{2}$$

[2]An SBS selection is also referred to as an *action*.

where $\mathbb{1}_A$ is the indicator function for event A. As opposed to the service energy consumption, the handover energy consumption only depends on the UE action $\mathbf{a}$.

Finally, the total energy consumption over T slots with handover cost can be written as

$$E_{\mathbf{a},T} = E_{\mathbf{a},T,1} + E_{\mathbf{a},T,2}, \tag{3}$$

and we are interested in finding a mobility management policy that minimizes $\mathbb{E}[E_{\mathbf{a},T}]$. It is worth noting that by including the handover cost (2) in the total energy consumption (3), a good handover algorithm not only has to balance the tradeoff between exploitation and exploration, but also needs to minimize the number of occurrences that the UE changes SBS associations. Hence, the FHO problem is implicitly solved when the UE total energy consumption is minimized.

B. The BREW Algorithm

To simplify the analysis, we assume without loss of generality that $E_{\min} = 0$, and both $E_{\max}$ and E_s are normalized as $E_{\max} + E_s = 1$. Thus, if we re-write the total energy consumption of selecting SBS a at time slot t as $\tilde{E}_t(a) := E_t(a) + E_s \mathbb{1}_{\{a_t \neq a_{t-1}, t>1\}}$, we have $\tilde{E}_a(t) \in [0, 1]$. We also assume that the energy consumption for SBS a is arbitrary but oblivious. In practice, this assumption is valid when the service energy consumption $E_t(a)$ of UE selecting SBS a at time t only depends on the current state of SBS a, such as its user load and traffic load, minimum transmit power to satisfy UE's QoS, etc. In other words, we do not consider the case that the SBS intelligently manipulates its service energy consumption to counter the UE mobility policy it learns from the past.

The proposed Batched Randomization with Exponential Weighting (BREW) solution is given in Algorithm 1. As the name suggests, it is a batched extension of the exponential weighting algorithm such as the celebrated EXP3 [18]. Note that in Algorithm 1, the EXP3 component is an adaptation of the algorithm originally proposed in [18], which uses a slightly different weighing scheme and works with loss functions instead of reward functions [19] to get rid of the uniform mixture term in the probabilistic decision rule of the EXP3 in [18]. We highlight several key design considerations. Firstly, because the energy consumption of each SBS can be generated arbitrarily, it is easy to show that for any *deterministic* mobility solution, there exist sequences of energy consumption that make the solution highly sub-optimal. In other words, no fixed algorithm can guarantee a small performance degradation against *all* possible energy consumption sequences. Hence, for the non-stochastic mobility problem, we introduce *randomization* in the proposed algorithm to avoid being stuck in a worst-case energy consumption. This is done by selecting SBS based on a probability distribution over N SBSs.

Subsequently, a natural question is what type of randomization one should introduce to achieve good energy consumption performance. We note that our mobility management problem can be viewed as a special case of *optimal sequential decision for individual sequences* [20], for which *exponential weighting* is a fundamental tool. The proposed algorithm uses exponential weighting to construct and update the probability for choosing SBS, as shown in (7).

Algorithm 1: The BREW mobility management algorithm.

Input : A non-increasing sequence $\{\gamma_l\}_{l\in\mathbb{N}}$, $\tau \in \mathbb{N}_+$
Initialize: $p_a(l) = 1/N$ and $\hat{L}_0(a) = 0$ for all $a \in \mathcal{N}_{\text{SBS}}$
while $l \geq 1$ **do**
Select SBS $a(l)$ randomly according to the probabilities $\{p_a(l)\}$, $a \in \mathcal{N}_{\text{SBS}}$
Keep UE on SBS $a(l)$ for the next τ time slots: $(l-1)\tau+1, \ldots, l\tau$
Observe total energy consumption $\{\tilde{E}_t(a(l))\}_{t=(l-1)\tau+1}^{l\tau}$, possibly including a one-time handover energy consumption E_s at $(l-1)\tau+1$
Calculate the average energy consumption incurred in batch l:

$$\bar{E}_t(a(l)) = \frac{1}{\tau} \sum_{t=(l-1)\tau+1}^{l\tau} \tilde{E}_t(a_t) \tag{4}$$

Calculate the estimated energy consumption of each $a \in \mathcal{N}_{\text{SBS}}$ in the batch

$$\hat{E}_l(a) = \frac{\bar{E}_t(a(l))}{p_a(l)} \mathbb{1}_{\{a=a(l)\}} \tag{5}$$

Update the cumulative estimated energy consumption of each $a \in \mathcal{N}_{\text{SBS}}$

$$\hat{L}_l(a) = \hat{L}_{l-1}(a) + \hat{E}_l(a) \tag{6}$$

For $a \in \mathcal{N}_{\text{SBS}}$ set

$$p_a(l+1) = \frac{\exp(-\gamma_l \hat{L}_l(a))}{\sum_{a' \in \mathcal{N}_{\text{SBS}}} \exp(-\gamma_l \hat{L}_l(a'))} \tag{7}$$

$l = l + 1$
end

Finally, in order to address the FHO problem and avoid incurring large accumulated handover energy consumption, we need to "explore in bulk". This is done by grouping time slots into batches and not switching within each batch. What separates the operations within a batch from outside is that the UE does not observe energy consumption on a per-slot basis and does not need to update the internal state. In general, BREW works as if the UE is unaware that a batch has happened as opposed to one time slot. At the end of the batch, though, the UE can receive a one-time energy consumption feedback, which is the average energy during the batch as shown in equation (4). The choice of the batch length plays a critical role in the overall performance – if it is too large, one may get the benefit of having little loss from the handover energy consumption, but also may stuck at a sub-optimal SBS for a long time, and vice versa. The BREW algorithm uses a parameter τ that determines the batch length.

C. Finite-Time Performance Analysis

To evaluate the performance of the proposed BREW mobility solution, we adopt a *regret* formulation that is commonly used in multi-armed bandit theory [21]. Specifically, we compare the energy consumption of the BREW algorithm with a "genie-aided" solution where UE chooses the SBS which has the minimum total energy consumption over T slots, i.e., chooses the best SBS with the minimum $E_{\mathbf{a},T,1}$ up to time T and incurs no handover cost $E_{\mathbf{a},T,2} = 0$. Our goal is to characterize the energy consumption regret for any finite time T. The smaller this regret is, the better the solution.

Formally, we define

$$E_{\text{best}} \doteq \min_{n \in \mathcal{N}_{\text{SBS}}} \sum_{t=1}^{T} E_t(n) \tag{8}$$

as the energy consumption of the single best SBS at time T. Then, the performance of any UE mobility solution $\mathbf{a}$ can be measured against the genie-aided optimal policy (8) in expectation. We formally define the regret of a UE mobility solution $\mathbf{a}$ as:

$$R_{\mathbf{a}}(T) := \sum_{t=1}^{T} \mathbb{E}\left[E_t(a_t) + E_s \mathbb{1}_{\{a_t \neq a_{t-1}, t>1\}}\right] - E_{\text{best}}, \tag{9}$$

which is the difference between the total energy consumption of the learning algorithm and the total energy consumption of the best fixed action by time T. Here a_t denotes the action chosen by the UE at time slot t and the expectation is taken over the randomization of the UE's algorithm.

Note that $R_{\mathbf{a}}(T)$ is a non-decreasing function of T. For any mobility algorithm to be able to learn effectively, $R_{\mathbf{a}}(T)$ has to grow *sublinearly* with T. In this way, one has $\lim_{T\to\infty} R_{\mathbf{a}}(T)/T = 0$, indicating that asymptotically the algorithm has no performance loss against the genie-aided solution. For the BREW mobility solution, we have the following theorem that upper bounds its regret for any finite time T.

Theorem 1. *For a given time horizon T, when BREW (Algorithm 1) runs with $\gamma_l = \sqrt{(2\log N)/(lN)}$ and batch size $\tau = \lceil B_N T^{1/3} \rceil$, where $B_N = (4.5N\log N)^{-1/3}$, its regret is bounded by*

$$R_{\mathbf{a}}(T) \leq 2B_N^{-1} T^{2/3} + \left(B_N + B_N^{-2}\right) T^{1/3} + 1. \tag{10}$$

Proof. See Appendix A. □

Theorem 1 provides a sublinear regret bound for BREW that guarantees the long-term optimal performance. Moreover, the bound in Theorem 1 applies to any finite time T, and can be used to characterize how fast the algorithm converges to the optimal action. Specifically, the total energy consumption of the UE that uses the BREW algorithm will approach, on average, the total energy consumption of the best fixed action $a \in \mathcal{N}_{\text{SBS}}$ at a rate no slower than $O(T^{-1/3})$. Although the BREW algorithm works for any non-increasing sequence of γ_l, the particular choice of $\gamma_l = \sqrt{(2\log N)/(lN)}$ gives a guaranteed upper bound of the regret in (10).

Another important remark is that the choice of τ given in Theorem 1 is optimal in terms of the time order of the regret, which is $O(T^{2/3})$. It is shown in [22] that for any learning algorithm, there exists a loss sequence under which that learning algorithm suffers $\tilde{\Omega}(T^{2/3})$ regret. Hence, our choice of τ in Theorem 1 results in a regret upper bound that matches the lower bound of $O(T^{2/3})$, proving its optimality

in terms of the time order. A smaller value for τ will make BREW incur a higher cost due to over-switching, while a larger value for τ will make the algorithm incur a higher cost due to staying too long on a suboptimal action.

D. Robustness Analysis

The development of BREW for energy-efficient mobility management captures the basic characteristics of the user mobility model in UDN, particularly with the inclusion of handover cost as well as the non-stochastic nature of the approach. However, in a real-world deployment, robustness issues often arise, such as delayed or missing feedback of the energy consumption. Whether the BREW algorithm can handle these problems and its impact on the total energy consumption is essential for its practical deployment.

1) Delayed Feedback: Delayed feedback constantly happens in practice. For example, in most cellular standards, sending the feedback to UE may be delayed because it has to wait for the frames that are dedicated to sending control and signalling packets.

The next theorem provides a regret bound when the UE receives the information about the energy consumption with a delay of d time slots.

Theorem 2. *Consider the UE-SBS mobility operation in Fig. 2 where a feedback at time slot t is d-delayed energy consumption $E_{t-d+1}(i,n)$, $d \geq 1$. For a given time horizon T, when BREW (Algorithm 1) runs with $\gamma_l = \sqrt{(2\log N)/(lN)}$ and batch size $\tau = \lceil B_N T^{1/3} \rceil$, where $B_N = (4.5N\log N)^{-1/3}$, its regret is bounded by*

$$R_{\mathbf{a}}(T) \leq (d+1)B_N^{-1}T^{2/3} + \left(B_N + B_N^{-2}\right)T^{1/3} + 1. \quad (11)$$

Proof. See Appendix B. □

The importance of Theorem 2 is that it preserves the order-optimality of the BREW algorithm in the presence of delayed feedback. In fact, comparing (11) to (10), we can see that only the coefficient before $T^{2/3}$ is increased due to the delayed feedback. Moreover, Theorem 2 holds when $d < B_N T^{1/3}$, implying that as long as the delay is sublinear in time with a small enough time exponent, the regret can be guaranteed to be sublinear in time. Finally, we note that in practice, the delay of feedback may be time-varying, and Theorem 2 can be applied to varying delays where d is chosen as the maximum feedback delay.

2) Missing Feedback: Another practical issue with respect to UE observing the energy consumption is that such feedback from the SBS to UE may be entirely missing. This can happen when the downlink transmission that carriers the feedback information is not correctly received at the UE.

We take a probabilistic approach when considering the missing feedback problem. Specifically, we assume that at each time slot t, the energy consumption feedback may be missing with probability p, which can be set, e.g., as the packet loss rate for the transmission.

Algorithm 2 is the modified BREW that can handle the missing feedback. The idea behind Algorithm 2 is to make the estimated energy consumption $\hat{E}_l(a)$ an *unbiased* estimate of the actual average energy consumption in batch l. Therefore, we normalize this quantity by dividing it with the probability that ζ feedbacks are observed in batch l. With this approach, the contribution of rare events (unlikely feedback sequences) to the cumulative estimated energy consumption is magnified. This idea is widely used in the design of exponential weighing algorithms and their variants [20].

Algorithm 2: The modified BREW algorithm for missing energy consumption feedback with probability p.

Input : A non-increasing sequence $\{\gamma_l\}_{l\in\mathbb{N}}$, $\tau \in \mathbb{N}_+$; probability p that feedback will be missing in each time slot.

Initialize: $p_a(1) = 1/N$ and $\hat{L}_0(a) = 0$ for all $a \in \mathcal{N}_{\text{SBS}}$

while $l \geq 1$ **do**

Select SBS $a(l)$ randomly according to the probabilities $\{p_a(l)\}$, $a \in \mathcal{N}_{\text{SBS}}$

Keep UE on SBS $a(l)$ for the next τ time slots: $(l-1)\tau+1, \ldots, l\tau$

Let $\mathcal{O}(l)$ be the set of time slots in $\{(l-1)\tau+1, \ldots, l\tau\}$ for which the feedback is observed

Observe total energy consumption $\{\tilde{E}_t(a(l))\}_{t\in\mathcal{O}(l)}$, possibly including a one-time handover energy consumption E_s at $(l-1)\tau+1$

Calculate the average energy consumption incurred in batch l:

$$\bar{E}_l = \frac{1}{|\mathcal{O}(l)|}\sum_{t\in\mathcal{O}(l)} \tilde{E}_t(a(l)) \quad (12)$$

Calculate the estimated energy consumption of each $a \in \mathcal{N}_{\text{SBS}}$ in the batch

$$\hat{E}_l(a) = \frac{\bar{E}_l}{p_a(l)\binom{\tau}{\zeta}(1-p)^\zeta p^{\tau-\zeta}} \mathbb{1}_{\{a=a(l)\}} \mathbb{1}_{\{|\mathcal{O}(l)|=\zeta\}} \quad (13)$$

Update the cumulative estimated energy consumption of each $a \in \mathcal{N}_{\text{SBS}}$

$$\hat{L}_l(a) = \hat{L}_{l-1}(a) + \hat{E}_l(a) \quad (14)$$

For $a \in \mathcal{N}_{\text{SBS}}$ set

$$p_a(l+1) = \frac{\exp(-\gamma_l \hat{L}_l(a))}{\sum_{a'\in\mathcal{N}_{\text{SBS}}} \exp(-\gamma_l \hat{L}_l(a'))} \quad (15)$$

$l = l+1$

end

Unfortunately, we are not able to prove a regret bound that grows sublinearly in time with respect to the best fixed action. Intuitively, for large T there will be approximately pT time slots where no feedback is received. Since we are dealing with the non-stochastic bandit problem, the worst-case loss for these slots can be linear with T even when T is large. This is a significant difference to the stochastic case where when T is large, the concentration property can guarantee a sublinear loss in time.

E. Analyze the 3GPP Mobility Protocols Under the Online Learning Framework

The proposed BREW mobility algorithm and its variations are developed under an online learning framework. In this section, we put the existing industry mobility mechanism under the same framework and characterize its regret performance through the lens of MAB.

The handover mechanism defined in 3GPP centers on the UE measuring the signal quality from candidate SBSs. Such measurements are generally filtered at the UE to rule out outliers. The original 3GPP handover protocol chooses the SBS with the highest signal quality to serve the UE, and sticks with the choice until some performance metric (such as RSRP or RSRQ when LTE is used [2]) drops below a threshold, at which time the UE measures all candidate SBSs again and hands over to the best neighbor.

The original protocol is optimization-based and has no restrictions on how frequent handovers can happen. Recognizing that FHO can happen when the network density is high, there have been some proposals in 3GPP to modify the handover parameters when frequent handovers are observed. The general principle is to first determine whether a FHO problem has happened, typically by counting the number of handovers within a sliding time window. If it is determined that there are too many handovers, mobility parameters such as hysteresis margin and time-to-trigger are modified to "slow down" future handovers, thus avoiding FHO and making the current serving SBS more "sticky".

The original handover mechanism and its variation can be viewed as a myopic rule that is both *greedy* (always select the SBS that is immediately the best) and *conservative* (only take actions when the selected SBS becomes bad enough). The following proposition shows the sub-optimality of these approaches.

Proposition 1. *Under both stochastic and non-stochastic MAB models for SBS energy consumption, the original and enhanced 3GPP handover protocols describe above achieve an asymptotic regret of $O(T)$.*

Proof. See Appendix C. □

Proposition 1 proves a linear regret with respect to time T for the 3GPP handover solutions. Two important remarks are in place. Firstly, Proposition 1 is a strong result in the sense that it is proved for both stochastic and non-stochastic energy consumption, meaning that linear regret is inevitable regardless of the adopted model. Secondly, the sub-optimality is shown without considering the handover cost. In other words, existing industry mechanisms cannot converge to the best SBS even when $E_s = 0$. A non-zero E_s will further deteriorate the regret performance. Detailed numerical comparisons will be made in Section V.

IV. Dynamic SBS Presence

In this section we consider energy efficient mobility management for SBSs with dynamic presence. Notably, this is a new problem that arises with the increase of user-deployed SBSs. For both enterprise and residential small cell deployment, SBSs can be turned on and off by users, thus creating problems for mobility management. In particular, such on/off behaivor would disrupt the learning process. To capture this uncontrolled user behavior, we consider the following generic SBS on/off model. At each time slot t, a subset of SBSs, chosen *arbitrarily*, can be turned off and hence cannot serve the UE. As we will see, this problem is significantly harder. There are known results in the literature of *stochastic* multi-armed bandits with appearing and disappearing arms [23], [24], but the theoretical structure of these solutions are very different from the *non-stochastic* problem in this paper. Consequently, we have to develop new results in the non-stochastic bandit theory and design robust mobility management solutions.

The set of SBSs available at time slot t is denoted by $\mathcal{N}_t \subset \mathcal{N}_{\text{SBS}}$. An SBS in $\mathcal{N}_t$ is called an *active* SBS, while an SBS in $\mathcal{N}_{\text{SBS}} - \mathcal{N}_t := \{n : n \in \mathcal{N}_{\text{SBS}}, n \notin \mathcal{N}_t\}$ is called an *inactive* SBS. We require that the UE only selects from the active SBS set $\mathcal{N}_t$ at time slot t.

We first give an impossibility result regarding achieving the optimal performance asymptotically. Theorem 3 shows that in general it is impossible to obtain sublinear regret when $\mathcal{N}_t$ changes in an arbitrary way.

Theorem 3. *Assume that $\{\mathcal{N}_t\}_{t=1}^T$ is generated by an adaptive adversary which selects $\mathcal{N}_t$ based on the action chosen by the learning algorithm at time slot $t-1$, and $\{E_t\}_{t=1}^T$ is generated by an oblivious adversary. Then, for $E_s \geq E_{\max} + 1/(N-1)$, no learning algorithm can guarantee sublinear regret in time.*

Proof. See Appendix D. □

In light of the impossibility result in Theorem 3, the pursue of good energy consumption performance in the presence of dynamic SBS on/off is only viable when the generation of $\mathcal{N}_t$ is constrained. In the following discussion, we will focus on an *i.i.d. SBS activity model*, where SBS a is present at time slot t with probability p_a independently from other time slots and other SBSs. We emphasize that the i.i.d. activity model generally represents a case that is worse than practice, where the SBS on/off introduced by end-users has some memory. The correlation over time can be exploited by a learning algorithm to achieve better mobility. We focus on the i.i.d. activity model because it presents a more challenging SBS dynamic, and the resulting algorithms and regret analysis can serve as a guideline to the real-world SBS on/off performance. The proposed algorithms for the i.i.d. model can be applied to other SBS activity models, such as the Markov model. Furthermore, note that both i.i.d. and Markov models are widely used in stochastic multi-armed bandit, but in our paper they are used for modelling the SBS on/off activities, not the reward distribution.

In order to address the SBS dynamics, we follow the general principle of *prediction with expert advice* [25]. In this setting, we assume that there is a set of experts which recommend the SBS for the UE to select, based on the past sequence of selections and energy consumption feedback to the UE. There is no assumption on the way these experts compute their predictions, and the only information UE receives from

these experts is the advice. Then, we will bound the regret of the UE with respect to the best of these experts.

A. Ranking Expert

The first proposed algorithm utilizes a concept called "ranking expert" [26] and the algorithm consists of two key elements. Firstly, we will need an efficient expert selection procedure that chooses the action based on all expert advices. In order to achieve low regret, a modified EXP4 procedure from [18] is used to select the expert at each time slot. The second component is how to construct expert advice, where we use *ranking* to sort the possible actions at each expert. The overall Ranking Expert (RE) algorithm is given in Algorithm 3.

Algorithm 3: The Ranking Expert (RE) mobility management algorithm.

Input: A non-increasing sequence $\{\gamma_t\}_{t\in\mathbb{N}}$; $\tilde{\mathcal{E}} = \mathcal{E} \cup \text{Unif}$ (see Definition 2)

Initialize: $q_e(1) = 1/|\tilde{\mathcal{E}}|$ and $\hat{L}_0(e) = 0$ for all $e \in \tilde{\mathcal{E}}$, $a_0 = \text{Rand}(\mathcal{N}_{\text{SBS}})$

while $t \geq 1$ **do**

Observe a_{t-1} and $\mathcal{A}_t$

Get ranking expert advices $\{\boldsymbol{\delta}^e(t)\}_{e\in\tilde{\mathcal{E}}}$

Set

$$p_a(t) = \sum_{e\in\tilde{\mathcal{E}}} q_e(t)\delta_a^e(t) \tag{16}$$

Select a_t randomly according to the probabilities $p_a(t)$, $a \in \mathcal{N}_{\text{SBS}}$

Observe energy consumption $\tilde{E}_t(a_t) \in [0,1]$

Set

$$\hat{X}_t(a) = \begin{cases} \tilde{E}_t(a)/p_a(t) & \text{if } a = a_t \\ 0 & \text{otherwise} \end{cases}, \text{ for } a \in \mathcal{N}_{\text{SBS}} \tag{17}$$

Update the cumulative estimated energy consumption of each ranking expert $e \in \tilde{\mathcal{E}}$

$$\hat{L}_t(e) = \hat{L}_{t-1}(e) + \hat{X}_t(a_t)\delta_{a_t}^e(t) \tag{18}$$

For $a \in \mathcal{N}_{\text{SBS}}$ set

$$q_e(t+1) = \frac{\exp(-\gamma_t \hat{L}_t(e))}{\sum_{e'\in\tilde{\mathcal{E}}} \exp(-\gamma_t \hat{L}_t(e'))} \tag{19}$$

$t = t + 1$

end

Let $\boldsymbol{\delta}^e = (\delta_1^e, \ldots, \delta_N^e)$ denote the action choice vector of expert e, where $\delta_a^e = 1$ denotes the event that expert e recommends SBS a, and $\delta_a^e = 0$ denotes the event that expert e does not recommend SBS a. It is assumed that each expert recommends only one SBS in $\mathcal{N}_{\text{SBS}}$. Since we have both handover cost and dynamic SBS activity, we consider experts whose recommendation strategy at time t depends on both a_{t-1} and $\mathcal{N}_t$. This is a critical step, because otherwise the handover energy consumption will not be considered by the pool of experts. We specifically focus on *ranking experts* whose preference over the set of SBSs is given by a previous action-dependent ranking.

Definition 1. *An expert is called a* ranking expert *if for each previous action $a \in \mathcal{N}_{SBS}$, expert e has a ranking over $\mathcal{N}_{SBS}$ given by $\sigma_{e,a}$. Let $\mathcal{E}$ denote the set of all possible ranking experts, with size $N_E := |\mathcal{E}|$.*

One benefit of considering ranking experts is that the ranking can be performed on the entire set of SBS $\mathcal{N}_{\text{SBS}}$, but the recommendation can take into account of the SBSs that are turned off, by removing them from the ordered set. Specifically, given $\mathcal{N}_t$, expert $e \in \mathcal{E}$ recommends the action with the highest rank in $\mathcal{N}_t$, which is denoted by $\sigma_{e,a}(\mathcal{N}_t)$.

Different from the definition of regret in (9), which is with respect to the best fixed *action*, we define the regret of the UE in this section with respect to the best fixed *expert*. For expert $e \in \mathcal{E}$, let a_t^e denote the action recommended at time t, and a_t denote the random variable that represents the action chosen by the UE at time t. Since e is a ranking expert, a_t^e depends on $\mathcal{N}_t$ and a_{t-1}^e. We define the regret with respect to the best expert from a pool of experts $\mathcal{E}$ as

$$\begin{aligned} R_{\mathbf{a}}(\mathcal{E}, T) := &\sum_{t=1}^{T} \mathbb{E}\left[E_t(a_t) + E_s \mathbb{1}_{\{a_t \neq a_{t-1}\}}\right] \\ &- \mathbb{E}\left[\min_{e\in\mathcal{E}} \sum_{t=1}^{T} \left[E_t(a_t^e) + E_s \mathbb{1}_{\{a_t^e \neq a_{t-1}^e\}}\right]\right] \end{aligned} \tag{20}$$

Due to a technicality, we need to introduce a *uniform expert* (denoted by Unif) in order to bound the regret [18]. It is defined as the following.

Definition 2. *An expert is called a* uniform expert *('Unif') if it recommends action $a \in \mathcal{N}_{SBS}$ with probability $1/N$, regardless of whether $a \in \mathcal{N}_t$.*

We denote the extended pool of experts which includes all the experts in $\mathcal{E}$ and the uniform expert as $\tilde{\mathcal{E}}$. Instead of having a deterministic SBS selection rule like the other experts, the uniform expert selects its action at time slot t according to the uniform distribution on $\mathcal{N}_{\text{SBS}}$ independently from past action and $\mathcal{N}_t$. Hence the uniform expert can recommend actions that are not in $\mathcal{N}_t$, which is now allowed. To address this issue, we assume that the UE randomly selects one of the actions in $\mathcal{N}_t$ if Algorithm 3 recommends an action that is not in $\mathcal{N}_t$.

The theorem below gives a regret bound with respect to the best expert from the pool of experts defined above.

Theorem 4. *Assume that the UE uses the RE algorithm with the pool of ranking experts $\mathcal{E}$ defined in Definition 1 with $\gamma_t = \sqrt{\frac{\log(1+N_E)}{tN}}$. We have*

$$R_{\mathbf{a}}(\mathcal{E}, T) \leq 2N\sqrt{TN\log N}. \tag{21}$$

Proof. See Appendix E. □

We have the following two remarks regarding the RE algorithm and its regret analysis for dynamic SBS on/off. Firstly, when the SBS can be turned on and off, the definition of the regret with respect to the best fixed SBS is no longer a strong definition, as any fixed SBS can be off for some

slots. This is what motivated the regret definition with respect to the best fixed expert. Second, the ranking expert approach can simultaneously take care of handover cost (by letting the expert consider the previous action) and SBS on/off (by letting the expert recommend ranked SBS that is not off).

B. Contextual Ranking Expert

Although the RE algorithm defined in Algorithm 3 achieves sublinear regret with respect to T and the regret bound given in Theorem 4 depends logarithmically on the number of ranking experts, one significant drawback is that maximum number of experts equals $N_E = (N!)^N$, which makes practical implementations computationally challenging since the algorithm needs to keep a probability distribution over the set of experts, which is very large even for moderate N.

To address this practical issue, we re-visit the definition of ranking expert and reduce the number of experts by *defining ranking experts in the context of previous actions*. Consider the following contextual experts problem. Let $\mathcal{X}$ denote the context space. For a sequence of contexts $x_1, x_2, \ldots, x_T$, let $\tau_x \subset \{1, \ldots, T\}$ denote the set of time slots for which the context is $x \in \mathcal{X}$. For the mobility management problem, we take the context at time t to be the last action selected by the learning algorithm, i.e., $x_t = a_{t-1}$.[3] Based on this, the contextual regret of the learning algorithm that works on a set of experts $\mathcal{E}$ is defined as

$$R_C(\mathcal{E}, T) := \sum_{b \in \mathcal{N}_{\text{SBS}}} \mathbb{E}\Big[\sum_{t \in \tau_b} \Big[E_t(a_t) + E_s \mathbb{1}_{\{a_t \neq b\}} \Big] - \min_{e \in \mathcal{E}} \sum_{t \in \tau_b} \Big[E_t(a_t^e(b)) + E_s \mathbb{1}_{\{a_t^e(b) \neq b\}} \Big] \Big] \tag{22}$$

where $a_t^e(b)$ denotes the action chosen by expert e at time t based on context b and the set of available actions $\mathcal{N}_t$, and the expectation is taken with respect to the randomization of the learning algorithm.

With the introduction of contextual experts, we can reduce the number of ranking experts exponentially by using a variant of the RE algorithm. Formally, we define the set of contextual ranking experts, which in contrast to the set of experts given in Definition 1, do not take into account the previous action when ranking the actions.

Definition 3. *An expert e is called a* basic ranking expert *if it has a ranking over $\mathcal{N}_{SBS}$ given by σ_e. Let $\mathcal{E}$ denote the set of all possible basic ranking experts.*

We are now in the position to propose a Contextual Ranking Expert (CRE) algorithm, which is given in Algorithm 4. The CRE algorithm uses the last action as the *context* and learns the best expert independently for each context. CRE runs a different instance of ranking experts for each context. It keeps a different probability vector over the set of experts and actions for each $x \in \mathcal{X}$, and updates these probability vectors only when the corresponding context is observed. The parameter $\kappa(x)$ counts the number of times context x has occurred up to the current time. Instead of t, $\kappa(x)$ is used to adjust the learning rate of each ranking expert that runs for different contexts. This way, each RE algorithm is guaranteed to achieve sublinear regret with respect to the best expert for its context.

[3]Note that in this definition the context is endogenously defined, i.e., it depends on the actions selected by the learning algorithm.

Algorithm 4: The Contextual Ranking Expert (CRE) mobility management algorithm.

Input: A non-increasing sequence $\{\gamma_t\}_{t \in \mathbb{N}}$; $\tilde{\mathcal{E}} = \mathcal{E} \cup \text{Unif}$
Initialize: $q_{e,x}(1) = 1/|\tilde{\mathcal{E}}|$, $\kappa(x) = 1$ and $\hat{L}_{0,x}(e) = 0$ for all $e \in \tilde{\mathcal{E}}$, $x \in \mathcal{N}_{\text{SBS}}$, $a_0 = \text{Rand}(\mathcal{N}_{\text{SBS}})$
while $t \geq 1$ **do**
Observe $x_t = a_{t-1}$ and $\mathcal{A}_t$
Get ranking expert advices $\{\boldsymbol{\delta}^e(t)\}_{e \in \tilde{\mathcal{E}}}$
Set

$$p_a(t) = \sum_{e \in \tilde{\mathcal{E}}} q_{e,x_t}(\kappa(x_t)) \delta_a^e(t) \tag{23}$$

Select a_t randomly according to the probabilities $p_a(t)$, $a \in \mathcal{N}_{\text{SBS}}$
Observe energy consumption $\tilde{E}_t(a_t) \in [0, 1]$
Set

$$\hat{X}_t(a) = \begin{cases} \tilde{E}_t(a)/p_a(t) & \text{if } a = a_t \\ 0 & \text{otherwise} \end{cases}, \text{ for } a \in \mathcal{N}_{\text{SBS}} \tag{24}$$

Update the cumulative estimated energy consumption of each ranking expert $e \in \mathcal{E}$ for context x_t

$$\hat{L}_{\kappa(x_t),x_t}(e) = \hat{L}_{\kappa(x_t)-1,x_t}(e) + \hat{X}_t(a_t) \delta_{a_t}^e(t) \tag{25}$$

For $a \in \mathcal{N}_{\text{SBS}}$ set

$$q_{e,x_t}(\kappa(x_t)+1) = \frac{\exp(-\gamma_{\kappa(x_t)} \hat{L}_{\kappa(x_t),x_t}(e))}{\sum_{e' \in \mathcal{E}} \exp(-\gamma_{\kappa(x_t)} \hat{L}_{\kappa(x_t),x_t}(e'))} \tag{26}$$

Set $\kappa(x_t) = \kappa(x_t) + 1$
$t = t + 1$
end

The following theorem bounds the contextual regret of Algorithm 4.

Theorem 5. *Assume that the UE uses the CRE algorithm with the pool of ranking experts $\mathcal{E}$ given in Definition 3 with $\gamma_t = \sqrt{\frac{\log(N!+1)}{tN}}$. Then we have*

$$R_{\mathbf{a}}(\mathcal{E}, T) \leq 2N^2 \sqrt{T \log N}. \tag{27}$$

Proof. See Appendix F. □

V. Simulation Results

In order to verify the proposed mobility management design, we resort to numerical simulations. In particular, a system-level simulator is developed in which the geometry of UE/SBS and the UE movement are explicitly modelled. Our simulator adopts the general urban deployment model in [27]. The simulation setting is created to highlight the FHO problem for stationary or slow-moving user, which is the focus of our paper. Specifically, we assume that there is a house of size 14×14 square meters in the middle of the simulated area, and

TABLE II
SIMULATION PARAMETERS

Parameters	Value	Parameters	Value
N	6, 12	Pathloss model	3GPP in-to-out [27]
M	6	Shadowing	log-normal with 5dB variance
Maximum UEs per SBS	3	SBS transmit power	15dBm
Energy threshold for 3GPP-macro	10%	FHO count threshold for 3GPP-FHO	4 out of 20
Thermal noise density	-174dBm/Hz	UE noise figure	5.5dB
Carrier frequency	2.1GHz	Bandwidth	20MHz
Penetration loss (L_{ow})	10dB	d_0	1m

N SBSs are symmetrically placed around the room. Note that the symmetrical layout is made to speed up the simulations as well as to create a more severe FHO environment. The distance from each SBS to the center of the house is 80 meters. SBSs are transmitting at a fixed power of 15dBm. On average, there are M UEs in the room, and we adopt a simple random waypoint mobility model [28] with low speed to address user mobility. In particular, we trace the slow movement of one particular UE (the UE of interest), while allowing other UEs to randomly leave or enter the network, and move around with different serving SBSs. As a result, the UE of interest will see dynamic energy consumption from its varying serving SBSs. The total energy consumption is normalized. We consider the 3GPP pathloss model that is recommended for system simulations of small cells and heterogeneous networks. Particularly, we consider the pathloss model suggested in [27]:

$$PL(d)[dB] = 15.3 + 37.6 \times \log_{10}(d) + L_{ow}, d > d_0. \quad (28)$$

Some other system simulation parameters are summarized in Table II.

We first study the energy consumption performance of the proposed BREW algorithm and compare with the existing 3GPP solutions described in Section III-E. In particular, we include both the original threshold-based handover rule and the enhanced FHO-aware policy, labeled as *3GPP-macro* and *3GPP-FHO*, respectively, in the plots. The SINR threshold for 3GPP-macro is set such that the corresponding normalized average energy consumption is above 10%, with no additional offset. The enhanced FHO-aware solution adopts a freezing period that is of the same length as the BREW batch length for fair comparison. Furthermore, the threshold is set to be 4 handovers over the past 20 slots.

The performance comparison is reported in Figure 3 for $N = 6$ and Figure 4 for $N = 12$, respectively, where the latter represents an extreme UDN deployment. In particular, the total energy consumption of each algorithm is compared against the genie-aided solution where the UE selects the best SBS with the minimum energy consumption from the very beginning. The regret of each algorithm is normailized by time. A few important observations can be made from this system level simulation. First of all, we can see that 3GPP-FHO outperforms 3GPP-macro in terms of energy consumption, but both solutions do not exhibit a decaying per-slot regret, confirming the regret analysis in Section III-E. As a result, these solutions cannot converge to the optimal SBS asymptotically. The proposed BREW algorithm, however, has

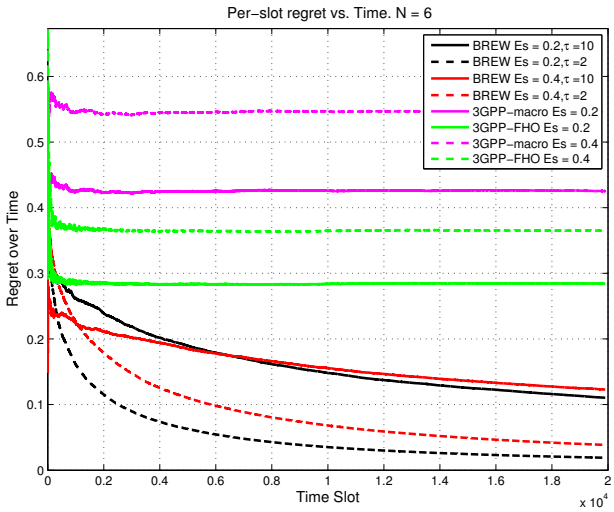


Fig. 3. Comparison of the per-time-slot energy consumption loss versus time for BREW, 3GPP-macro and 3GPP-FHO. $N = 6$.

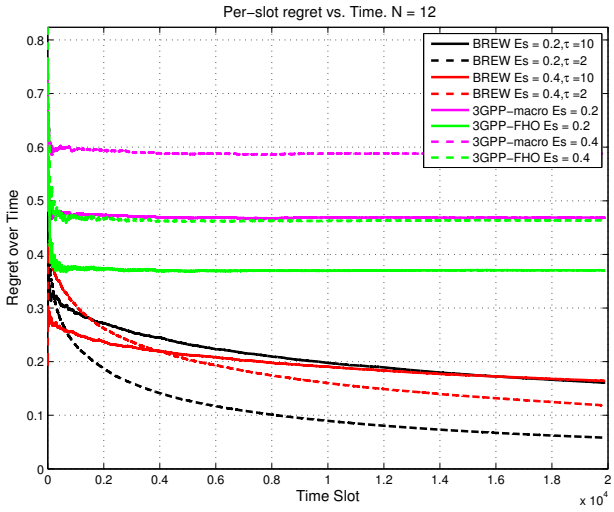
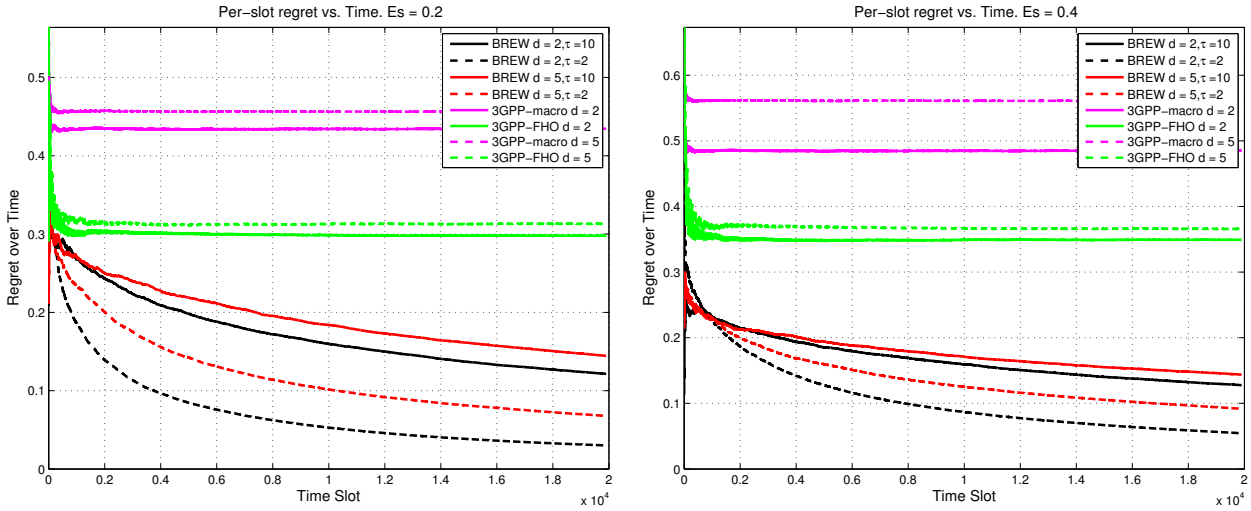


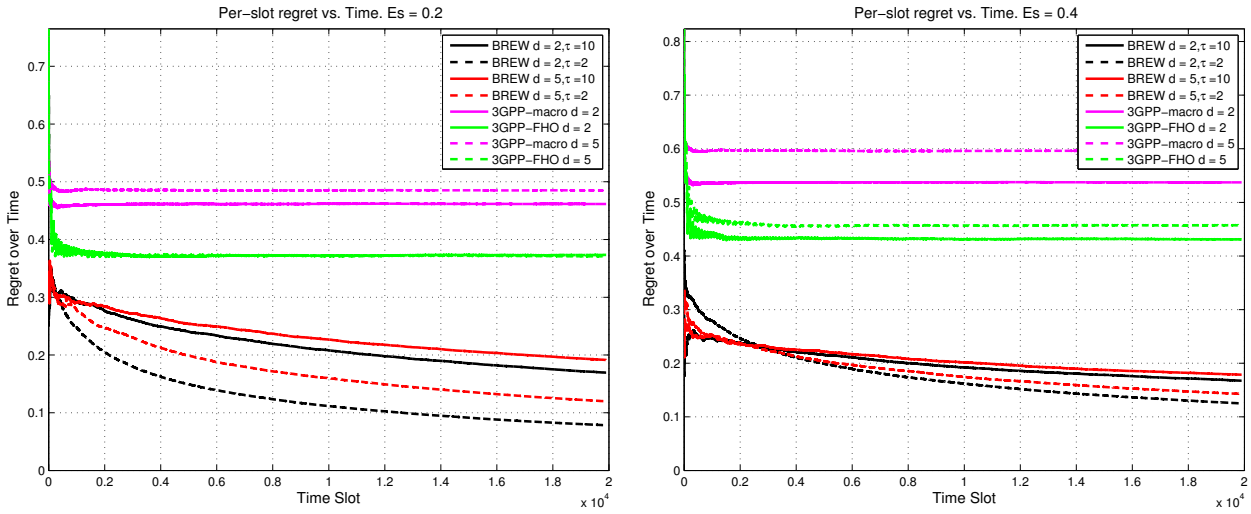
Fig. 4. Comparison of the per-time-slot energy consumption loss versus time for BREW, 3GPP-macro and 3GPP-FHO. $N = 12$.

a diminishing per-slot regret and will converge to the best SBS, supporting the regret analysis in Section III-C. Second, the performance of existing solutions degrade significantly once the handover cost is explicitly taken into account, and such degradation remains constant over time. The increased handover cost also impacts the energy consumption of the proposed BREW algorithm, but thanks to the batched nature



(a) $N = 6$, $E_s = 0.2$

(b) $N = 6$, $E_s = 0.4$



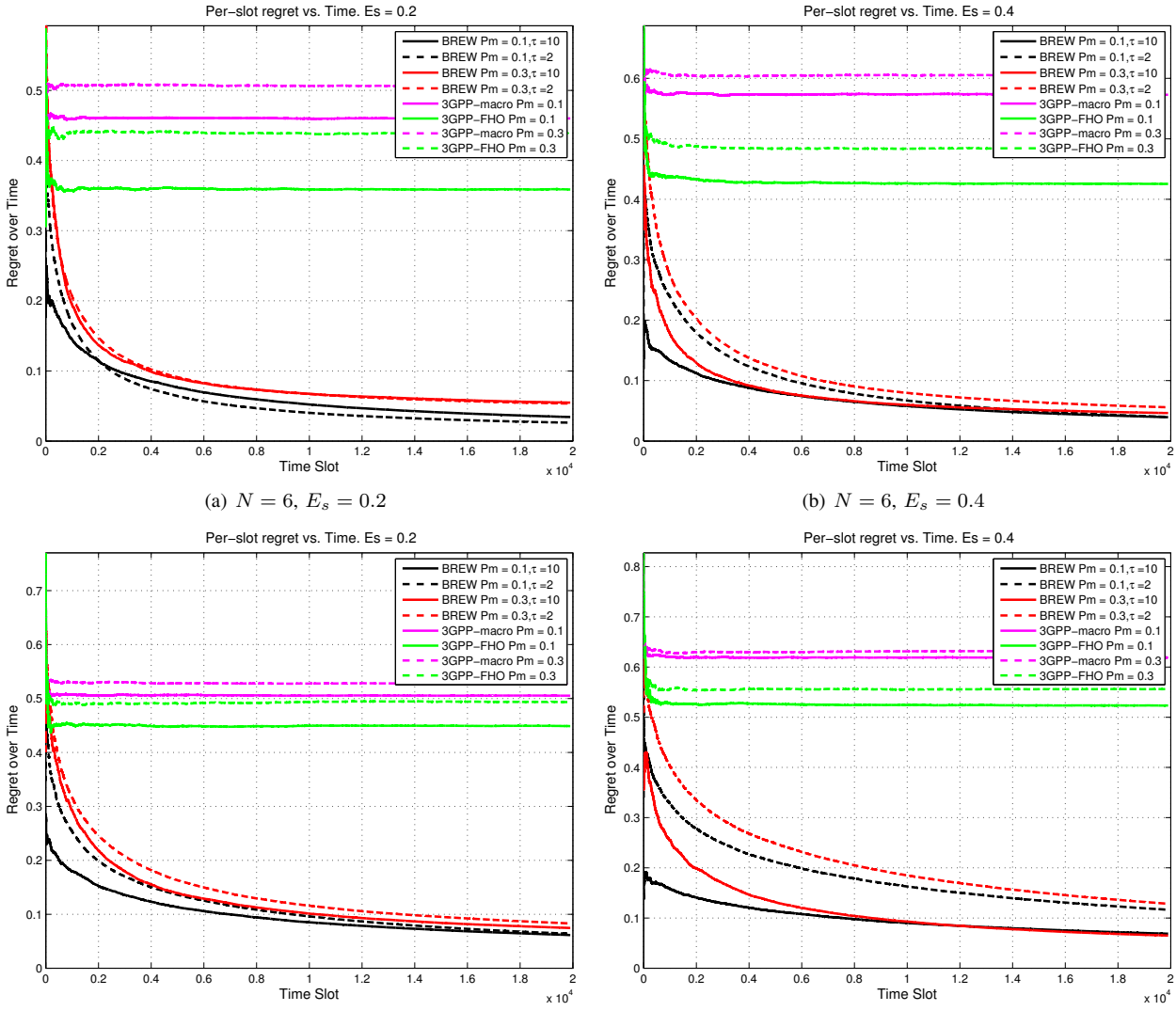
(c) $N = 12$, $E_s = 0.2$

(d) $N = 12$, $E_s = 0.4$

Fig. 5. Impact of delayed feedback to the regret for BREW, 3GPP-macro and 3GPP-FHO.

and the built-in exploration-exploitation tradeoff, the amount of handovers will gradually reduce over time, mitigating the effect of the increased handover cost. For the considered system simulations, the proposed BREW solution can achieve $20\% - 30\%$ less energy consumption (depending on the parameter setting) with a moderate time duration, and more than 60% gain asymptotically, over the existing solutions. Finally, the effect of batch size τ can be analyzed from the figures, which reveals the inherent tradeoff between quick exploration (and hence finding the optimal SBS faster) and the handover cost associated with such exploration. As we can see, for both $N = 6$ and $N = 12$, there exists an initial period where a large batch size results in less energy consumption. This is because in the initial time slots, a small batch size would lead to more frequent handovers for exploration, which results in both more handover costs and selecting sub-optimal SBSs more. However, as time goes by, the speed of exploration slows down, and we will enter a separate region where a large batch size leads to more time spent on sub-optimal SBSs, which increases the energy consumption.

Figure 5 studies the impact of delayed feedback on the performance of the three algorithms. We can see that additional delays of sending the energy consumption feedback increases the regret for all algorithms, and larger delay leads to more severe regret increase. However, an important observation from Figure 5 is that the impact of delayed feedback on BREW is mild when the batch size is moderate or the handover cost is large. This is due to the fact that when the batch size is not very small, the handover decision will be slightly postponed due to the delayed feedback, and the UE of interest stays on the same SBS while waiting for feedback. Because the accumulated feedback comes from a batch, a slight offset will not significantly alter the averaged feedback of the batch, which provides robustness against information obsolete. Additionally, the larger handover cost will further penalize myopic protocols, where the handover decisions are based on outdated information. The simulation results show the robustness of the BREW algorithm against delayed feedback.



(a) $N = 6$, $E_s = 0.2$

(b) $N = 6$, $E_s = 0.4$

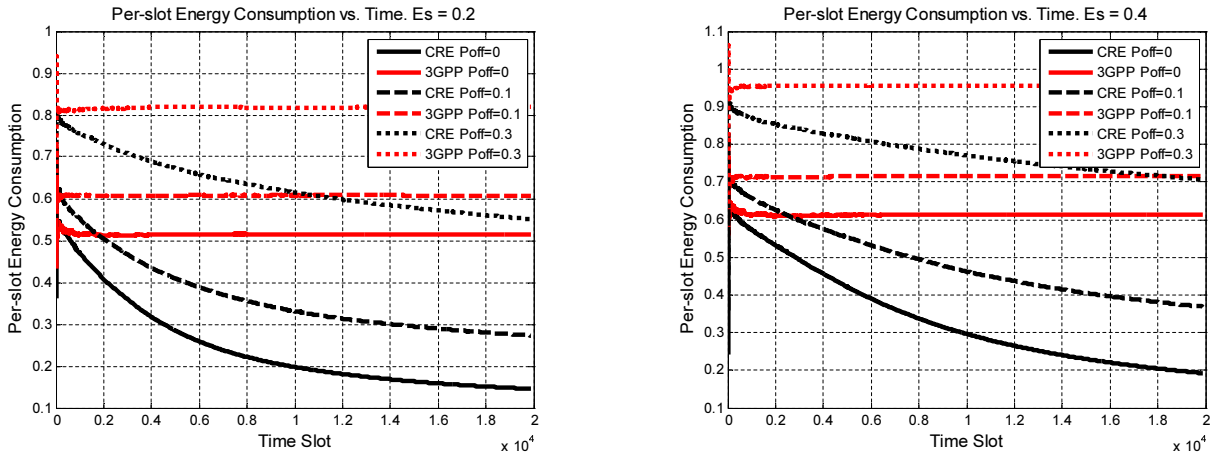
(c) $N = 12$, $E_s = 0.2$

(d) $N = 12$, $E_s = 0.4$

Fig. 6. Impact of missing feedback to the regret for BREW, 3GPP-macro and 3GPP-FHO.

Similarly, Figure 6 reports the simulation results when each time slot, the feedback energy consumption may be missing following an i.i.d. Bernoulli model with missing probability P_m. The BREW algorithm used in the simulation is the extended version as in Algorithm 2 that considers P_m. It can be concluded that missing feedback impact all three algorithms in terms of the regret performance, but with different behavior. For the two 3GPP solutions, the regret quickly converges and there exists an almost constant gap asymptotically. For the extended BREW algorithm, however, there exists a rather large gap during the initial period. This is due to the lack of accurate information and hence missing feedback have a bigger impact to the regret. As the algorithm gradually learns the loss information, the impact of missing feedback diminishes, as shown by the very small gap between different P_m values for large t.

Finally, we study the impact of dynamic SBS on/off to the mobility algorithms. We assume that the N neighboring SBSs are installed by end-users and they can be turned on and off at the users' discretion. To model the dynamic SBS presence, we assume that at each time slot, all N SBSs may independently be turned on or off following an identical Bernoulli distribution with off-probability P_{off}. It is worth noting that this model presents a much bigger challenge than models where SBS dynamics have patterns. As has been discussed, the CRE algorithm presented in Section IV achieves a good tradeoff between complexity (number of experts) and performance. We compare the average per-slot energy consumption of CRE to 3GPP, which at each slot selects the best SBS that is not turned off. For a fair comparison, we also assume that the 3GPP metric takes into account the potential handover cost E_s, which is not considered in the standard 3GPP-macro algorithm. Figure 7 presents the numerical comparison of these two algorithms with different P_{off} values. Clearly, dynamic SBS affects both algorithms, but CRE quickly outperforms the extended 3GPP-macro algorithm and the per-slot energy consumption decreases as time goes by. This is due to the gradual convergence to the best expert in the expert pool.



(a) $E_s = 0.2$ (b) $E_s = 0.4$

Fig. 7. Impact of dynamic SBS on/off to the per-time-slot energy consumption versus time for CRE and 3GPP-macro.

VI. Conclusions

Emerging wireless networks have become more heterogeneous and the network density has increased significantly, both of which pose significant challenges to energy efficient mobility management. Existing solutions, mostly based on optimizing immediate system objectives, fail to achieve long-term minimum energy consumption in highly dynamic and complex wireless networks. To address this problem, we have made two novel contributions. The first is that we adopt a *non-stochastic* online-learning approach to model the UDN mobility management. The key benefit of this approach, as its name suggests, is that we do not need any assumption on the statistical behavior of the SBS activities. This is extremely desirable for UDN. The other novelty is that we explicitly add the *handover cost* to the utility function, which forces the resulting solution to minimize frequent handovers.

Built upon these two key ideas, we have proposed the BREW algorithm which relies on batching to explore in bulk, thus reducing the handovers that are typically required for exploration. A sublinear regret upper bound for BREW is proved. We then study how the BREW algorithm can be adjusted to deal with various system imperfections, including delayed or missing feedback. Most importantly, we have studied the impact of dynamic SBS on/off, which often arises in user-deployed small cell networks. We first prove an impossibility result with respect to any arbitrary SBS on/off. Then, a novel strategy, called ranking expert (RE), is proposed to simultaneously address the handover cost and the availability of SBS. The complete RE algorithm results in a large number of experts, which incurs significant complexity. We further propose a contextual ranking expert (CRE) algorithm that reduces the number of experts significantly. Regret bound is proved for both RE and CRE with respect to the best expert. Simulation results show a significant improvement to the overall system energy consumption. More importantly, the gain is robust against various system dynamics.

There are some interesting problems that have not been fully addressed in this work, which are the subjects of potential future work. For example, the regret upper bounds developed in this work are mostly for a given set of algorithm parameters, for which sublinear regret is rigorously proven. It is of interest to study the performance bound variation and tightness with respect to the algorithm parameters. Another important question is how to further enhance the ranking expert solutions in Algorithm 3 and 4, in terms of the algorithm complexity and the corresponding regret bound.

Appendix A
Proof of Theorem 1

We cite two known results in non-stochastic bandit theory that will be used in the proof. These results are modified to fit into the problem setting of Theorem 1.

Proposition 2. *(Part of Theorem 2.1 in [19]) The standard pseudo-regret bound for the any-time EXP3 algorithm [19], where the parameter γ_t does not depend on the time horizon T, is given by*

$$R(T) \leq \sqrt{4.5TN \log N}, \tag{29}$$

for $\gamma_t = \sqrt{(2 \log N)/(tN)}$, $t \in \mathbb{N}_+$.

Proposition 3. *(Part of Theorem 2 in [29]) When $\tau > 1$, the regret $R'(T)$ of an algorithm with respect to the constant actions when the reward sequence is generated by an m-memory-bounded adaptive adversary is bounded by*

$$R'(T) \leq \tau R(\frac{T}{\tau}) + \frac{Tm}{\tau} + \tau \tag{30}$$

where $R(T)$ denotes an upper bound on the standard pseudo-regret of the algorithm.[4]

The proof of Theorem 1 then follows by recognizing that the adversarial bandit problem with switching costs is a

[4]Definitions of m-memory-bounded adaptive adversary and standard pseudo-regret can be found in [29].

special case of a $m = 1$ memory bounded adversary. With $\tau = \lceil B_N T^{1/3} \rceil$, we have

$$R\left(\frac{T}{\tau}\right) \leq \sqrt{\frac{TB_N^{-3}}{\lceil B_N T^{1/3} \rceil}} \leq T^{1/3} B_N^{-2}, \tag{31}$$

and thus

$$\begin{aligned} R_{\mathbf{a}}(T) &\leq \tau T^{1/3} B_N^{-2} + \frac{T}{\tau} + \tau \\ &\leq 2B_N^{-1} T^{2/3} + \left(B_N + B_N^{-2}\right) T^{1/3} + 1. \end{aligned} \tag{32}$$

Appendix B
Proof of Theorem 2

In [29] a d-memory-bounded adversary is defined as an adversary which is restricted to choose a loss function that depends on the $d+1$ most recent actions of the learner. Hence, the loss of the learner generated by a d-memory-bounded adversary at time slot t can be written as $f_t(a_{t-d}, \ldots, a_t)$. Now consider our setting in which the energy cost of choosing an SBS at time t only depends on the state of the network at time t. This cost is given by the function $\{E_t(a)\}_{a \in \mathcal{N}_{\text{SBS}}}$. When the feedback is received by the learner with a delay of d time slots, we can model the cost incurred by the learner as cost assigned by a d-memory-bounded adversary whose loss function is

$$f_t(a_{t-d}, \ldots, a_t) = E_{t-d}(a_{t-d}) + E_s \mathbb{1}_{\{a_{t-1} \neq a_t\}} \tag{33}$$

The result then follows the same steps as Appendix A.

Appendix C
Proof of Proposition 1

We separately prove the linear regret in T for stochastic and non-stochastic energy consumption models. For a stochastic model, we denote the average RSRP or RSRQ of each SBS as $r_n = \mathbb{E}[R_t(n)]$, and the probability that the metric of SBS n falls below the pre-determined threshold θ as $\sigma_n = P(R_t(n) < \theta)$. Without loss of generality and to avoid trivial conditions, we assume that $r_1 > r_2 > \cdots > r_N$.

A regret lower bound for the 3GPP handover protocols described in Section III-E can be achieved by a genie-aided policy where switching only happens between SBS 1 and 2. In other words, whenever the performance metric of the best SBS falls below θ, the user only switches to the second-best SBS; when the performance metric of the second-best SBS falls below θ, it comes back to the best SBS. This policy can be modelled as a two-state Markov process where each state n has a transition probability σ_n. We denote the steady-state distribution for the sub-optimal SBS 2 as ρ_2, and it can be shown $\rho_2 > 0$ for non-trivial cases. Thus, the genie-aided policy achieves a linear regret $\rho_2(r_1 - r_2)T$ asymptotically.

For a non-stochastic model, we prove the linear regret in T by constructing a specific sequence of metrics $\{R_t(n)\}$. For simplicity, we only give one example for $N = 2$. Consider $R_1(1) > R_1(2)$ so that at time slot 1 the best SBS is selected. Then we let $R_2(1) < \theta < R_2(2)$ so that the UE switches to the sub-optimal SBS. We then fix $R_t(1) > R_t(2) > \theta$ for all $t > 2$. In this example, the UE will be stuck with the sub-optimal SBS 2 from time slot 2 to T, thus achieving a linear regret in T.

Appendix D
Proof of Theorem 3

The adversary defined in Theorem 3 generates $\mathcal{N}_{t+1}$ based on a_t. Consider the worst-case scenario where it simply lets $\mathcal{N}_{t+1} = \mathcal{N}_{\text{SBS}} - \{a_t\}$. This forces the UE to switch at every time slot. Hence it incurs a handover loss of $E_s T$. As a result, the loss of the learning algorithm is at least $E_s T$.

Since only one SBS is inactive in each time slot, there exists at least one SBS which is active in at least $T(1 - 1/N)$ time slots. Let $\tilde{a}$ denote such an SBS. SBS $\tilde{a}$ will be inactive in at most T/N time slots, which means that any policy that selects SBS $\tilde{a}$ when it is available needs to switch at most T/N times. Thus, the cost of such a policy is bounded above by $T(1-1/N)E_{\max} + T/N$, where the first term denotes the worst-case energy consumption from $\tilde{a}$ at time slots when it is active and the second term denotes the worst-case energy consumption plus handover cost due to the slots in which $\tilde{a}$ is inactive[5]. As a result, we have that the cost of $\tilde{a}$ is bounded above by $T(1 - 1/N)E_{\max} + T/N$.

Let $\tilde{a}^*$ denote the best SBS (the one whose cumulative loss is minimum). Then, the loss of $\tilde{a}^*$ is upper bounded by $T(1 - 1/N)E_{\max} + T/N$.

Hence, the difference between the loss of the learning algorithm and the loss of $\tilde{a}^*$ is at least

$$E_s T - T(1 - \frac{1}{N})E_{\max} - \frac{T}{N} \tag{34}$$

$$= \frac{T}{N}(NE_s - (N-1)E_{\max} - 1) \tag{35}$$

$$\geq \frac{TE_{\max}}{N} \tag{36}$$

where the inequality follows from $E_s \geq E_{\max} + 1/(N-1)$. This proves that the regret is linear in T.

Appendix E
Proof of Theorem 4

Note that the SBS activity evolves independently of the actions of the UE. Hence, the adversary is only able to modify the current reward of the UE based on its current action. Hence, the adversary is oblivious to the actions of the UE. Therefore, we can use Theorem 4.2 in [21] to bound the regret. The number of experts including the uniform expert is $(N!)^N + 1$. We obtain the result by observing that

$$\begin{aligned} \log((N!)^N + 1) \leq \log((N! + 1)^N) &= N \log(N! + 1) \\ &\leq N^2 \log N, \end{aligned} \tag{37}$$

where the last inequality comes from $\log(N! + 1) \leq N \log N$.

Appendix F
Proof of Theorem 5

We have that $\{\tau_b\}_{b \in \mathcal{N}_{\text{SBS}}}$ is a random variable which depends on the randomization and the history of actions selected by CRE. Our regret bound will hold for any realization of $\{\tau_b\}_{b \in \mathcal{N}_{\text{SBS}}}$. Consider the loss function

$$E_t(a_t) + E_s \mathbb{1}_{\{a_t \neq b\}}.$$

[5]Recall that the normalized energy consumption in a time slot is upper bounded by 1.

By the definition of contextual regret and because E_t is generated by an oblivious adversary, for any $b \in \mathcal{N}_{\text{SBS}}$ we have

$$\begin{aligned}
&\sum_{t\in\tau_b} \mathbb{E}\left[E_t(a_t) + E_s \mathbb{1}_{\{a_t \neq b\}}\right] \\
&\quad - \min_{e\in\mathcal{E}} \sum_{t\in\tau_b} \left[E_t(a_t^e(b)) + E_s \mathbb{1}_{\{a_t^e(b)\neq b\}}\right] \\
\leq\ & 2\sqrt{|\tau_b| N \log(N!+1)} && (38)\\
\leq\ & 2\sqrt{|\tau_b| N^2 \log(N)} && (39)\\
\leq\ & 2\sqrt{T N^2 \log(N)} && (40)
\end{aligned}$$

where (38) comes from the standard regret bound of EXP4, and the expectation is taken with respect to the randomization of CRE when the context is b. Although $\{\tau_b\}_{b\in\mathcal{N}_{\text{SBS}}}$ depends on the randomization of CRE, since the bound derived in (40) is independent of the randomization of CRE, we get the final result by summing (40) over all $b \in \mathcal{N}_{\text{SBS}}$.

REFERENCES

[1] T. Q. S. Quek, G. de la Roche, I. Guvenc, and M. Kountouris, *Small Cell Networks: Deployment, PHY Techniques, and Resource Allocation*. Cambridge University Press, 2013.

[2] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*, 2nd ed. Wiley, 2011.

[3] J. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, "An overview of load balancing in HetNets: old myths and open problems," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 18–25, April 2014.

[4] D. Xenakis, N. Passas, L. Merakos, and C. Verikoukis, "Energy-efficient and interference-aware handover decision for the LTE-Advanced femtocell network," in *IEEE International Conference on Communications (ICC)*, June 2013, pp. 2464–2468.

[5] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, June 2013.

[6] A. Mesodiakaki, F. Adelantado, L. Alonso, and C. Verikoukis, "Energy-efficient context-aware user association for outdoor small cell heterogeneous networks," in *IEEE International Conference on Communications (ICC)*, June 2014, pp. 1614–1619.

[7] S. Althunibat, K. Kontovasilis, and F. Granelli, "A handover policy for energy efficient network connectivity through proportionally fair access," in *Proceedings of European Wireless 2014*, May 2014, pp. 1–6.

[8] C. Bottai, C. Cicconetti, A. Morelli, M. Rosellini, and C. Vitale, "Energy-efficient user association in extremely dense small cell networks," in *2014 European Conference on Networks and Communications (EuCNC)*, June 2014, pp. 1–5.

[9] 3GPP, "Evolved Universal Terrestrial Radio Access; Mobility enhancements in heterogeneous networks," TR 36.839.

[10] M. Simsek, M. Bennis, and I. Guvenc, "Context-aware mobility management in HetNets: A reinforcement learning approach," in *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2015, pp. 1536–1541.

[11] V. Capdevielle, A. Feki, and E. Sorsy, "Joint interference management and handover optimization in LTE small cells network," in *IEEE International Conference on Communications (ICC)*, June 2012, pp. 6769–6773.

[12] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, pp. 4–22, 1985.

[13] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, pp. 235–256, 2002.

[14] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. Info. Theory*, vol. 58, no. 8, pp. 5588–5611, 2012.

[15] "In building solutions," White Paper, Nokia Siemens Networks, 2001.

[16] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, February 2014.

[17] 3GPP, "Evolved Universal Terrestrial Radio Access; S1 Application Protocol (S1AP)," TR 36.413.

[18] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, pp. 48–77, 2002.

[19] S. Bubeck, "Jeux de bandits et fondations du clustering," Ph.D. dissertation, Université Lille 1, 2010.

[20] N. Merhav, E. Ordentlich, G. Seroussi, and M. Weinberger, "On sequential strategies for loss functions with memory," *IEEE Trans. Info. Theory*, vol. 48, no. 7, pp. 1947–1958, Jul 2002.

[21] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[22] O. Dekel, J. Ding, T. Koren, and Y. Peres, "Bandits with switching costs: T 2/3 regret," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. ACM, 2014, pp. 459–467.

[23] D. Chakrabarti, R. Kumar, F. Radlinski, and E. Upfal, "Mortal multi-armed bandits," in *Advances in Neural Information Processing Systems*, 2008.

[24] Z. Bnaya, R. Puzis, R. Stern, and A. Felner, "Volatile multi-armed bandits for guaranteed targeted social crawling," in *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[25] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[26] R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma, "Regret bounds for sleeping experts and bandits," *Machine learning*, vol. 80, no. 2-3, pp. 245–272, 2010.

[27] 3GPP, "Evolved Universal Terrestrial Radio Access; Further advancements for E-UTRA physical layer aspects," TR 36.814.

[28] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[29] R. Arora, O. Dekel, and A. Tewari, "Online bandit learning against an adaptive adversary: from regret to policy regret," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 1503–1510.

Cong Shen received the B.S. and M.S. degrees, in 2002 and 2004 respectively, from the Electronic Engineering Department, Tsinghua University, Beijing, China, and the Ph.D. degree in 2009 from the Electrical Engineering Department, University of California, Los Angeles (UCLA). He is currently a professor at the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China. His research interests include wireless networks and machine learning.

Cem Tekin is an Assistant Professor in Electrical and Electronics Engineering Department at Bilkent University, Ankara, Turkey. He received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2008, the M.S.E. degree in electrical engineering: systems, M.S. degree in mathematics, Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, in 2010, 2011 and 2013, respectively. From February 2013 to January 2015, he was a Postdoctoral Scholar at University of California, Los Angeles. His research interests include machine learning, multi-armed bandit problems, data mining, multi-agent systems and smart healthcare. He received the University of Michigan Electrical Engineering Departmental Fellowship in 2008, and the Fred W. Ellersick award for the best paper in MILCOM 2009.

Mihaela van der Schaar is Chancellor's Professor in the Electrical Engineering Department at UCLA. Her research interests include communications, engineering economics and game theory, strategic design, online reputation and social media, dynamic multi-user networks, and system designs.